ckb-next

v0.2.8 at branch master

Generated by Doxygen 1.8.6

Sat Jan 20 2018 16:59:47

Contents

1 README 1

2 Building ckb 9

3 cbk Improvements Roadmap 11

4 DAEMON 13

5 Todo List 21

6 Deprecated List 23

7 Namespace Index 25

7.1 Namespace List . . . 25

8 Hierarchical Index 27

8.1 Class Hierarchy . . . 27

9 Data Structure Index 31

9.1 Data Structures . . . 31

10 File Index 35

10.1 File List . . . 35

11 Namespace Documentation 41

11.1 Ui Namespace Reference . . . 41

12 Data Structure Documentation 43

12.1 Ui::AnimAddDialog Class Reference . . . 43

12.1.1 Detailed Description . . . 44

12.1.2 Member Function Documentation . . . 44

12.1.2.1 retranslateUi . . . 44

12.1.2.2 setupUi . . . 45

12.1.3 Field Documentation . . . 46

12.1.3.1 animBox . . . 46

12.1.3.2 buttonBox . . . 47

12.1.3.3 gridLayout . . . 47
12.1.3.4 horizontalLayout . . . 47
12.1.3.5 horizontalSpacer . . . 47
12.1.3.6 horizontalSpacer_2 . . . 47
12.1.3.7 label . . . 47
12.1.3.8 label_2 . . . 47
12.1.3.9 presetBox . . . 47
12.1.3.10 previewBox . . . 47
12.1.3.11 verticalSpacer_2 . . . 48
12.1.3.12 verticalSpacer_3 . . . 48
12.2 AnimAddDialog Class Reference . . . 48
12.2.1 Detailed Description . . . 49
12.2.2 Constructor & Destructor Documentation . . . 49
12.2.2.1 AnimAddDialog . . . 50
12.2.2.2 ~AnimAddDialog . . . 50
12.2.3 Member Function Documentation . . . 50
12.2.3.1 chosenPreset . . . 50
12.2.3.2 chosenScript . . . 51
12.2.3.3 on_animBox_activated . . . 51
12.2.3.4 on_presetBox_activated . . . 52
12.2.3.5 on_previewBox_clicked . . . 53
12.2.4 Field Documentation . . . 53
12.2.4.1 _keys . . . 53
12.2.4.2 _light . . . 53
12.2.4.3 scripts . . . 53
12.2.4.4 showPreview . . . 54
12.2.4.5 ui . . . 54
12.3 AnimScript Class Reference . . . 54
12.3.1 Detailed Description . . . 58
12.3.2 Data Structure Documentation . . . 58
12.3.2.1 struct AnimScript::Param . . . 58
12.3.3 Member Typedef Documentation . . . 59
12.3.3.1 PresetValue . . . 59
12.3.4 Constructor & Destructor Documentation . . . 59
12.3.4.1 ~AnimScript . . . 59
12.3.4.2 AnimScript . . . 59
12.3.4.3 AnimScript . . . 60
12.3.5 Member Function Documentation . . . 60
12.3.5.1 advance . . . 60
12.3.5.2 author . . . 60

12.3.5.3 begin . . . 61
12.3.5.4 colors . . . 62
12.3.5.5 copy . . . 62
12.3.5.6 copyright . . . 63
12.3.5.7 count . . . 63
12.3.5.8 description . . . 64
12.3.5.9 end . . . 64
12.3.5.10 frame . . . 65
12.3.5.11 guid . . . 66
12.3.5.12 hasFrame . . . 66
12.3.5.13 hasKeypress . . . 67
12.3.5.14 hasParam . . . 67
12.3.5.15 init . . . 68
12.3.5.16 keypress . . . 69
12.3.5.17 license . . . 70
12.3.5.18 list . . . 70
12.3.5.19 load . . . 71
12.3.5.20 name . . . 74
12.3.5.21 param . . . 74
12.3.5.22 parameters . . . 75
12.3.5.23 paramIterator . . . 76
12.3.5.24 path . . . 76
12.3.5.25 preset . . . 77
12.3.5.26 presets . . . 77
12.3.5.27 printParams . . . 77
12.3.5.28 readProcess . . . 78
12.3.5.29 retrigger . . . 79
12.3.5.30 scan . . . 80
12.3.5.31 setDuration . . . 81
12.3.5.32 stop . . . 81
12.3.5.33 version . . . 82
12.3.5.34 year . . . 82
12.3.6 Field Documentation . . . 83
12.3.6.1 _colorBuffer . . . 83
12.3.6.2 _colors . . . 83
12.3.6.3 _info . . . 83
12.3.6.4 _keys . . . 83
12.3.6.5 _map . . . 83
12.3.6.6 _paramValues . . . 83
12.3.6.7 _path . . . 83

12.3.6.8 _presets . . . 84
12.3.6.9 _presetValues . . . 84
12.3.6.10 durationMsec . . . 84
12.3.6.11 firstFrame . . . 84
12.3.6.12 inFrame . . . 84
12.3.6.13 initialized . . . 84
12.3.6.14 KP_NAME . . . 84
12.3.6.15 KP_NONE . . . 84
12.3.6.16 KP_POSITION . . . 84
12.3.6.17 lastFrame . . . 85
12.3.6.18 minX . . . 85
12.3.6.19 minY . . . 85
12.3.6.20 process . . . 85
12.3.6.21 readAnyFrame . . . 85
12.3.6.22 readFrame . . . 85
12.3.6.23 repeatMsec . . . 85
12.3.6.24 scripts . . . 85
12.3.6.25 stopped . . . 85
12.4 Ui::AnimSettingDialog Class Reference . . . 86
12.4.1 Detailed Description . . . 88
12.4.2 Member Function Documentation . . . 88
12.4.2.1 retranslateUi . . . 88
12.4.2.2 setupUi . . . 89
12.4.3 Field Documentation . . . 93
12.4.3.1 animName . . . 93
12.4.3.2 buttonBox . . . 93
12.4.3.3 delayBox . . . 93
12.4.3.4 gridLayout . . . 94
12.4.3.5 gridLayout_2 . . . 94
12.4.3.6 horizontalSpacer . . . 94
12.4.3.7 horizontalSpacer_2 . . . 94
12.4.3.8 horizontalSpacer_3 . . . 94
12.4.3.9 horizontalSpacer_4 . . . 94
12.4.3.10 horizontalSpacer_5 . . . 94
12.4.3.11 horizontalSpacer_6 . . . 94
12.4.3.12 infoTab . . . 94
12.4.3.13 kpDelayBox . . . 95
12.4.3.14 kpModeStopBox . . . 95
12.4.3.15 kpReleaseBox . . . 95
12.4.3.16 kpRepeatBox . . . 95

12.4.3.17 kpRepeatLabel . . . 95
12.4.3.18 kpRepeatLabel_2 . . . 95
12.4.3.19 label . . . 95
12.4.3.20 label_10 . . . 95
12.4.3.21 label_11 . . . 95
12.4.3.22 label_12 . . . 96
12.4.3.23 label_13 . . . 96
12.4.3.24 label_2 . . . 96
12.4.3.25 label_3 . . . 96
12.4.3.26 label_4 . . . 96
12.4.3.27 label_5 . . . 96
12.4.3.28 label_6 . . . 96
12.4.3.29 label_7 . . . 96
12.4.3.30 label_8 . . . 96
12.4.3.31 label_9 . . . 97
12.4.3.32 line . . . 97
12.4.3.33 line_2 . . . 97
12.4.3.34 repeatBox . . . 97
12.4.3.35 repeatLabel . . . 97
12.4.3.36 repeatLabel_2 . . . 97
12.4.3.37 sAuthorLabel . . . 97
12.4.3.38 sDescLabel . . . 97
12.4.3.39 settingsGrid . . . 97
12.4.3.40 settingTab . . . 98
12.4.3.41 sLicenseLabel . . . 98
12.4.3.42 sNameLabel . . . 98
12.4.3.43 sVerLabel . . . 98
12.4.3.44 sYearLabel . . . 98
12.4.3.45 tabWidget . . . 98
12.4.3.46 timeGrid . . . 98
12.4.3.47 timeTab . . . 98
12.4.3.48 verticalSpacer . . . 98
12.4.3.49 verticalSpacer_2 . . . 99
12.4.3.50 verticalSpacer_3 . . . 99
12.5 AnimSettingDialog Class Reference . . . 99
12.5.1 Detailed Description . . . 102
12.5.2 Constructor & Destructor Documentation . . . 102
12.5.2.1 AnimSettingDialog . . . 102
12.5.2.2 ~AnimSettingDialog . . . 107
12.5.3 Member Function Documentation . . . 107

12.5.3.1 angleDialChanged . . . 108
12.5.3.2 angleSpinnerChanged . . . 108
12.5.3.3 hLine . . . 109
12.5.3.4 name . . . 110
12.5.3.5 newDuration . . . 110
12.5.3.6 on_delayBox_valueChanged . . . 111
12.5.3.7 on_kpDelayBox_valueChanged . . . 112
12.5.3.8 on_kpReleaseBox_stateChanged . . . 112
12.5.3.9 on_kpRepeatBox_valueChanged . . . 113
12.5.3.10 on_repeatBox_valueChanged . . . 114
12.5.3.11 updateParam . . . 114
12.5.3.12 updateStops . . . 117
12.5.4 Field Documentation . . . 118
12.5.4.1 _anim . . . 118
12.5.4.2 angleDialMapper . . . 118
12.5.4.3 angleSpinnerMapper . . . 118
12.5.4.4 angleSpinners . . . 119
12.5.4.5 hasRepeat . . . 119
12.5.4.6 kpStopCheck . . . 119
12.5.4.7 lastDuration . . . 119
12.5.4.8 settingWidgets . . . 119
12.5.4.9 stopCheck . . . 119
12.5.4.10 ui . . . 119
12.5.4.11 updateMapper . . . 119
12.6 AutoRun Class Reference . . . 120
12.6.1 Detailed Description . . . 120
12.6.2 Member Function Documentation . . . 120
12.6.2.1 available . . . 120
12.6.2.2 disable . . . 121
12.6.2.3 enable . . . 121
12.6.2.4 isEnabled . . . 122
12.6.2.5 once . . . 123
12.7 CkbSettings Class Reference . . . 124
12.7.1 Detailed Description . . . 126
12.7.2 Constructor & Destructor Documentation . . . 126
12.7.2.1 CkbSettings . . . 126
12.7.2.2 CkbSettings . . . 126
12.7.2.3 CkbSettings . . . 127
12.7.2.4 ~CkbSettings . . . 127
12.7.3 Member Function Documentation . . . 127

12.7.3.1 beginGroup . . . 127
12.7.3.2 childGroups . . . 127
12.7.3.3 childKeys . . . 128
12.7.3.4 cleanUp . . . 129
12.7.3.5 contains . . . 130
12.7.3.6 containsGroup . . . 130
12.7.3.7 endGroup . . . 131
12.7.3.8 get . . . 131
12.7.3.9 isBusy . . . 132
12.7.3.10 pwd . . . 133
12.7.3.11 pwd . . . 134
12.7.3.12 remove . . . 135
12.7.3.13 set . . . 135
12.7.3.14 setValue . . . 137
12.7.3.15 value . . . 137
12.7.4 Field Documentation . . . 138
12.7.4.1 backing . . . 138
12.7.4.2 groups . . . 138
12.7.4.3 removeCache . . . 139
12.7.4.4 writeCache . . . 139
12.8 CkbSettingsWriter Class Reference . . . 139
12.8.1 Detailed Description . . . 140
12.8.2 Constructor & Destructor Documentation . . . 140
12.8.2.1 CkbSettingsWriter . . . 140
12.8.2.2 ∼CkbSettingsWriter . . . 141
12.8.3 Member Function Documentation . . . 141
12.8.3.1 run . . . 141
12.8.4 Field Documentation . . . 141
12.8.4.1 _backing . . . 141
12.8.4.2 _removals . . . 141
12.8.4.3 _updates . . . 141
12.9 ColorButton Class Reference . . . 141
12.9.1 Detailed Description . . . 144
12.9.2 Constructor & Destructor Documentation . . . 144
12.9.2.1 ColorButton . . . 144
12.9.3 Member Function Documentation . . . 144
12.9.3.1 allowAlpha . . . 144
12.9.3.2 allowAlpha . . . 145
12.9.3.3 bigIcons . . . 145
12.9.3.4 bigIcons . . . 146

12.9.3.5 color . . . 146
12.9.3.6 color . . . 147
12.9.3.7 colorChanged . . . 148
12.9.3.8 pickColor . . . 148
12.9.3.9 setLabel . . . 149
12.9.3.10 setLabel . . . 149
12.9.3.11 updateImage . . . 149
12.9.4 Field Documentation . . . 150
12.9.4.1 _alpha . . . 150
12.9.4.2 _bigIcons . . . 150
12.9.4.3 _color . . . 150
12.9.4.4 _setLabel . . . 150
12.10 ColorMap Class Reference . . . 151
12.10.1 Detailed Description . . . 152
12.10.2 Constructor & Destructor Documentation . . . 152
12.10.2.1 ColorMap . . . 152
12.10.2.2 ~ColorMap . . . 152
12.10.2.3 ColorMap . . . 152
12.10.3 Member Function Documentation . . . 153
12.10.3.1 alloc . . . 153
12.10.3.2 clear . . . 153
12.10.3.3 colorForName . . . 154
12.10.3.4 colorForName . . . 155
12.10.3.5 colors . . . 155
12.10.3.6 colors . . . 156
12.10.3.7 count . . . 156
12.10.3.8 deAlloc . . . 156
12.10.3.9 init . . . 157
12.10.3.10 keyNames . . . 158
12.10.3.11 operator= . . . 158
12.10.3.12 operator== . . . 159
12.10.4 Field Documentation . . . 159
12.10.4.1 _colors . . . 159
12.10.4.2 _count . . . 159
12.10.4.3 _keyNames . . . 159
12.10.4.4 _mapCount . . . 159
12.11 devcmd.__unnamed__ Struct Reference . . . 160
12.11.1 Detailed Description . . . 161
12.11.2 Field Documentation . . . 161
12.11.2.1 active . . . 161

12.11.2.2 allocprofile . . . 161
12.11.2.3 bind . . . 161
12.11.2.4 dpi . . . 161
12.11.2.5 dpisel . . . 161
12.11.2.6 erase . . . 161
12.11.2.7 eraseprofile . . . 161
12.11.2.8 freeprofile . . . 161
12.11.2.9 fwupdate . . . 161
12.11.2.10 get . . . 161
12.11.2.11 hwload . . . 161
12.11.2.12 hwsave . . . 161
12.11.2.13 iauto . . . 161
12.11.2.14 id . . . 161
12.11.2.15 idle . . . 161
12.11.2.16 inotify . . . 161
12.11.2.17 ioff . . . 162
12.11.2.18 ion . . . 162
12.11.2.19 ilift . . . 162
12.11.2.20 loadprofile . . . 162
12.11.2.21 macro . . . 162
12.11.2.22 name . . . 162
12.11.2.23 notify . . . 162
12.11.2.24 pollrate . . . 162
12.11.2.25 profileid . . . 162
12.11.2.26 profilename . . . 162
12.11.2.27 rebind . . . 162
12.11.2.28 restart . . . 162
12.11.2.29 rgb . . . 162
12.11.2.30 setmodeindex . . . 162
12.11.2.31 snap . . . 162
12.11.2.32 start . . . 162
12.11.2.33 unbind . . . 162
12.11.2.34 updatedpi . . . 162
12.11.2.35 updateindicators . . . 162
12.11.2.36 updatergb . . . 162
12.12 Ui::ExtraSettingsWidget Class Reference . . . 162
12.12.1 Detailed Description . . . 164
12.12.2 Member Function Documentation . . . 165
12.12.2.1 retranslateUi . . . 165
12.12.2.2 setupUi . . . 166

12.12.3 Field Documentation . . . 170
12.12.3.1 animCountLabel . . . 170
12.12.3.2 animPathLabel . . . 170
12.12.3.3 animScanButton . . . 170
12.12.3.4 brightnessBox . . . 170
12.12.3.5 buttonBox . . . 170
12.12.3.6 delayBox . . . 170
12.12.3.7 ditherBox . . . 171
12.12.3.8 fpsBox . . . 171
12.12.3.9 fpsWarnLabel . . . 171
12.12.3.10 gridLayout . . . 171
12.12.3.11 horizontalLayout . . . 171
12.12.3.12 horizontalLayout_2 . . . 171
12.12.3.13 horizontalSpacer . . . 171
12.12.3.14 horizontalSpacer_2 . . . 171
12.12.3.15 horizontalSpacer_3 . . . 171
12.12.3.16 label . . . 172
12.12.3.17 label_10 . . . 172
12.12.3.18 label_11 . . . 172
12.12.3.19 label_2 . . . 172
12.12.3.20 label_3 . . . 172
12.12.3.21 label_5 . . . 172
12.12.3.22 label_6 . . . 172
12.12.3.23 line . . . 172
12.12.3.24 line_2 . . . 172
12.12.3.25 line_3 . . . 173
12.12.3.26 mAccelBox . . . 173
12.12.3.27 osxLabel . . . 173
12.12.3.28 osxLine . . . 173
12.12.3.29 sAccelBox . . . 173
12.12.3.30 sSpeedBox . . . 173
12.12.3.31 sSpeedWidget . . . 173
12.12.3.32 trayBox . . . 173
12.12.3.33 verticalSpacer . . . 174
12.12.3.34 verticalSpacer_10 . . . 174
12.12.3.35 verticalSpacer_11 . . . 174
12.12.3.36 verticalSpacer_2 . . . 174
12.12.3.37 verticalSpacer_3 . . . 174
12.12.3.38 verticalSpacer_4 . . . 174
12.12.3.39 verticalSpacer_5 . . . 174

12.12.3.40 verticalSpacer_6 . . . 174
12.12.3.41 verticalSpacer_7 . . . 174
12.12.3.42 verticalSpacer_8 . . . 175
12.12.3.43 verticalSpacer_9 . . . 175
12.13 ExtraSettingsWidget Class Reference . . . 175
12.13.1 Detailed Description . . . 176
12.13.2 Constructor & Destructor Documentation . . . 177
12.13.2.1 ExtraSettingsWidget . . . 177
12.13.2.2 ~ExtraSettingsWidget . . . 178
12.13.3 Member Function Documentation . . . 178
12.13.3.1 on_animScanButton_clicked . . . 178
12.13.3.2 on_brightnessBox_clicked . . . 179
12.13.3.3 on_delayBox_clicked . . . 179
12.13.3.4 on_ditherBox_clicked . . . 180
12.13.3.5 on_fpsBox_valueChanged . . . 180
12.13.3.6 on_mAccelBox_clicked . . . 180
12.13.3.7 on_sAccelBox_clicked . . . 181
12.13.3.8 on_sSpeedBox_valueChanged . . . 181
12.13.3.9 on_trayBox_clicked . . . 182
12.13.3.10 pollUpdates . . . 182
12.13.4 Field Documentation . . . 183
12.13.4.1 ui . . . 183
12.14 KbFirmware::FW Struct Reference . . . 183
12.14.1 Detailed Description . . . 184
12.14.2 Constructor & Destructor Documentation . . . 184
12.14.2.1 FW . . . 184
12.14.3 Field Documentation . . . 184
12.14.3.1 ckbVersion . . . 184
12.14.3.2 fileName . . . 184
12.14.3.3 fwVersion . . . 184
12.14.3.4 hash . . . 184
12.14.3.5 url . . . 184
12.15 FwUpgradeDialog Class Reference . . . 184
12.15.1 Detailed Description . . . 186
12.15.2 Constructor & Destructor Documentation . . . 186
12.15.2.1 FwUpgradeDialog . . . 186
12.15.2.2 ~FwUpgradeDialog . . . 187
12.15.3 Member Function Documentation . . . 187
12.15.3.1 cleanBlob . . . 187
12.15.3.2 closeEvent . . . 188

12.15.3.3 exec . . . 188
12.15.3.4 fwUpdateFinished . . . 189
12.15.3.5 fwUpdateProgress . . . 190
12.15.3.6 on_actionButton_clicked . . . 190
12.15.3.7 on_cancelButton_clicked . . . 191
12.15.3.8 removeDev . . . 191
12.15.3.9 saveBlob . . . 192
12.15.4 Field Documentation . . . 192
12.15.4.1 blob . . . 193
12.15.4.2 evLoop . . . 193
12.15.4.3 exitSuccess . . . 193
12.15.4.4 kb . . . 193
12.15.4.5 savePath . . . 193
12.15.4.6 ui . . . 193
12.16 Ui::FwUpgradeDialog Class Reference . . . 193
12.16.1 Detailed Description . . . 195
12.16.2 Member Function Documentation . . . 196
12.16.2.1 retranslateUi . . . 196
12.16.2.2 setupUi . . . 196
12.16.3 Field Documentation . . . 198
12.16.3.1 actionButton . . . 198
12.16.3.2 cancelButton . . . 198
12.16.3.3 curLabel . . . 198
12.16.3.4 devLabel . . . 199
12.16.3.5 gridLayout . . . 199
12.16.3.6 gridLayout_2 . . . 199
12.16.3.7 horizontalSpacer . . . 199
12.16.3.8 horizontalSpacer_2 . . . 199
12.16.3.9 horizontalSpacer_3 . . . 199
12.16.3.10 horizontalSpacer_4 . . . 199
12.16.3.11 label_2 . . . 199
12.16.3.12 label_4 . . . 199
12.16.3.13 newLabel . . . 200
12.16.3.14 progressBar . . . 200
12.16.3.15 statusLabel . . . 200
12.17 GradientButton Class Reference . . . 200
12.17.1 Detailed Description . . . 201
12.17.2 Constructor & Destructor Documentation . . . 202
12.17.2.1 GradientButton . . . 202
12.17.3 Member Function Documentation . . . 202

12.17.3.1 fromString . . . 202
12.17.3.2 gradientChanged . . . 203
12.17.3.3 pickGradient . . . 203
12.17.3.4 toString . . . 204
12.17.3.5 updateImage . . . 205
12.17.4 Field Documentation . . . 206
12.17.4.1 _alpha . . . 206
12.17.4.2 _stops . . . 206
12.18 Ui::GradientDialog Class Reference . . . 206
12.18.1 Detailed Description . . . 208
12.18.2 Member Function Documentation . . . 208
12.18.2.1 retranslateUi . . . 208
12.18.2.2 setupUi . . . 209
12.18.3 Field Documentation . . . 211
12.18.3.1 buttonBox . . . 211
12.18.3.2 gridLayout . . . 211
12.18.3.3 gridLayout_2 . . . 212
12.18.3.4 gridLayout_3 . . . 212
12.18.3.5 groupBox_2 . . . 212
12.18.3.6 horizontalLayout . . . 212
12.18.3.7 horizontalLayout_2 . . . 212
12.18.3.8 horizontalSpacer . . . 212
12.18.3.9 horizontalSpacer_2 . . . 212
12.18.3.10 horizontalSpacer_3 . . . 212
12.18.3.11 label . . . 212
12.18.3.12 label_2 . . . 213
12.18.3.13 presetDelete . . . 213
12.18.3.14 presetList . . . 213
12.18.3.15 presetName . . . 213
12.18.3.16 presetSave . . . 213
12.18.3.17 stopBox . . . 213
12.18.3.18 stopColor . . . 213
12.18.3.19 stopOpacity . . . 213
12.18.3.20 stopPos . . . 214
12.18.3.21 verticalSpacer . . . 214
12.18.3.22 verticalSpacer_2 . . . 214
12.18.3.23 widget . . . 214
12.19 GradientDialog Class Reference . . . 214
12.19.1 Detailed Description . . . 217
12.19.2 Constructor & Destructor Documentation . . . 217

12.19.2.1 GradientDialog . . . 217
12.19.2.2 ~GradientDialog . . . 218
12.19.3 Member Function Documentation . . . 218
12.19.3.1 addPreset . . . 218
12.19.3.2 colorChanged . . . 219
12.19.3.3 currentChanged . . . 219
12.19.3.4 getGradient . . . 220
12.19.3.5 makeIcon . . . 221
12.19.3.6 on_presetDelete_clicked . . . 222
12.19.3.7 on_presetList_currentItemChanged . . . 222
12.19.3.8 on_presetName_textEdited . . . 223
12.19.3.9 on_presetSave_clicked . . . 223
12.19.3.10 on_stopOpacity_valueChanged . . . 224
12.19.3.11 on_stopPos_valueChanged . . . 224
12.19.3.12 setPreset . . . 225
12.19.3.13 updatePresets . . . 226
12.19.4 Field Documentation . . . 226
12.19.4.1 currentPreset . . . 226
12.19.4.2 presets . . . 227
12.19.4.3 ui . . . 227
12.20 GradientDialogWidget Class Reference . . . 227
12.20.1 Detailed Description . . . 230
12.20.2 Constructor & Destructor Documentation . . . 230
12.20.2.1 GradientDialogWidget . . . 230
12.20.3 Member Function Documentation . . . 230
12.20.3.1 colorAt . . . 230
12.20.3.2 current . . . 231
12.20.3.3 currentChanged . . . 231
12.20.3.4 fillRect . . . 231
12.20.3.5 makeStops . . . 232
12.20.3.6 mouseMoveEvent . . . 233
12.20.3.7 mousePressEvent . . . 234
12.20.3.8 mouseReleaseEvent . . . 235
12.20.3.9 moveCurrent . . . 235
12.20.3.10 paintEvent . . . 236
12.20.3.11 selectedStop . . . 237
12.20.3.12 selectionColors . . . 238
12.20.3.13 setCurrentColor . . . 239
12.20.3.14 setStops . . . 239
12.20.3.15 stopCount . . . 240

12.20.3.16stopRect . . . 241
12.20.3.17stops . . . 241
12.20.4 Field Documentation . . . 242
12.20.4.1 _colors . . . 242
12.20.4.2 _current . . . 242
12.20.4.3 _stops . . . 242
12.20.4.4 selected . . . 242
12.20.4.5 selectedOffset . . . 242
12.20.4.6 selectedPos . . . 242
12.21JlCompress Class Reference . . . 243
12.21.1 Detailed Description . . . 244
12.21.2 Member Function Documentation . . . 244
12.21.2.1 compressDir . . . 244
12.21.2.2 compressFile . . . 245
12.21.2.3 compressFile . . . 246
12.21.2.4 compressFiles . . . 248
12.21.2.5 compressSubDir . . . 250
12.21.2.6 extractDir . . . 252
12.21.2.7 extractFile . . . 254
12.21.2.8 extractFile . . . 256
12.21.2.9 extractFiles . . . 257
12.21.2.10getFileList . . . 258
12.21.2.11removeFile . . . 260
12.22Kb Class Reference . . . 261
12.22.1 Detailed Description . . . 266
12.22.2 Constructor & Destructor Documentation . . . 266
12.22.2.1 ∼Kb . . . 266
12.22.2.2 Kb . . . 267
12.22.3 Member Function Documentation . . . 269
12.22.3.1 appendProfile . . . 269
12.22.3.2 autoSave . . . 269
12.22.3.3 currentBind . . . 270
12.22.3.4 currentLight . . . 270
12.22.3.5 currentMode . . . 271
12.22.3.6 currentPerf . . . 271
12.22.3.7 currentProfile . . . 272
12.22.3.8 deleteHw . . . 272
12.22.3.9 deletePrevious . . . 273
12.22.3.10dither . . . 273
12.22.3.11dither . . . 274

12.22.3.12find . . . 274
12.22.3.13frameRate . . . 275
12.22.3.14frameRate . . . 275
12.22.3.15frameUpdate . . . 276
12.22.3.16fwUpdate . . . 277
12.22.3.17fwUpdateFinished . . . 278
12.22.3.18fwUpdateProgress . . . 278
12.22.3.19getKeyMap . . . 279
12.22.3.20getMacroNumber . . . 279
12.22.3.21getMacroPath . . . 280
12.22.3.22hwProfile . . . 280
12.22.3.23hwProfile . . . 280
12.22.3.24hwSave . . . 281
12.22.3.25indexOf . . . 282
12.22.3.26infoUpdated . . . 283
12.22.3.27isKeyboard . . . 283
12.22.3.28isMouse . . . 284
12.22.3.29isOpen . . . 285
12.22.3.30layout . . . 285
12.22.3.31layout . . . 286
12.22.3.32load . . . 286
12.22.3.33macroDelay . . . 287
12.22.3.34macroDelay . . . 287
12.22.3.35matches . . . 288
12.22.3.36modeChanged . . . 288
12.22.3.37model . . . 289
12.22.3.38modeRenamed . . . 289
12.22.3.39mouseAccel . . . 289
12.22.3.40mouseAccel . . . 290
12.22.3.41needsSave . . . 290
12.22.3.42newMode . . . 290
12.22.3.43newMode . . . 291
12.22.3.44newProfile . . . 292
12.22.3.45newProfile . . . 292
12.22.3.46profileAdded . . . 293
12.22.3.47profileChanged . . . 293
12.22.3.48profileRenamed . . . 294
12.22.3.49profiles . . . 294
12.22.3.50profiles . . . 295
12.22.3.51readNotify . . . 295

12.22.3.52 run . . . 299
12.22.3.53 save . . . 300
12.22.3.54 scrollSpeed . . . 301
12.22.3.55 scrollSpeed . . . 301
12.22.3.56 setCurrentMode . . . 302
12.22.3.57 setCurrentMode . . . 302
12.22.3.58 setCurrentMode . . . 303
12.22.3.59 setCurrentProfile . . . 304
12.22.3.60 updateLayout . . . 305
12.22.3.61 writeProfileHeader . . . 306
12.22.4 Friends And Related Function Documentation . . . 307
12.22.4.1 KbManager . . . 307
12.22.5 Field Documentation . . . 307
12.22.5.1 _currentMode . . . 307
12.22.5.2 _currentProfile . . . 307
12.22.5.3 _delay . . . 307
12.22.5.4 _dither . . . 307
12.22.5.5 _frameRate . . . 307
12.22.5.6 _hwProfile . . . 308
12.22.5.7 _layout . . . 308
12.22.5.8 _model . . . 308
12.22.5.9 _mouseAccel . . . 308
12.22.5.10 _needsSave . . . 308
12.22.5.11 _profiles . . . 308
12.22.5.12 _scrollSpeed . . . 308
12.22.5.13 cmd . . . 308
12.22.5.14 cmdpath . . . 308
12.22.5.15 devpath . . . 309
12.22.5.16 features . . . 309
12.22.5.17 firmware . . . 309
12.22.5.18 fwUpdPath . . . 309
12.22.5.19 hwLoading . . . 309
12.22.5.20 HWMODE_MAX . . . 309
12.22.5.21 hwModeCount . . . 309
12.22.5.22 lState . . . 309
12.22.5.23 lastAutoSave . . . 309
12.22.5.24 macroNumber . . . 310
12.22.5.25 macroPath . . . 310
12.22.5.26 monochrome . . . 310
12.22.5.27 notifyNumber . . . 310

12.22.5.28 notifyPath . . . 310
12.22.5.29 pollrate . . . 310
12.22.5.30 prefsPath . . . 310
12.22.5.31 prevMode . . . 310
12.22.5.32 prevProfile . . . 310
12.22.5.33 usbModel . . . 311
12.22.5.34 usbSerial . . . 311
12.23 KbAnim Class Reference . . . 311
12.23.1 Detailed Description . . . 314
12.23.2 Member Enumeration Documentation . . . 314
12.23.2.1 Mode . . . 314
12.23.3 Constructor & Destructor Documentation . . . 315
12.23.3.1 KbAnim . . . 315
12.23.3.2 KbAnim . . . 316
12.23.3.3 KbAnim . . . 317
12.23.4 Member Function Documentation . . . 317
12.23.4.1 blend . . . 317
12.23.4.2 catchUp . . . 319
12.23.4.3 commitParams . . . 320
12.23.4.4 effectiveParams . . . 321
12.23.4.5 guid . . . 321
12.23.4.6 hasParameter . . . 322
12.23.4.7 isActive . . . 322
12.23.4.8 isRunning . . . 322
12.23.4.9 keypress . . . 323
12.23.4.10 keys . . . 324
12.23.4.11 keys . . . 325
12.23.4.12 map . . . 325
12.23.4.13 map . . . 326
12.23.4.14 mode . . . 326
12.23.4.15 mode . . . 326
12.23.4.16 name . . . 327
12.23.4.17 name . . . 327
12.23.4.18 needsSave . . . 327
12.23.4.19 newId . . . 328
12.23.4.20 opacity . . . 328
12.23.4.21 opacity . . . 328
12.23.4.22 parameter . . . 329
12.23.4.23 parameter . . . 329
12.23.4.24 reInit . . . 330

12.23.4.25 resetParams . . . 330
12.23.4.26 save . . . 331
12.23.4.27 script . . . 332
12.23.4.28 scriptName . . . 332
12.23.4.29 stop . . . 333
12.23.4.30 trigger . . . 334
12.23.4.31 updateParams . . . 335
12.23.5 Field Documentation . . . 336
12.23.5.1 _guid . . . 336
12.23.5.2 _isActive . . . 336
12.23.5.3 _isActiveKp . . . 336
12.23.5.4 _keys . . . 336
12.23.5.5 _map . . . 336
12.23.5.6 _mode . . . 336
12.23.5.7 _name . . . 336
12.23.5.8 _needsSave . . . 337
12.23.5.9 _opacity . . . 337
12.23.5.10 _parameters . . . 337
12.23.5.11 _script . . . 337
12.23.5.12 _scriptGuid . . . 337
12.23.5.13 _scriptName . . . 337
12.23.5.14 _tempParameters . . . 337
12.23.5.15 kpRepeatMsec . . . 337
12.23.5.16 kpRepeatTime . . . 337
12.23.5.17 kpStopTime . . . 338
12.23.5.18 repeatKey . . . 338
12.23.5.19 repeatMsec . . . 338
12.23.5.20 repeatTime . . . 338
12.23.5.21 stopTime . . . 338
12.24 Ui::KbAnimWidget Class Reference . . . 338
12.24.1 Detailed Description . . . 340
12.24.2 Member Function Documentation . . . 340
12.24.2.1 retranslateUi . . . 340
12.24.2.2 setupUi . . . 341
12.24.3 Field Documentation . . . 344
12.24.3.1 aCopyLabel . . . 344
12.24.3.2 aMissingLabel . . . 344
12.24.3.3 aMissingPage . . . 344
12.24.3.4 aNameLabel . . . 344
12.24.3.5 animList . . . 344

12.24.3.6 animPage . . . 345
12.24.3.7 aVerLabel . . . 345
12.24.3.8 blendBox . . . 345
12.24.3.9 deleteButton . . . 345
12.24.3.10 gridLayout . . . 345
12.24.3.11 gridLayout_2 . . . 345
12.24.3.12 gridLayout_3 . . . 345
12.24.3.13 gridLayout_4 . . . 345
12.24.3.14 horizontalLayout . . . 345
12.24.3.15 horizontalSpacer . . . 346
12.24.3.16 horizontalSpacer_2 . . . 346
12.24.3.17 horizontalSpacer_3 . . . 346
12.24.3.18 horizontalSpacer_4 . . . 346
12.24.3.19 horizontalSpacer_5 . . . 346
12.24.3.20 keyButton . . . 346
12.24.3.21 label_5 . . . 346
12.24.3.22 label_6 . . . 346
12.24.3.23 label_7 . . . 346
12.24.3.24 nameBox . . . 347
12.24.3.25 noAnimLabel . . . 347
12.24.3.26 noSelPage . . . 347
12.24.3.27 opacityBox . . . 347
12.24.3.28 propertyButton . . . 347
12.24.3.29 selectionStack . . . 347
12.24.3.30 verticalLayout . . . 347
12.24.3.31 verticalSpacer . . . 347
12.24.3.32 verticalSpacer_2 . . . 347
12.24.3.33 verticalSpacer_3 . . . 348
12.25 KbAnimWidget Class Reference . . . 348
12.25.1 Detailed Description . . . 350
12.25.2 Constructor & Destructor Documentation . . . 350
12.25.2.1 KbAnimWidget . . . 350
12.25.2.2 ~KbAnimWidget . . . 351
12.25.3 Member Function Documentation . . . 351
12.25.3.1 addAnim . . . 351
12.25.3.2 animChanged . . . 352
12.25.3.3 clearSelection . . . 353
12.25.3.4 didUpdateSelection . . . 353
12.25.3.5 duplicateAnim . . . 354
12.25.3.6 on_animList_currentItemChanged . . . 355

12.25.3.7 on_animList_customContextMenuRequested . . . 355
12.25.3.8 on_animList_itemChanged . . . 356
12.25.3.9 on_blendBox_activated . . . 356
12.25.3.10 on_deleteButton_clicked . . . 357
12.25.3.11 on_keyButton_clicked . . . 358
12.25.3.12 on_nameBox_textEdited . . . 358
12.25.3.13 on_opacityBox_valueChanged . . . 359
12.25.3.14 on_propertyButton_clicked . . . 359
12.25.3.15 refreshList . . . 360
12.25.3.16 reorderAnims . . . 361
12.25.3.17 setCurrent . . . 362
12.25.3.18 setLight . . . 363
12.25.3.19 setSelectedKeys . . . 364
12.25.4 Field Documentation . . . 365
12.25.4.1 animations . . . 365
12.25.4.2 current . . . 365
12.25.4.3 light . . . 365
12.25.4.4 noReorder . . . 365
12.25.4.5 selectedKeys . . . 365
12.25.4.6 ui . . . 365
12.26 KbBind Class Reference . . . 366
12.26.1 Detailed Description . . . 369
12.26.2 Constructor & Destructor Documentation . . . 369
12.26.2.1 KbBind . . . 369
12.26.2.2 KbBind . . . 369
12.26.3 Member Function Documentation . . . 370
12.26.3.1 action . . . 370
12.26.3.2 bindAction . . . 371
12.26.3.3 defaultAction . . . 371
12.26.3.4 defaultAction . . . 372
12.26.3.5 devParent . . . 372
12.26.3.6 didLoad . . . 373
12.26.3.7 friendlyActionName . . . 373
12.26.3.8 friendlyName . . . 374
12.26.3.9 getMacroNumber . . . 374
12.26.3.10 getMacroPath . . . 375
12.26.3.11 globalRemap . . . 376
12.26.3.12 handleNotificationChannel . . . 376
12.26.3.13 isISO . . . 377
12.26.3.14 isKeyboard . . . 378

12.26.3.15 isMouse . . . 378
12.26.3.16 keyEvent . . . 379
12.26.3.17 layoutChanged . . . 380
12.26.3.18 light . . . 381
12.26.3.19 load . . . 382
12.26.3.20 loadGlobalRemap . . . 383
12.26.3.21 map . . . 383
12.26.3.22 map . . . 384
12.26.3.23 modeParent . . . 384
12.26.3.24 needsSave . . . 385
12.26.3.25 noAction . . . 385
12.26.3.26 noAction . . . 385
12.26.3.27 perf . . . 386
12.26.3.28 resetAction . . . 387
12.26.3.29 resetAction . . . 387
12.26.3.30 save . . . 388
12.26.3.31 saveGlobalRemap . . . 389
12.26.3.32 setAction . . . 390
12.26.3.33 setAction . . . 391
12.26.3.34 setGlobalRemap . . . 391
12.26.3.35 setNeedsUpdate . . . 392
12.26.3.36 update . . . 392
12.26.3.37 updated . . . 394
12.26.3.38 winLock . . . 394
12.26.3.39 winLock . . . 395
12.26.4 Friends And Related Function Documentation . . . 395
12.26.4.1 KeyAction . . . 395
12.26.5 Field Documentation . . . 395
12.26.5.1 _bind . . . 395
12.26.5.2 _devParent . . . 395
12.26.5.3 _globalRemap . . . 395
12.26.5.4 _map . . . 395
12.26.5.5 _needsSave . . . 396
12.26.5.6 _needsUpdate . . . 396
12.26.5.7 _winLock . . . 396
12.26.5.8 globalRemapTime . . . 396
12.26.5.9 lastCmd . . . 396
12.26.5.10 lastGlobalRemapTime . . . 396
12.27 Ui::KbBindWidget Class Reference . . . 396
12.27.1 Detailed Description . . . 398

12.27.2 Member Function Documentation . . . 398
12.27.2.1 retranslateUi . . . 398
12.27.2.2 setupUi . . . 399
12.27.3 Field Documentation . . . 401
12.27.3.1 copyButton . . . 401
12.27.3.2 gridLayout . . . 401
12.27.3.3 gridLayout_2 . . . 401
12.27.3.4 horizontalLayout_2 . . . 401
12.27.3.5 horizontalSpacer . . . 401
12.27.3.6 horizontalSpacer_2 . . . 401
12.27.3.7 keyWidget . . . 401
12.27.3.8 label . . . 402
12.27.3.9 label_2 . . . 402
12.27.3.10line . . . 402
12.27.3.11line_2 . . . 402
12.27.3.12rbWidget . . . 402
12.27.3.13resetButton . . . 402
12.27.3.14selectLabel . . . 402
12.27.3.15verticalLayout . . . 402
12.27.3.16verticalSpacer . . . 402
12.27.3.17widget . . . 403
12.28KbBindWidget Class Reference . . . 403
12.28.1 Detailed Description . . . 404
12.28.2 Constructor & Destructor Documentation . . . 405
12.28.2.1 KbBindWidget . . . 405
12.28.2.2 ∼KbBindWidget . . . 405
12.28.3 Member Function Documentation . . . 406
12.28.3.1 newLayout . . . 406
12.28.3.2 newSelection . . . 407
12.28.3.3 on_copyButton_clicked . . . 408
12.28.3.4 on_resetButton_clicked . . . 409
12.28.3.5 setBind . . . 409
12.28.3.6 updateBind . . . 411
12.28.3.7 updateSelDisplay . . . 412
12.28.4 Field Documentation . . . 413
12.28.4.1 bind . . . 413
12.28.4.2 currentSelection . . . 413
12.28.4.3 profile . . . 413
12.28.4.4 ui . . . 413
12.29KbFirmware Class Reference . . . 413

12.29.1 Detailed Description . . . 416
12.29.2 Member Enumeration Documentation . . . 416
12.29.2.1 anonymous enum . . . 416
12.29.3 Constructor & Destructor Documentation . . . 416
12.29.3.1 KbFirmware . . . 416
12.29.3.2 ~KbFirmware . . . 416
12.29.4 Member Function Documentation . . . 416
12.29.4.1 _checkUpdates . . . 416
12.29.4.2 _fileForBoard . . . 417
12.29.4.3 _latestForBoard . . . 418
12.29.4.4 checkUpdates . . . 419
12.29.4.5 dataForBoard . . . 419
12.29.4.6 downloaded . . . 420
12.29.4.7 downloadFinished . . . 420
12.29.4.8 hasDownloaded . . . 421
12.29.4.9 processDownload . . . 421
12.29.4.10 versionForBoard . . . 423
12.29.5 Field Documentation . . . 424
12.29.5.1 fwTable . . . 424
12.29.5.2 fwTableHash . . . 424
12.29.5.3 hasGPG . . . 424
12.29.5.4 instance . . . 424
12.29.5.5 lastCheck . . . 424
12.29.5.6 lastFinished . . . 424
12.29.5.7 networkManager . . . 424
12.29.5.8 tableDownload . . . 425
12.30 KbLight Class Reference . . . 425
12.30.1 Detailed Description . . . 428
12.30.2 Member Typedef Documentation . . . 429
12.30.2.1 AnimList . . . 429
12.30.3 Constructor & Destructor Documentation . . . 429
12.30.3.1 KbLight . . . 429
12.30.3.2 KbLight . . . 429
12.30.3.3 ~KbLight . . . 430
12.30.4 Member Function Documentation . . . 430
12.30.4.1 addAnim . . . 430
12.30.4.2 animKeypress . . . 431
12.30.4.3 animList . . . 432
12.30.4.4 animList . . . 432
12.30.4.5 base . . . 432

12.30.4.6 close . . . 433
12.30.4.7 color . . . 434
12.30.4.8 color . . . 435
12.30.4.9 colorMap . . . 436
12.30.4.10 didLoad . . . 436
12.30.4.11 dimming . . . 436
12.30.4.12 dimming . . . 437
12.30.4.13 duplicateAnim . . . 437
12.30.4.14 findAnim . . . 438
12.30.4.15 findAnimIdx . . . 439
12.30.4.16 frameDisplayed . . . 439
12.30.4.17 frameUpdate . . . 440
12.30.4.18 isStarted . . . 441
12.30.4.19 load . . . 442
12.30.4.20 map . . . 443
12.30.4.21 map . . . 444
12.30.4.22 needsSave . . . 445
12.30.4.23 open . . . 445
12.30.4.24 previewAnim . . . 446
12.30.4.25 printRGB . . . 447
12.30.4.26 rebuildBaseMap . . . 448
12.30.4.27 resetIndicators . . . 448
12.30.4.28 restartAnimation . . . 449
12.30.4.29 save . . . 450
12.30.4.30 setIndicator . . . 451
12.30.4.31 shareDimming . . . 451
12.30.4.32 shareDimming . . . 452
12.30.4.33 stopPreview . . . 452
12.30.4.34 updated . . . 453
12.30.5 Field Documentation . . . 453
12.30.5.1 _animList . . . 453
12.30.5.2 _animMap . . . 453
12.30.5.3 _colorMap . . . 454
12.30.5.4 _dimming . . . 454
12.30.5.5 _indicatorList . . . 454
12.30.5.6 _indicatorMap . . . 454
12.30.5.7 _lastFrameAnimMap . . . 454
12.30.5.8 _lastFrameDimming . . . 454
12.30.5.9 _lastFrameIndicatorMap . . . 454
12.30.5.10 _map . . . 454

12.30.5.11 _needsMapRefresh . . . 454
12.30.5.12 _needsSave . . . 455
12.30.5.13 _previewAnim . . . 455
12.30.5.14 _qColorMap . . . 455
12.30.5.15 _start . . . 455
12.30.5.16 lastFrameSignal . . . 455
12.30.5.17 MAX_DIM . . . 455
12.31 Ui::KbLightWidget Class Reference . . . 455
12.31.1 Detailed Description . . . 457
12.31.2 Member Function Documentation . . . 457
12.31.2.1 retranslateUi . . . 457
12.31.2.2 setupUi . . . 458
12.31.3 Field Documentation . . . 461
12.31.3.1 animButton . . . 461
12.31.3.2 animWidget . . . 461
12.31.3.3 bgButton . . . 461
12.31.3.4 brightnessBox . . . 461
12.31.3.5 gridLayout . . . 461
12.31.3.6 gridLayout_2 . . . 461
12.31.3.7 gridLayout_4 . . . 461
12.31.3.8 horizontalLayout . . . 462
12.31.3.9 horizontalLayout_2 . . . 462
12.31.3.10 horizontalLayout_3 . . . 462
12.31.3.11 horizontalSpacer_3 . . . 462
12.31.3.12 horizontalSpacer_4 . . . 462
12.31.3.13 horizontalSpacer_5 . . . 462
12.31.3.14 horizontalSpacer_6 . . . 462
12.31.3.15 keyWidget . . . 462
12.31.3.16 label . . . 463
12.31.3.17 label_2 . . . 463
12.31.3.18 label_3 . . . 463
12.31.3.19 line . . . 463
12.31.3.20 line_2 . . . 463
12.31.3.21 selLabel . . . 463
12.31.3.22 showAnimBox . . . 463
12.31.3.23 verticalLayout . . . 463
12.31.3.24 verticalSpacer_2 . . . 463
12.31.3.25 verticalSpacer_3 . . . 464
12.31.3.26 widget . . . 464
12.32 KbLightWidget Class Reference . . . 464

12.32.1 Detailed Description . . . 466
12.32.2 Constructor & Destructor Documentation . . . 466
12.32.2.1 KbLightWidget . . . 466
12.32.2.2 ~KbLightWidget . . . 467
12.32.3 Member Function Documentation . . . 467
12.32.3.1 changeAnim . . . 467
12.32.3.2 changeAnimKeys . . . 467
12.32.3.3 changeColor . . . 468
12.32.3.4 newSelection . . . 469
12.32.3.5 on_animButton_clicked . . . 470
12.32.3.6 on_bgButton_clicked . . . 471
12.32.3.7 on_brightnessBox_activated . . . 471
12.32.3.8 on_showAnimBox_clicked . . . 472
12.32.3.9 setLight . . . 472
12.32.3.10 setMonochrome . . . 474
12.32.3.11 toggleSidelight . . . 474
12.32.3.12 updateLight . . . 475
12.32.4 Field Documentation . . . 476
12.32.4.1 currentSelection . . . 476
12.32.4.2 light . . . 476
12.32.4.3 ui . . . 476
12.33 KbManager Class Reference . . . 477
12.33.1 Detailed Description . . . 479
12.33.2 Constructor & Destructor Documentation . . . 479
12.33.2.1 KbManager . . . 479
12.33.3 Member Function Documentation . . . 480
12.33.3.1 ckbDaemonVersion . . . 480
12.33.3.2 ckbDaemonVersionF . . . 480
12.33.3.3 ckbGuiVersion . . . 481
12.33.3.4 ckbGuiVersionF . . . 481
12.33.3.5 devices . . . 482
12.33.3.6 eventTimer . . . 482
12.33.3.7 fps . . . 482
12.33.3.8 init . . . 483
12.33.3.9 kbConnected . . . 484
12.33.3.10 kbDisconnected . . . 484
12.33.3.11 kbManager . . . 484
12.33.3.12 parseVersionString . . . 485
12.33.3.13 scanKeyboards . . . 485
12.33.3.14 scanTimer . . . 487

12.33.3.15 stop . . . 487
12.33.3.16 versionUpdated . . . 488
12.33.4 Field Documentation . . . 488
12.33.4.1 _daemonVersion . . . 488
12.33.4.2 _devices . . . 488
12.33.4.3 _eventTimer . . . 488
12.33.4.4 _guiVersion . . . 489
12.33.4.5 _kbManager . . . 489
12.33.4.6 _scanTimer . . . 489
12.34 KbMode Class Reference . . . 489
12.34.1 Detailed Description . . . 492
12.34.2 Constructor & Destructor Documentation . . . 492
12.34.2.1 KbMode . . . 492
12.34.2.2 KbMode . . . 493
12.34.2.3 KbMode . . . 493
12.34.2.4 ~KbMode . . . 494
12.34.3 Member Function Documentation . . . 494
12.34.3.1 bind . . . 494
12.34.3.2 doUpdate . . . 495
12.34.3.3 id . . . 495
12.34.3.4 id . . . 496
12.34.3.5 keyMap . . . 496
12.34.3.6 light . . . 497
12.34.3.7 name . . . 498
12.34.3.8 name . . . 498
12.34.3.9 needsSave . . . 498
12.34.3.10 newId . . . 499
12.34.3.11 perf . . . 500
12.34.3.12 save . . . 500
12.34.3.13 setNeedsSave . . . 501
12.34.3.14 setNeedsUpdate . . . 502
12.34.3.15 updated . . . 502
12.34.4 Field Documentation . . . 503
12.34.4.1 _bind . . . 503
12.34.4.2 _id . . . 503
12.34.4.3 _light . . . 503
12.34.4.4 _name . . . 503
12.34.4.5 _needsSave . . . 503
12.34.4.6 _perf . . . 503
12.35 KbPerf Class Reference . . . 503

12.35.1 Detailed Description . . . 507
12.35.2 Member Enumeration Documentation . . . 507
12.35.2.1 height . . . 507
12.35.2.2 i_hw . . . 508
12.35.2.3 indicator . . . 508
12.35.3 Constructor & Destructor Documentation . . . 508
12.35.3.1 KbPerf . . . 508
12.35.3.2 KbPerf . . . 509
12.35.4 Member Function Documentation . . . 510
12.35.4.1 _curDpi . . . 510
12.35.4.2 angleSnap . . . 510
12.35.4.3 angleSnap . . . 511
12.35.4.4 applyIndicators . . . 511
12.35.4.5 baseDpiIdx . . . 512
12.35.4.6 baseDpiIdx . . . 513
12.35.4.7 bind . . . 513
12.35.4.8 didLoad . . . 514
12.35.4.9 dpi . . . 514
12.35.4.10 dpi . . . 515
12.35.4.11 dpi . . . 516
12.35.4.12 dpiColor . . . 516
12.35.4.13 dpiColor . . . 517
12.35.4.14 dpiCycleDown . . . 517
12.35.4.15 dpiCycleUp . . . 518
12.35.4.16 dpiDown . . . 518
12.35.4.17 dpiEnabled . . . 519
12.35.4.18 dpiEnabled . . . 519
12.35.4.19 dpiIndicator . . . 520
12.35.4.20 dpiIndicator . . . 520
12.35.4.21 dpiUp . . . 520
12.35.4.22 getIndicator . . . 521
12.35.4.23 iOpacity . . . 521
12.35.4.24 iOpacity . . . 522
12.35.4.25 liftHeight . . . 522
12.35.4.26 liftHeight . . . 522
12.35.4.27 light . . . 523
12.35.4.28 lightIndicator . . . 523
12.35.4.29 load . . . 524
12.35.4.30 modeParent . . . 526
12.35.4.31 needsSave . . . 526

12.35.4.32 operator= . . . 527
12.35.4.33 popDpi . . . 527
12.35.4.34 pushDpi . . . 528
12.35.4.35 pushDpi . . . 529
12.35.4.36 pushSniper . . . 530
12.35.4.37 save . . . 530
12.35.4.38 setIndicator . . . 531
12.35.4.39 setNeedsUpdate . . . 532
12.35.4.40 settingsUpdated . . . 532
12.35.4.41 sniperDpi . . . 533
12.35.4.42 sniperDpi . . . 533
12.35.4.43 sniperDpi . . . 533
12.35.4.44 update . . . 534
12.35.5 Field Documentation . . . 535
12.35.5.1 _angleSnap . . . 535
12.35.5.2 _dpiIndicator . . . 535
12.35.5.3 _iOpacity . . . 536
12.35.5.4 _liftHeight . . . 536
12.35.5.5 _needsSave . . . 536
12.35.5.6 _needsUpdate . . . 536
12.35.5.7 DPI_COUNT . . . 536
12.35.5.8 DPI_MAX . . . 536
12.35.5.9 DPI_MIN . . . 536
12.35.5.10 dpiBaseIdx . . . 536
12.35.5.11 dpiClr . . . 536
12.35.5.12 dpiCurX . . . 537
12.35.5.13 dpiCurY . . . 537
12.35.5.14 dpiOn . . . 537
12.35.5.15 dpiX . . . 537
12.35.5.16 dpiY . . . 537
12.35.5.17 HW_I_COUNT . . . 537
12.35.5.18 hwIType . . . 537
12.35.5.19 I_COUNT . . . 537
12.35.5.20 iColor . . . 537
12.35.5.21 iEnable . . . 538
12.35.5.22 light100Color . . . 538
12.35.5.23 muteNAColor . . . 538
12.35.5.24 OTHER . . . 538
12.35.5.25 pushedDpis . . . 538
12.35.5.26 runningPushIdx . . . 538

12.35.5.27 SNIPER . . . 538
12.36 KbProfile Class Reference . . . 538
12.36.1 Detailed Description . . . 541
12.36.2 Member Typedef Documentation . . . 541
12.36.2.1 ModeList . . . 541
12.36.3 Constructor & Destructor Documentation . . . 541
12.36.3.1 KbProfile . . . 541
12.36.3.2 KbProfile . . . 542
12.36.3.3 KbProfile . . . 542
12.36.4 Member Function Documentation . . . 543
12.36.4.1 append . . . 543
12.36.4.2 currentMode . . . 543
12.36.4.3 currentMode . . . 544
12.36.4.4 find . . . 544
12.36.4.5 id . . . 545
12.36.4.6 id . . . 545
12.36.4.7 indexOf . . . 546
12.36.4.8 insert . . . 546
12.36.4.9 keyMap . . . 547
12.36.4.10 keyMap . . . 547
12.36.4.11 modeCount . . . 548
12.36.4.12 modes . . . 548
12.36.4.13 modes . . . 549
12.36.4.14 move . . . 549
12.36.4.15 name . . . 550
12.36.4.16 name . . . 550
12.36.4.17 needsSave . . . 551
12.36.4.18 newId . . . 551
12.36.4.19 removeAll . . . 552
12.36.4.20 save . . . 553
12.36.4.21 setNeedsSave . . . 554
12.36.4.22 setNeedsUpdate . . . 554
12.36.5 Field Documentation . . . 555
12.36.5.1 _currentMode . . . 555
12.36.5.2 _id . . . 555
12.36.5.3 _keyMap . . . 555
12.36.5.4 _modes . . . 555
12.36.5.5 _name . . . 556
12.36.5.6 _needsSave . . . 556
12.37 KbProfileDialog Class Reference . . . 556

12.37.1 Detailed Description . . . 557
12.37.2 Constructor & Destructor Documentation . . . 558
12.37.2.1 KbProfileDialog . . . 558
12.37.2.2 ~KbProfileDialog . . . 558
12.37.3 Member Function Documentation . . . 558
12.37.3.1 addNewProfileItem . . . 558
12.37.3.2 on_profileList_currentItemChanged . . . 559
12.37.3.3 on_profileList_customContextMenuRequested . . . 559
12.37.3.4 on_profileList_itemChanged . . . 561
12.37.3.5 on_profileList_itemClicked . . . 562
12.37.3.6 profileList_reordered . . . 563
12.37.3.7 repopulate . . . 563
12.37.4 Field Documentation . . . 564
12.37.4.1 device . . . 564
12.37.4.2 GUID . . . 565
12.37.4.3 NEW_FLAG . . . 565
12.37.4.4 ui . . . 565
12.38 Ui::KbProfileDialog Class Reference . . . 565
12.38.1 Detailed Description . . . 567
12.38.2 Member Function Documentation . . . 567
12.38.2.1 retranslateUi . . . 567
12.38.2.2 setupUi . . . 568
12.38.3 Field Documentation . . . 569
12.38.3.1 buttonBox . . . 569
12.38.3.2 gridLayout . . . 569
12.38.3.3 horizontalSpacer . . . 569
12.38.3.4 horizontalSpacer_2 . . . 569
12.38.3.5 profileList . . . 569
12.39 KbWidget Class Reference . . . 570
12.39.1 Detailed Description . . . 572
12.39.2 Constructor & Destructor Documentation . . . 572
12.39.2.1 KbWidget . . . 572
12.39.2.2 ~KbWidget . . . 574
12.39.3 Member Function Documentation . . . 574
12.39.3.1 addNewModeItem . . . 574
12.39.3.2 devUpdate . . . 575
12.39.3.3 modeChanged . . . 575
12.39.3.4 modeIcon . . . 577
12.39.3.5 modesList_reordered . . . 578
12.39.3.6 modeUpdate . . . 579

12.39.3.7 name . . . 579
12.39.3.8 on_fwUpdButton_clicked . . . 580
12.39.3.9 on_hwSaveButton_clicked . . . 581
12.39.3.10 on_modesList_currentItemChanged . . . 582
12.39.3.11 on_modesList_customContextMenuRequested . . . 583
12.39.3.12 on_modesList_itemChanged . . . 585
12.39.3.13 on_modesList_itemClicked . . . 585
12.39.3.14 on_profileBox_activated . . . 586
12.39.3.15 on_tabWidget_currentChanged . . . 586
12.39.3.16 profileChanged . . . 587
12.39.3.17 showFirstTab . . . 588
12.39.3.18 showFwUpdate . . . 588
12.39.3.19 showLastTab . . . 589
12.39.3.20 updateFwButton . . . 589
12.39.3.21 updateProfileList . . . 590
12.39.4 Field Documentation . . . 591
12.39.4.1 currentMode . . . 591
12.39.4.2 device . . . 591
12.39.4.3 GUID . . . 592
12.39.4.4 hasShownNewFW . . . 592
12.39.4.5 lastAutoSave . . . 592
12.39.4.6 NEW_FLAG . . . 592
12.39.4.7 ui . . . 592
12.40 Ui::KbWidget Class Reference . . . 592
12.40.1 Detailed Description . . . 594
12.40.2 Member Function Documentation . . . 594
12.40.2.1 retranslateUi . . . 594
12.40.2.2 setupUi . . . 595
12.40.3 Field Documentation . . . 598
12.40.3.1 bindTab . . . 598
12.40.3.2 bindWidget . . . 598
12.40.3.3 devTab . . . 598
12.40.3.4 fwLabel . . . 598
12.40.3.5 fwUpdButton . . . 599
12.40.3.6 fwUpdLabel . . . 599
12.40.3.7 fwUpdLayout . . . 599
12.40.3.8 gridLayout . . . 599
12.40.3.9 gridLayout_2 . . . 599
12.40.3.10 gridLayout_3 . . . 599
12.40.3.11 gridLayout_4 . . . 599

12.40.3.12 gridLayout_5 . . . 599
12.40.3.13 gridLayout_7 . . . 599
12.40.3.14 horizontalSpacer_2 . . . 600
12.40.3.15 horizontalSpacer_3 . . . 600
12.40.3.16 hwSaveButton . . . 600
12.40.3.17 kPerfTab . . . 600
12.40.3.18 kPerfWidget . . . 600
12.40.3.19 label_3 . . . 600
12.40.3.20 label_4 . . . 600
12.40.3.21 label_9 . . . 600
12.40.3.22 lightTab . . . 600
12.40.3.23 lightWidget . . . 601
12.40.3.24 modesList . . . 601
12.40.3.25 mPerfTab . . . 601
12.40.3.26 mPerfWidget . . . 601
12.40.3.27 pollLabel . . . 601
12.40.3.28 pollLabel2 . . . 601
12.40.3.29 profileBox . . . 601
12.40.3.30 serialLabel . . . 601
12.40.3.31 tabWidget . . . 601
12.40.3.32 verticalSpacer . . . 602
12.41 Key Struct Reference . . . 602
12.41.1 Detailed Description . . . 603
12.41.2 Member Function Documentation . . . 603
12.41.2.1 friendlyName . . . 603
12.41.2.2 operator bool . . . 603
12.41.2.3 operator! . . . 604
12.41.2.4 storageName . . . 604
12.41.3 Field Documentation . . . 604
12.41.3.1 _friendlyName . . . 604
12.41.3.2 _storageName . . . 604
12.41.3.3 hasLed . . . 604
12.41.3.4 hasScan . . . 604
12.41.3.5 height . . . 605
12.41.3.6 name . . . 605
12.41.3.7 width . . . 605
12.41.3.8 x . . . 605
12.41.3.9 y . . . 605
12.42 KeyAction Class Reference . . . 605
12.42.1 Detailed Description . . . 609

12.42.2 Member Enumeration Documentation . . . 609
12.42.2.1 Type . . . 609
12.42.3 Constructor & Destructor Documentation . . . 610
12.42.3.1 KeyAction . . . 610
12.42.3.2 KeyAction . . . 610
12.42.3.3 ~KeyAction . . . 610
12.42.3.4 KeyAction . . . 610
12.42.4 Member Function Documentation . . . 610
12.42.4.1 adjustDisplay . . . 610
12.42.4.2 animAction . . . 611
12.42.4.3 animInfo . . . 612
12.42.4.4 defaultAction . . . 612
12.42.4.5 dpiAction . . . 613
12.42.4.6 dpiInfo . . . 614
12.42.4.7 driverName . . . 614
12.42.4.8 friendlyName . . . 615
12.42.4.9 isAnim . . . 616
12.42.4.10 isDPI . . . 617
12.42.4.11 isMacro . . . 617
12.42.4.12 isMedia . . . 617
12.42.4.13 isMouse . . . 618
12.42.4.14 isNormal . . . 618
12.42.4.15 isProgram . . . 619
12.42.4.16 isSpecial . . . 619
12.42.4.17 isUnbound . . . 620
12.42.4.18 isValidMacro . . . 620
12.42.4.19 keyEvent . . . 621
12.42.4.20 keyRelease . . . 625
12.42.4.21 lightAction . . . 625
12.42.4.22 lockAction . . . 625
12.42.4.23 macroAction . . . 625
12.42.4.24 macroContent . . . 626
12.42.4.25 macroDisplay . . . 627
12.42.4.26 macroFullLine . . . 627
12.42.4.27 macroLine . . . 628
12.42.4.28 macroTiming . . . 629
12.42.4.29 modeAction . . . 629
12.42.4.30 noAction . . . 630
12.42.4.31 operator QString . . . 630
12.42.4.32 operator= . . . 630

12.42.4.33 programAction . . . 630
12.42.4.34 programInfo . . . 630
12.42.4.35 specialInfo . . . 631
12.42.4.36 type . . . 632
12.42.4.37 value . . . 632
12.42.5 Field Documentation . . . 632
12.42.5.1 _value . . . 632
12.42.5.2 DPI_CUSTOM . . . 633
12.42.5.3 DPI_CYCLE_DOWN . . . 633
12.42.5.4 DPI_CYCLE_UP . . . 633
12.42.5.5 DPI_DOWN . . . 633
12.42.5.6 DPI_SNIPER . . . 633
12.42.5.7 DPI_UP . . . 633
12.42.5.8 LIGHT_DOWN . . . 633
12.42.5.9 LIGHT_DOWN_WRAP . . . 633
12.42.5.10 LIGHT_UP . . . 633
12.42.5.11 LIGHT_UP_WRAP . . . 634
12.42.5.12 LOCK_OFF . . . 634
12.42.5.13 LOCK_ON . . . 634
12.42.5.14 LOCK_TOGGLE . . . 634
12.42.5.15 MODE_NEXT . . . 634
12.42.5.16 MODE_NEXT_WRAP . . . 634
12.42.5.17 MODE_PREV . . . 634
12.42.5.18 MODE_PREV_WRAP . . . 634
12.42.5.19 preProgram . . . 634
12.42.5.20 PROGRAM_PR_INDEF . . . 635
12.42.5.21 PROGRAM_PR_KPSTOP . . . 635
12.42.5.22 PROGRAM_PR_KRSTOP . . . 635
12.42.5.23 PROGRAM_PR_MULTI . . . 635
12.42.5.24 PROGRAM_RE_INDEF . . . 635
12.42.5.25 PROGRAM_RE_KPSTOP . . . 635
12.42.5.26 PROGRAM_RE_MULTI . . . 635
12.42.5.27 relProgram . . . 635
12.42.5.28 sniperValue . . . 635
12.43 KeyMap Class Reference . . . 636
12.43.1 Detailed Description . . . 638
12.43.2 Member Enumeration Documentation . . . 638
12.43.2.1 Layout . . . 638
12.43.2.2 Model . . . 639
12.43.3 Constructor & Destructor Documentation . . . 639

12.43.3.1 KeyMap . . . 639
12.43.3.2 KeyMap . . . 640
12.43.4 Member Function Documentation . . . 640
12.43.4.1 byPosition . . . 640
12.43.4.2 contains . . . 641
12.43.4.3 count . . . 641
12.43.4.4 friendlyName . . . 641
12.43.4.5 fromName . . . 643
12.43.4.6 fromStorage . . . 644
12.43.4.7 getLayout . . . 644
12.43.4.8 getLayout . . . 645
12.43.4.9 getModel . . . 646
12.43.4.10 getModel . . . 646
12.43.4.11 height . . . 647
12.43.4.12 isISO . . . 647
12.43.4.13 isISO . . . 648
12.43.4.14 isJP . . . 648
12.43.4.15 isJP . . . 648
12.43.4.16 isKeyboard . . . 649
12.43.4.17 isKeyboard . . . 650
12.43.4.18 isMouse . . . 650
12.43.4.19 isMouse . . . 651
12.43.4.20 isPTBR . . . 652
12.43.4.21 isPTBR . . . 652
12.43.4.22 key . . . 652
12.43.4.23 keys . . . 653
12.43.4.24 layout . . . 653
12.43.4.25 layoutNames . . . 654
12.43.4.26 locale . . . 654
12.43.4.27 map . . . 655
12.43.4.28 model . . . 656
12.43.4.29 modelHeight . . . 656
12.43.4.30 modelWidth . . . 656
12.43.4.31 name . . . 657
12.43.4.32 operator const QHash< QString, Key > & . . . 657
12.43.4.33 operator[] . . . 658
12.43.4.34 positions . . . 658
12.43.4.35 strLayout . . . 658
12.43.4.36 strModel . . . 659
12.43.4.37 toStorage . . . 659

12.43.4.38 width . . . 660
12.43.5 Field Documentation . . . 660
12.43.5.1 _keys . . . 660
12.43.5.2 keyHeight . . . 660
12.43.5.3 keyLayout . . . 661
12.43.5.4 keyModel . . . 661
12.43.5.5 keyWidth . . . 661
12.44 KeyWidget Class Reference . . . 661
12.44.1 Detailed Description . . . 664
12.44.2 Member Typedef Documentation . . . 664
12.44.2.1 BindMap . . . 664
12.44.3 Member Enumeration Documentation . . . 665
12.44.3.1 anonymous enum . . . 665
12.44.4 Constructor & Destructor Documentation . . . 665
12.44.4.1 KeyWidget . . . 665
12.44.5 Member Function Documentation . . . 665
12.44.5.1 bindMap . . . 665
12.44.5.2 bindMap . . . 666
12.44.5.3 clearAnimation . . . 666
12.44.5.4 clearSelection . . . 666
12.44.5.5 colorMap . . . 667
12.44.5.6 colorMap . . . 667
12.44.5.7 displayColorMap . . . 667
12.44.5.8 drawInfo . . . 668
12.44.5.9 map . . . 668
12.44.5.10 map . . . 669
12.44.5.11 monochrome . . . 669
12.44.5.12 monochrome . . . 670
12.44.5.13 mouseMoveEvent . . . 670
12.44.5.14 mousePressEvent . . . 671
12.44.5.15 mouseReleaseEvent . . . 672
12.44.5.16 paintEvent . . . 673
12.44.5.17 rgbMode . . . 678
12.44.5.18 rgbMode . . . 679
12.44.5.19 selectAll . . . 679
12.44.5.20 selectionChanged . . . 680
12.44.5.21 setAnimation . . . 680
12.44.5.22 setAnimationToSelection . . . 681
12.44.5.23 setSelection . . . 681
12.44.5.24 sidelightToggled . . . 682

12.44.6 Field Documentation . . . 682
12.44.6.1 _bindMap . . . 682
12.44.6.2 _colorMap . . . 683
12.44.6.3 _displayColorMap . . . 683
12.44.6.4 _indicators . . . 683
12.44.6.5 _monochrome . . . 683
12.44.6.6 _rgbMode . . . 683
12.44.6.7 animation . . . 683
12.44.6.8 keyMap . . . 683
12.44.6.9 mouseCurrentX . . . 683
12.44.6.10 mouseCurrentY . . . 683
12.44.6.11 mouseDownMode . . . 684
12.44.6.12 mouseDownX . . . 684
12.44.6.13 mouseDownY . . . 684
12.44.6.14 newSelection . . . 684
12.44.6.15 selection . . . 684
12.45 Ui::KPerfWidget Class Reference . . . 684
12.45.1 Detailed Description . . . 688
12.45.2 Member Function Documentation . . . 688
12.45.2.1 retranslateUi . . . 688
12.45.2.2 setupUi . . . 689
12.45.3 Field Documentation . . . 696
12.45.3.1 capsBox . . . 696
12.45.3.2 capsColorOff . . . 696
12.45.3.3 capsColorOn . . . 696
12.45.3.4 copyButton . . . 696
12.45.3.5 gridLayout . . . 697
12.45.3.6 horizontalLayout . . . 697
12.45.3.7 horizontalLayout_2 . . . 697
12.45.3.8 horizontalSpacer . . . 697
12.45.3.9 horizontalSpacer_2 . . . 697
12.45.3.10 horizontalSpacer_3 . . . 697
12.45.3.11 horizontalSpacer_4 . . . 697
12.45.3.12 horizontalSpacer_5 . . . 697
12.45.3.13 horizontalSpacer_6 . . . 697
12.45.3.14 horizontalSpacer_7 . . . 698
12.45.3.15 horizontalSpacer_8 . . . 698
12.45.3.16 intensityBox . . . 698
12.45.3.17 k95Label1 . . . 698
12.45.3.18 k95Label2 . . . 698

12.45.3.19 k95Label3 . . . 698
12.45.3.20 k95Label4 . . . 698
12.45.3.21 k95Label5 . . . 698
12.45.3.22 k95Label6 . . . 698
12.45.3.23 k95Line . . . 699
12.45.3.24 k95Spacer . . . 699
12.45.3.25 label . . . 699
12.45.3.26 label_10 . . . 699
12.45.3.27 label_11 . . . 699
12.45.3.28 label_12 . . . 699
12.45.3.29 label_14 . . . 699
12.45.3.30 label_15 . . . 699
12.45.3.31 label_16 . . . 699
12.45.3.32 label_17 . . . 700
12.45.3.33 label_18 . . . 700
12.45.3.34 label_19 . . . 700
12.45.3.35 label_2 . . . 700
12.45.3.36 label_20 . . . 700
12.45.3.37 label_21 . . . 700
12.45.3.38 label_22 . . . 700
12.45.3.39 label_23 . . . 700
12.45.3.40 label_24 . . . 700
12.45.3.41 label_3 . . . 701
12.45.3.42 label_4 . . . 701
12.45.3.43 label_5 . . . 701
12.45.3.44 label_6 . . . 701
12.45.3.45 label_7 . . . 701
12.45.3.46 label_8 . . . 701
12.45.3.47 lightBox . . . 701
12.45.3.48 lightColor1 . . . 701
12.45.3.49 lightColor2 . . . 701
12.45.3.50 lightColor3 . . . 702
12.45.3.51 line . . . 702
12.45.3.52 line_2 . . . 702
12.45.3.53 line_3 . . . 702
12.45.3.54 lockBox . . . 702
12.45.3.55 lockColorOff . . . 702
12.45.3.56 lockColorOn . . . 702
12.45.3.57 macroBox . . . 702
12.45.3.58 macroColorOff . . . 702

12.45.3.59 macroColorOn . . . 703
12.45.3.60 modeBox . . . 703
12.45.3.61 modeColorOff . . . 703
12.45.3.62 modeColorOn . . . 703
12.45.3.63 muteBox . . . 703
12.45.3.64 muteColorNA . . . 703
12.45.3.65 muteColorOff . . . 703
12.45.3.66 muteColorOn . . . 703
12.45.3.67 numBox . . . 703
12.45.3.68 numColorOff . . . 704
12.45.3.69 numColorOn . . . 704
12.45.3.70 scrollBox . . . 704
12.45.3.71 scrollColorOff . . . 704
12.45.3.72 scrollColorOn . . . 704
12.45.3.73 verticalSpacer . . . 704
12.45.3.74 verticalSpacer_2 . . . 704
12.45.3.75 verticalSpacer_3 . . . 704
12.45.3.76 verticalSpacer_4 . . . 704
12.46 KPerfWidget Class Reference . . . 705
12.46.1 Detailed Description . . . 707
12.46.2 Data Structure Documentation . . . 707
12.46.2.1 struct KPerfWidget::IndicatorUi . . . 707
12.46.3 Member Typedef Documentation . . . 708
12.46.3.1 i_hw . . . 709
12.46.3.2 indicator . . . 709
12.46.4 Member Enumeration Documentation . . . 709
12.46.4.1 HwMode . . . 709
12.46.5 Constructor & Destructor Documentation . . . 709
12.46.5.1 KPerfWidget . . . 709
12.46.5.2 ~KPerfWidget . . . 711
12.46.6 Member Function Documentation . . . 711
12.46.6.1 mode2Raw . . . 711
12.46.6.2 on_copyButton_clicked . . . 712
12.46.6.3 on_intensityBox_valueChanged . . . 713
12.46.6.4 raw2Mode . . . 713
12.46.6.5 setPerf . . . 714
12.46.6.6 uiUpdated . . . 715
12.46.7 Field Documentation . . . 716
12.46.7.1 HW_I_COUNT . . . 716
12.46.7.2 I_COUNT . . . 717

12.46.7.3 indicators . . . 717
12.46.7.4 k95Widgets . . . 717
12.46.7.5 perf . . . 717
12.46.7.6 profile . . . 717
12.46.7.7 ui . . . 717
12.46.7.8 updateMapper . . . 717
12.47 Ui::LayoutDialog Class Reference . . . 717
12.47.1 Detailed Description . . . 719
12.47.2 Member Function Documentation . . . 719
12.47.2.1 retranslateUi . . . 719
12.47.2.2 setupUi . . . 720
12.47.3 Field Documentation . . . 721
12.47.3.1 buttonBox . . . 721
12.47.3.2 comboBox . . . 721
12.47.3.3 gridLayout . . . 721
12.47.3.4 horizontalSpacer . . . 721
12.47.3.5 horizontalSpacer_2 . . . 721
12.47.3.6 label . . . 722
12.47.3.7 label_2 . . . 722
12.47.3.8 verticalSpacer . . . 722
12.47.3.9 verticalSpacer_2 . . . 722
12.47.3.10 verticalSpacer_3 . . . 722
12.47.3.11 verticalSpacer_4 . . . 722
12.48 LayoutDialog Class Reference . . . 722
12.48.1 Detailed Description . . . 724
12.48.2 Constructor & Destructor Documentation . . . 724
12.48.2.1 LayoutDialog . . . 724
12.48.2.2 ~LayoutDialog . . . 724
12.48.3 Member Function Documentation . . . 725
12.48.3.1 selected . . . 725
12.48.4 Field Documentation . . . 725
12.48.4.1 ui . . . 725
12.49 MacroReader Class Reference . . . 725
12.49.1 Detailed Description . . . 727
12.49.2 Constructor & Destructor Documentation . . . 728
12.49.2.1 MacroReader . . . 728
12.49.2.2 MacroReader . . . 728
12.49.2.3 ~MacroReader . . . 728
12.49.3 Member Function Documentation . . . 728
12.49.3.1 startWorkInAThread . . . 728

12.49.4 Field Documentation . . . 729
12.49.4.1 macroReaderThread . . . 729
12.50 MacroReaderThread Class Reference . . . 729
12.50.1 Detailed Description . . . 731
12.50.2 Constructor & Destructor Documentation . . . 731
12.50.2.1 MacroReaderThread . . . 731
12.50.3 Member Function Documentation . . . 732
12.50.3.1 readMacro . . . 732
12.50.3.2 run . . . 732
12.50.4 Field Documentation . . . 733
12.50.4.1 macroBox . . . 733
12.50.4.2 macroNumber . . . 733
12.50.4.3 macroPath . . . 733
12.50.4.4 macroText . . . 733
12.51 Ui::MainWindow Class Reference . . . 733
12.51.1 Detailed Description . . . 735
12.51.2 Member Function Documentation . . . 735
12.51.2.1 retranslateUi . . . 735
12.51.2.2 setupUi . . . 735
12.51.3 Field Documentation . . . 736
12.51.3.1 actionAbout . . . 736
12.51.3.2 actionExit . . . 736
12.51.3.3 centralWidget . . . 737
12.51.3.4 gridLayout . . . 737
12.51.3.5 tabWidget . . . 737
12.52 MainWindow Class Reference . . . 737
12.52.1 Detailed Description . . . 740
12.52.2 Constructor & Destructor Documentation . . . 740
12.52.2.1 MainWindow . . . 740
12.52.2.2 ~MainWindow . . . 742
12.52.3 Member Function Documentation . . . 742
12.52.3.1 addDevice . . . 742
12.52.3.2 checkFwUpdates . . . 743
12.52.3.3 cleanup . . . 744
12.52.3.4 closeEvent . . . 745
12.52.3.5 iconClicked . . . 745
12.52.3.6 quitApp . . . 746
12.52.3.7 removeDevice . . . 746
12.52.3.8 showFwUpdateNotification . . . 747
12.52.3.9 showWindow . . . 748

12.52.3.10 stateChange . . . 748
12.52.3.11 timerTick . . . 749
12.52.3.12 toggleTrayIcon . . . 750
12.52.3.13 updateVersion . . . 751
12.52.4 Field Documentation . . . 752
12.52.4.1 closeAction . . . 752
12.52.4.2 kbWidgets . . . 752
12.52.4.3 mainWindow . . . 752
12.52.4.4 restoreAction . . . 752
12.52.4.5 settingsWidget . . . 752
12.52.4.6 trayIcon . . . 752
12.52.4.7 trayIconMenu . . . 752
12.52.4.8 ui . . . 753
12.53 Ui::ModeSelectDialog Class Reference . . . 753
12.53.1 Detailed Description . . . 754
12.53.2 Member Function Documentation . . . 754
12.53.2.1 retranslateUi . . . 754
12.53.2.2 setupUi . . . 755
12.53.3 Field Documentation . . . 756
12.53.3.1 buttonBox . . . 756
12.53.3.2 gridLayout . . . 756
12.53.3.3 label . . . 756
12.53.3.4 modeList . . . 756
12.53.3.5 selAllButton . . . 756
12.53.3.6 selNoneButton . . . 757
12.54 ModeSelectDialog Class Reference . . . 757
12.54.1 Detailed Description . . . 758
12.54.2 Constructor & Destructor Documentation . . . 759
12.54.2.1 ModeSelectDialog . . . 759
12.54.2.2 ~ModeSelectDialog . . . 759
12.54.3 Member Function Documentation . . . 759
12.54.3.1 on_selAllButton_clicked . . . 759
12.54.3.2 on_selNoneButton_clicked . . . 760
12.54.3.3 selection . . . 760
12.54.4 Field Documentation . . . 760
12.54.4.1 _modeList . . . 760
12.54.4.2 ui . . . 760
12.55 Ui::MPerfWidget Class Reference . . . 761
12.55.1 Detailed Description . . . 763
12.55.2 Member Function Documentation . . . 764

12.55.2.1 retranslateUi . . . 764
12.55.2.2 setupUi . . . 765
12.55.3 Field Documentation . . . 773
12.55.3.1 aSnapBox . . . 773
12.55.3.2 copyButton . . . 773
12.55.3.3 eBox1 . . . 773
12.55.3.4 eBox2 . . . 773
12.55.3.5 eBox3 . . . 773
12.55.3.6 eBox4 . . . 773
12.55.3.7 eBox5 . . . 773
12.55.3.8 gridLayout . . . 774
12.55.3.9 horizontalLayout . . . 774
12.55.3.10 horizontalLayout_2 . . . 774
12.55.3.11 horizontalLayout_4 . . . 774
12.55.3.12 horizontalSpacer . . . 774
12.55.3.13 horizontalSpacer_2 . . . 774
12.55.3.14 horizontalSpacer_3 . . . 774
12.55.3.15 horizontalSpacer_4 . . . 774
12.55.3.16 horizontalSpacer_5 . . . 774
12.55.3.17 horizontalSpacer_6 . . . 775
12.55.3.18 horizontalSpacer_7 . . . 775
12.55.3.19 iButton0 . . . 775
12.55.3.20 iButton1 . . . 775
12.55.3.21 iButton2 . . . 775
12.55.3.22 iButton3 . . . 775
12.55.3.23 iButton4 . . . 775
12.55.3.24 iButton5 . . . 775
12.55.3.25 iButtonO . . . 775
12.55.3.26 indicBox . . . 776
12.55.3.27 label . . . 776
12.55.3.28 label_11 . . . 776
12.55.3.29 label_12 . . . 776
12.55.3.30 label_13 . . . 776
12.55.3.31 label_14 . . . 776
12.55.3.32 label_15 . . . 776
12.55.3.33 label_2 . . . 776
12.55.3.34 label_25 . . . 776
12.55.3.35 label_3 . . . 777
12.55.3.36 label_4 . . . 777
12.55.3.37 label_5 . . . 777

12.55.3.38label_6 . . . 777
12.55.3.39label_7 . . . 777
12.55.3.40label_8 . . . 777
12.55.3.41llHeightBox . . . 777
12.55.3.42line . . . 777
12.55.3.43line_2 . . . 777
12.55.3.44line_3 . . . 778
12.55.3.45spinBox . . . 778
12.55.3.46verticalLayout . . . 778
12.55.3.47verticalSpacer . . . 778
12.55.3.48verticalSpacer_2 . . . 778
12.55.3.49verticalSpacer_3 . . . 778
12.55.3.50xBox0 . . . 778
12.55.3.51xBox1 . . . 778
12.55.3.52xBox2 . . . 778
12.55.3.53xBox3 . . . 779
12.55.3.54xBox4 . . . 779
12.55.3.55xBox5 . . . 779
12.55.3.56xSlider0 . . . 779
12.55.3.57xSlider1 . . . 779
12.55.3.58xSlider2 . . . 779
12.55.3.59xSlider3 . . . 779
12.55.3.60xSlider4 . . . 779
12.55.3.61xSlider5 . . . 779
12.55.3.62xyBox . . . 780
12.55.3.63yBox0 . . . 780
12.55.3.64yBox1 . . . 780
12.55.3.65yBox2 . . . 780
12.55.3.66yBox3 . . . 780
12.55.3.67yBox4 . . . 780
12.55.3.68yBox5 . . . 780
12.55.3.69ySlider0 . . . 780
12.55.3.70ySlider1 . . . 780
12.55.3.71ySlider2 . . . 781
12.55.3.72ySlider3 . . . 781
12.55.3.73ySlider4 . . . 781
12.55.3.74ySlider5 . . . 781
12.56MPerfWidget Class Reference . . . 781
12.56.1 Detailed Description . . . 784
12.56.2 Data Structure Documentation . . . 784

12.56.2.1 struct MPerfWidget::DpiUi . . . 784
12.56.3 Constructor & Destructor Documentation . . . 785
12.56.3.1 MPerfWidget . . . 785
12.56.3.2 ~MPerfWidget . . . 787
12.56.4 Member Function Documentation . . . 787
12.56.4.1 boxXChanged . . . 787
12.56.4.2 boxYChanged . . . 788
12.56.4.3 colorChanged . . . 789
12.56.4.4 colorClicked . . . 790
12.56.4.5 enableChanged . . . 790
12.56.4.6 on_aSnapBox_clicked . . . 791
12.56.4.7 on_copyButton_clicked . . . 791
12.56.4.8 on_indicBox_clicked . . . 792
12.56.4.9 on_lHeightBox_activated . . . 792
12.56.4.10 on_spinBox_valueChanged . . . 793
12.56.4.11 on_xyBox_clicked . . . 793
12.56.4.12 setPerf . . . 794
12.56.4.13 sliderXMoved . . . 795
12.56.4.14 sliderYMoved . . . 796
12.56.5 Field Documentation . . . 797
12.56.5.1 _xyLink . . . 797
12.56.5.2 boxXMapper . . . 797
12.56.5.3 boxYMapper . . . 797
12.56.5.4 buttonMapper1 . . . 797
12.56.5.5 buttonMapper2 . . . 798
12.56.5.6 colorLink . . . 798
12.56.5.7 DPI_COUNT . . . 798
12.56.5.8 DPI_MAX . . . 798
12.56.5.9 DPI_MIN . . . 798
12.56.5.10 enableMapper . . . 798
12.56.5.11 isSetting . . . 798
12.56.5.12 perf . . . 798
12.56.5.13 profile . . . 798
12.56.5.14 sliderXMapper . . . 798
12.56.5.15 sliderYMapper . . . 799
12.56.5.16 stages . . . 799
12.56.5.17 ui . . . 799
12.57 GradientDialog::Preset Struct Reference . . . 799
12.57.1 Detailed Description . . . 800
12.57.2 Constructor & Destructor Documentation . . . 800

12.57.2.1 Preset . . . 800
12.57.2.2 Preset . . . 800
12.57.2.3 Preset . . . 800
12.57.3 Field Documentation . . . 800
12.57.3.1 builtIn . . . 800
12.57.3.2 gradient . . . 800
12.57.3.3 name . . . 801
12.58 QDialog Class Reference . . . 801
12.59 QIODevice Class Reference . . . 802
12.60 QIODevice_descriptor Struct Reference . . . 803
12.60.1 Detailed Description . . . 803
12.60.2 Constructor & Destructor Documentation . . . 803
12.60.2.1 QIODevice_descriptor . . . 803
12.60.3 Field Documentation . . . 803
12.60.3.1 pos . . . 803
12.61 QListWidget Class Reference . . . 804
12.62 QMainWindow Class Reference . . . 805
12.63 QObject Class Reference . . . 806
12.64 QPushButton Class Reference . . . 807
12.65 QThread Class Reference . . . 808
12.66 QuaAdler32 Class Reference . . . 809
12.66.1 Detailed Description . . . 810
12.66.2 Constructor & Destructor Documentation . . . 811
12.66.2.1 QuaAdler32 . . . 811
12.66.3 Member Function Documentation . . . 811
12.66.3.1 calculate . . . 811
12.66.3.2 reset . . . 811
12.66.3.3 update . . . 812
12.66.3.4 value . . . 812
12.66.4 Field Documentation . . . 812
12.66.4.1 checksum . . . 812
12.67 QuaChecksum32 Class Reference . . . 812
12.67.1 Detailed Description . . . 814
12.67.2 Member Function Documentation . . . 814
12.67.2.1 calculate . . . 814
12.67.2.2 reset . . . 814
12.67.2.3 update . . . 814
12.67.2.4 value . . . 814
12.68 QuaCrc32 Class Reference . . . 815
12.68.1 Detailed Description . . . 816

12.68.2 Constructor & Destructor Documentation . . . 817
12.68.2.1 QuaCrc32 . . . 817
12.68.3 Member Function Documentation . . . 817
12.68.3.1 calculate . . . 817
12.68.3.2 reset . . . 817
12.68.3.3 update . . . 818
12.68.3.4 value . . . 818
12.68.4 Field Documentation . . . 818
12.68.4.1 checksum . . . 818
12.69 QuaGzipFile Class Reference . . . 818
12.69.1 Detailed Description . . . 821
12.69.2 Constructor & Destructor Documentation . . . 821
12.69.2.1 QuaGzipFile . . . 821
12.69.2.2 QuaGzipFile . . . 821
12.69.2.3 QuaGzipFile . . . 822
12.69.2.4 ~QuaGzipFile . . . 822
12.69.2.5 QuaGzipFile . . . 822
12.69.3 Member Function Documentation . . . 822
12.69.3.1 close . . . 822
12.69.3.2 flush . . . 823
12.69.3.3 getFileName . . . 823
12.69.3.4 isSequential . . . 823
12.69.3.5 open . . . 823
12.69.3.6 open . . . 824
12.69.3.7 operator= . . . 824
12.69.3.8 readData . . . 824
12.69.3.9 setFileName . . . 824
12.69.3.10 writeData . . . 825
12.69.4 Field Documentation . . . 825
12.69.4.1 d . . . 825
12.70 QuaZIODevice Class Reference . . . 825
12.70.1 Detailed Description . . . 828
12.70.2 Constructor & Destructor Documentation . . . 828
12.70.2.1 QuaZIODevice . . . 828
12.70.2.2 ~QuaZIODevice . . . 828
12.70.3 Member Function Documentation . . . 828
12.70.3.1 close . . . 828
12.70.3.2 flush . . . 829
12.70.3.3 getIoDevice . . . 830
12.70.3.4 isSequential . . . 830

12.70.3.5 open . . . 830
12.70.3.6 readData . . . 831
12.70.3.7 writeData . . . 832
12.70.4 Field Documentation . . . 832
12.70.4.1 d . . . 832
12.71 QuaZip Class Reference . . . 833
12.71.1 Detailed Description . . . 835
12.71.2 Member Enumeration Documentation . . . 836
12.71.2.1 CaseSensitivity . . . 836
12.71.2.2 Constants . . . 836
12.71.2.3 Mode . . . 836
12.71.3 Constructor & Destructor Documentation . . . 837
12.71.3.1 QuaZip . . . 837
12.71.3.2 QuaZip . . . 837
12.71.3.3 QuaZip . . . 837
12.71.3.4 QuaZip . . . 837
12.71.3.5 ~QuaZip . . . 837
12.71.4 Member Function Documentation . . . 838
12.71.4.1 close . . . 838
12.71.4.2 convertCaseSensitivity . . . 839
12.71.4.3 getComment . . . 840
12.71.4.4 getCommentCodec . . . 840
12.71.4.5 getCurrentFileInfo . . . 841
12.71.4.6 getCurrentFileInfo . . . 842
12.71.4.7 getCurrentFileName . . . 843
12.71.4.8 getEntriesCount . . . 844
12.71.4.9 getFileInfoList . . . 845
12.71.4.10 getFileInfoList64 . . . 846
12.71.4.11 getFileNameCodec . . . 846
12.71.4.12 getFileNameList . . . 847
12.71.4.13 getIoDevice . . . 847
12.71.4.14 getMode . . . 847
12.71.4.15 getUnzFile . . . 848
12.71.4.16 getZipError . . . 849
12.71.4.17 getZipFile . . . 850
12.71.4.18 getZipName . . . 850
12.71.4.19 goToFirstFile . . . 851
12.71.4.20 goToNextFile . . . 852
12.71.4.21 hasCurrentFile . . . 853
12.71.4.22 isAutoClose . . . 853

12.71.4.23 isDataDescriptorWritingEnabled . . . 853
12.71.4.24 isOpen . . . 854
12.71.4.25 isZip64Enabled . . . 854
12.71.4.26 open . . . 855
12.71.4.27 operator= . . . 858
12.71.4.28 setAutoClose . . . 858
12.71.4.29 setComment . . . 859
12.71.4.30 setCommentCodec . . . 859
12.71.4.31 setCommentCodec . . . 859
12.71.4.32 setCurrentFile . . . 860
12.71.4.33 setDataDescriptorWritingEnabled . . . 861
12.71.4.34 setDefaultFileNameCodec . . . 862
12.71.4.35 setDefaultFileNameCodec . . . 863
12.71.4.36 setFileNameCodec . . . 863
12.71.4.37 setFileNameCodec . . . 863
12.71.4.38 setIoDevice . . . 863
12.71.4.39 setZip64Enabled . . . 864
12.71.4.40 setZipName . . . 864
12.71.5 Friends And Related Function Documentation . . . 865
12.71.5.1 QuaZipPrivate . . . 865
12.71.6 Field Documentation . . . 865
12.71.6.1 p . . . 865
12.72 QuaZipDir Class Reference . . . 865
12.72.1 Detailed Description . . . 868
12.72.2 Constructor & Destructor Documentation . . . 868
12.72.2.1 QuaZipDir . . . 868
12.72.2.2 QuaZipDir . . . 868
12.72.2.3 ~QuaZipDir . . . 869
12.72.3 Member Function Documentation . . . 869
12.72.3.1 caseSensitivity . . . 869
12.72.3.2 cd . . . 869
12.72.3.3 cdUp . . . 871
12.72.3.4 count . . . 871
12.72.3.5 dirName . . . 872
12.72.3.6 entryInfoList . . . 872
12.72.3.7 entryInfoList . . . 873
12.72.3.8 entryInfoList64 . . . 873
12.72.3.9 entryInfoList64 . . . 874
12.72.3.10 entryList . . . 874
12.72.3.11 entryList . . . 875

12.72.3.12 exists . . . 875
12.72.3.13 exists . . . 876
12.72.3.14 filePath . . . 877
12.72.3.15 filter . . . 878
12.72.3.16 isRoot . . . 878
12.72.3.17 nameFilters . . . 878
12.72.3.18 operator!= . . . 879
12.72.3.19 operator= . . . 879
12.72.3.20 operator== . . . 879
12.72.3.21 operator[] . . . 879
12.72.3.22 path . . . 880
12.72.3.23 relativeFilePath . . . 880
12.72.3.24 setCaseSensitivity . . . 880
12.72.3.25 setFilter . . . 881
12.72.3.26 setNameFilters . . . 881
12.72.3.27 setPath . . . 881
12.72.3.28 setSorting . . . 882
12.72.3.29 sorting . . . 882
12.72.4 Field Documentation . . . 882
12.72.4.1 d . . . 882
12.73 QuaZipFile Class Reference . . . 883
12.73.1 Detailed Description . . . 885
12.73.2 Sequential or random-access? . . . 886
12.73.3 Constructor & Destructor Documentation . . . 886
12.73.3.1 QuaZipFile . . . 886
12.73.3.2 QuaZipFile . . . 886
12.73.3.3 QuaZipFile . . . 886
12.73.3.4 QuaZipFile . . . 886
12.73.3.5 QuaZipFile . . . 887
12.73.3.6 QuaZipFile . . . 887
12.73.3.7 ~QuaZipFile . . . 888
12.73.4 Member Function Documentation . . . 888
12.73.4.1 atEnd . . . 888
12.73.4.2 bytesAvailable . . . 889
12.73.4.3 close . . . 889
12.73.4.4 csize . . . 890
12.73.4.5 getActualFileName . . . 891
12.73.4.6 getCaseSensitivity . . . 892
12.73.4.7 getFileInfo . . . 892
12.73.4.8 getFileInfo . . . 893

12.73.4.9 getFileName . . . 894
12.73.4.10 getZip . . . 894
12.73.4.11 getZipError . . . 894
12.73.4.12 getZipName . . . 895
12.73.4.13 isRaw . . . 895
12.73.4.14 isSequential . . . 896
12.73.4.15 open . . . 896
12.73.4.16 open . . . 897
12.73.4.17 open . . . 897
12.73.4.18 open . . . 898
12.73.4.19 operator= . . . 900
12.73.4.20 pos . . . 900
12.73.4.21 readData . . . 901
12.73.4.22 setFileName . . . 902
12.73.4.23 setZip . . . 902
12.73.4.24 setZipName . . . 903
12.73.4.25 size . . . 903
12.73.4.26 usize . . . 904
12.73.4.27 writeData . . . 905
12.73.5 Friends And Related Function Documentation . . . 905
12.73.5.1 QuaZipFilePrivate . . . 906
12.73.6 Field Documentation . . . 906
12.73.6.1 p . . . 906
12.74 QuaZipFileInfo Struct Reference . . . 906
12.74.1 Detailed Description . . . 907
12.74.2 Member Function Documentation . . . 907
12.74.2.1 getPermissions . . . 907
12.74.3 Field Documentation . . . 908
12.74.3.1 comment . . . 908
12.74.3.2 compressedSize . . . 908
12.74.3.3 crc . . . 908
12.74.3.4 dateTime . . . 908
12.74.3.5 diskNumberStart . . . 908
12.74.3.6 externalAttr . . . 908
12.74.3.7 extra . . . 908
12.74.3.8 flags . . . 909
12.74.3.9 internalAttr . . . 909
12.74.3.10 method . . . 909
12.74.3.11 name . . . 909
12.74.3.12 uncompressedSize . . . 909

12.74.3.13 versionCreated . . . 909
12.74.3.14 versionNeeded . . . 909
12.75 QuaZipFileInfo64 Struct Reference . . . 909
12.75.1 Detailed Description . . . 911
12.75.2 Member Function Documentation . . . 911
12.75.2.1 getNTFSaTime . . . 911
12.75.2.2 getNTFScTime . . . 912
12.75.2.3 getNTFSmTime . . . 912
12.75.2.4 getPermissions . . . 913
12.75.2.5 isEncrypted . . . 914
12.75.2.6 toQuaZipFileInfo . . . 914
12.75.3 Field Documentation . . . 915
12.75.3.1 comment . . . 915
12.75.3.2 compressedSize . . . 915
12.75.3.3 crc . . . 915
12.75.3.4 dateTime . . . 915
12.75.3.5 diskNumberStart . . . 915
12.75.3.6 externalAttr . . . 916
12.75.3.7 extra . . . 916
12.75.3.8 flags . . . 916
12.75.3.9 internalAttr . . . 916
12.75.3.10 method . . . 916
12.75.3.11 name . . . 916
12.75.3.12 uncompressedSize . . . 916
12.75.3.13 versionCreated . . . 916
12.75.3.14 versionNeeded . . . 916
12.76 QuaZipFilePrivate Class Reference . . . 917
12.76.1 Detailed Description . . . 918
12.76.2 Constructor & Destructor Documentation . . . 918
12.76.2.1 QuaZipFilePrivate . . . 918
12.76.2.2 QuaZipFilePrivate . . . 918
12.76.2.3 QuaZipFilePrivate . . . 918
12.76.2.4 QuaZipFilePrivate . . . 919
12.76.2.5 ~QuaZipFilePrivate . . . 919
12.76.3 Member Function Documentation . . . 919
12.76.3.1 resetZipError . . . 919
12.76.3.2 setZipError . . . 919
12.76.4 Friends And Related Function Documentation . . . 920
12.76.4.1 QuaZipFile . . . 920
12.76.5 Field Documentation . . . 920

12.76.5.1 caseSensitivity . . . 920
12.76.5.2 crc . . . 920
12.76.5.3 fileName . . . 920
12.76.5.4 internal . . . 921
12.76.5.5 q . . . 921
12.76.5.6 raw . . . 921
12.76.5.7 uncompressedSize . . . 921
12.76.5.8 writePos . . . 921
12.76.5.9 zip . . . 921
12.76.5.10 zipError . . . 921
12.77 QuaZipNewInfo Struct Reference . . . 922
12.77.1 Detailed Description . . . 923
12.77.2 Constructor & Destructor Documentation . . . 923
12.77.2.1 QuaZipNewInfo . . . 923
12.77.2.2 QuaZipNewInfo . . . 923
12.77.2.3 QuaZipNewInfo . . . 924
12.77.2.4 QuaZipNewInfo . . . 925
12.77.3 Member Function Documentation . . . 925
12.77.3.1 setFileDateTime . . . 925
12.77.3.2 setFileNTFSaTime . . . 925
12.77.3.3 setFileNTFScTime . . . 926
12.77.3.4 setFileNTFSmTime . . . 927
12.77.3.5 setFileNTFSTimes . . . 928
12.77.3.6 setFilePermissions . . . 928
12.77.3.7 setPermissions . . . 929
12.77.4 Field Documentation . . . 929
12.77.4.1 comment . . . 929
12.77.4.2 dateTime . . . 929
12.77.4.3 externalAttr . . . 930
12.77.4.4 extraGlobal . . . 930
12.77.4.5 extraLocal . . . 930
12.77.4.6 internalAttr . . . 930
12.77.4.7 name . . . 930
12.77.4.8 uncompressedSize . . . 930
12.78 QuaZipPrivate Class Reference . . . 930
12.78.1 Detailed Description . . . 932
12.78.2 Constructor & Destructor Documentation . . . 932
12.78.2.1 QuaZipPrivate . . . 932
12.78.2.2 QuaZipPrivate . . . 933
12.78.2.3 QuaZipPrivate . . . 933

12.78.3 Member Function Documentation . . . 933
12.78.3.1 addCurrentFileToDirectoryMap . . . 933
12.78.3.2 clearDirectoryMap . . . 934
12.78.3.3 getDefaultFileNameCodec . . . 935
12.78.3.4 getFileInfoList . . . 935
12.78.3.5 goToFirstUnmappedFile . . . 936
12.78.4 Friends And Related Function Documentation . . . 937
12.78.4.1 QuaZip . . . 937
12.78.5 Field Documentation . . . 937
12.78.5.1 "@4 . . . 937
12.78.5.2 autoClose . . . 937
12.78.5.3 comment . . . 937
12.78.5.4 commentCodec . . . 937
12.78.5.5 dataDescriptorWritingEnabled . . . 938
12.78.5.6 defaultFileNameCodec . . . 938
12.78.5.7 directoryCaseInsensitive . . . 938
12.78.5.8 directoryCaseSensitive . . . 938
12.78.5.9 fileNameCodec . . . 938
12.78.5.10 hasCurrentFile_f . . . 938
12.78.5.11 ioDevice . . . 938
12.78.5.12 lastMappedDirectoryEntry . . . 938
12.78.5.13 mode . . . 938
12.78.5.14 q . . . 939
12.78.5.15 zip64 . . . 939
12.78.5.16 zipError . . . 939
12.78.5.17 zipName . . . 939
12.79 QWidget Class Reference . . . 939
12.80 Ui::RebindWidget Class Reference . . . 940
12.80.1 Detailed Description . . . 943
12.80.2 Member Function Documentation . . . 943
12.80.2.1 retranslateUi . . . 943
12.80.2.2 setupUi . . . 946
12.80.3 Field Documentation . . . 956
12.80.3.1 actionTab . . . 956
12.80.3.2 animBox . . . 956
12.80.3.3 animButton . . . 956
12.80.3.4 animKrBox . . . 957
12.80.3.5 animOnceBox . . . 957
12.80.3.6 animTab . . . 957
12.80.3.7 applyButton . . . 957

12.80.3.8 btnClearMacro . . . 957
12.80.3.9 btnStartMacro . . . 957
12.80.3.10 btnStopMacro . . . 957
12.80.3.11 cancelButton . . . 957
12.80.3.12 delayButtons . . . 958
12.80.3.13 dpiBox . . . 958
12.80.3.14 dpiButton . . . 958
12.80.3.15 dpiCustLabel . . . 958
12.80.3.16 dpiCustXBox . . . 958
12.80.3.17 dpiCustYBox . . . 958
12.80.3.18 dpiWarning . . . 958
12.80.3.19 fnBox . . . 958
12.80.3.20 fnButton . . . 959
12.80.3.21 gridLayout . . . 959
12.80.3.22 gridLayout_10 . . . 959
12.80.3.23 gridLayout_2 . . . 959
12.80.3.24 gridLayout_3 . . . 959
12.80.3.25 gridLayout_4 . . . 959
12.80.3.26 gridLayout_5 . . . 959
12.80.3.27 gridLayout_8 . . . 959
12.80.3.28 groupBox_2 . . . 959
12.80.3.29 groupBox_3 . . . 960
12.80.3.30 groupBox_4 . . . 960
12.80.3.31 horizontalLayout . . . 960
12.80.3.32 horizontalLayout_2 . . . 960
12.80.3.33 horizontalLayout_3 . . . 960
12.80.3.34 horizontalSpacer . . . 960
12.80.3.35 horizontalSpacer_10 . . . 960
12.80.3.36 horizontalSpacer_11 . . . 960
12.80.3.37 horizontalSpacer_12 . . . 960
12.80.3.38 horizontalSpacer_13 . . . 961
12.80.3.39 horizontalSpacer_14 . . . 961
12.80.3.40 horizontalSpacer_15 . . . 961
12.80.3.41 horizontalSpacer_16 . . . 961
12.80.3.42 horizontalSpacer_17 . . . 961
12.80.3.43 horizontalSpacer_18 . . . 961
12.80.3.44 horizontalSpacer_19 . . . 961
12.80.3.45 horizontalSpacer_2 . . . 961
12.80.3.46 horizontalSpacer_20 . . . 961
12.80.3.47 horizontalSpacer_21 . . . 962

12.80.3.48 horizontalSpacer_22 . . . 962
12.80.3.49 horizontalSpacer_23 . . . 962
12.80.3.50 horizontalSpacer_24 . . . 962
12.80.3.51 horizontalSpacer_25 . . . 962
12.80.3.52 horizontalSpacer_26 . . . 962
12.80.3.53 horizontalSpacer_3 . . . 962
12.80.3.54 horizontalSpacer_4 . . . 962
12.80.3.55 horizontalSpacer_5 . . . 962
12.80.3.56 horizontalSpacer_6 . . . 963
12.80.3.57 horizontalSpacer_7 . . . 963
12.80.3.58 horizontalSpacer_8 . . . 963
12.80.3.59 horizontalSpacer_9 . . . 963
12.80.3.60 keyTab . . . 963
12.80.3.61 label . . . 963
12.80.3.62 layoutWidget . . . 963
12.80.3.63 layoutWidget1 . . . 963
12.80.3.64 layoutWidget2 . . . 963
12.80.3.65 layoutWidget_2 . . . 964
12.80.3.66 lbl_macro . . . 964
12.80.3.67 lightBox . . . 964
12.80.3.68 lightButton . . . 964
12.80.3.69 lightWrapBox . . . 964
12.80.3.70 lockBox . . . 964
12.80.3.71 lockButton . . . 964
12.80.3.72 macroTab . . . 964
12.80.3.73 mb2Box . . . 965
12.80.3.74 mb2Button . . . 965
12.80.3.75 mbBox . . . 965
12.80.3.76 mbButton . . . 965
12.80.3.77 mediaBox . . . 965
12.80.3.78 mediaButton . . . 965
12.80.3.79 modBox . . . 965
12.80.3.80 modButton . . . 965
12.80.3.81 modeBox . . . 966
12.80.3.82 modeButton . . . 966
12.80.3.83 modeWrapBox . . . 966
12.80.3.84 mouseTab . . . 966
12.80.3.85 numBox . . . 966
12.80.3.86 numButton . . . 966
12.80.3.87 programKpBox . . . 966

12.80.3.88 programKpButton . . . 967
12.80.3.89 programKpExtra . . . 967
12.80.3.90 programKpModeBox . . . 967
12.80.3.91 programKpSIBox . . . 967
12.80.3.92 programKrBox . . . 967
12.80.3.93 programKrButton . . . 967
12.80.3.94 programKrExtra . . . 967
12.80.3.95 programKrModeBox . . . 967
12.80.3.96 programKrSIBox . . . 968
12.80.3.97 programTab . . . 968
12.80.3.98 progTipLabel . . . 968
12.80.3.99 pteMacroBox . . . 968
12.80.3.100 pteMacroComment . . . 968
12.80.3.101 pteMacroText . . . 968
12.80.3.102 rb_delay_asTyped . . . 968
12.80.3.103 rb_delay_default . . . 968
12.80.3.104 rb_delay_no . . . 969
12.80.3.105 resetButton . . . 969
12.80.3.106 tabWidget . . . 969
12.80.3.107 txtBuffer . . . 969
12.80.3.108 typingBox . . . 969
12.80.3.109 typingButton . . . 969
12.80.3.110 unbindButton . . . 969
12.80.3.111 verticalLayout . . . 969
12.80.3.112 verticalLayout_2 . . . 970
12.80.3.113 verticalSpacer . . . 970
12.80.3.114 verticalSpacer_10 . . . 970
12.80.3.115 verticalSpacer_11 . . . 970
12.80.3.116 verticalSpacer_2 . . . 970
12.80.3.117 verticalSpacer_3 . . . 970
12.80.3.118 verticalSpacer_4 . . . 970
12.80.3.119 verticalSpacer_5 . . . 970
12.80.3.120 verticalSpacer_6 . . . 970
12.80.3.121 verticalSpacer_7 . . . 971
12.80.3.122 verticalSpacer_8 . . . 971
12.80.3.123 wheelBox . . . 971
12.80.3.124 wheelButton . . . 971
12.81 RebindWidget Class Reference . . . 971
12.81.1 Detailed Description . . . 975
12.81.2 Constructor & Destructor Documentation . . . 975

12.81.2.1 RebindWidget . . . 975
12.81.2.2 ~RebindWidget . . . 976
12.81.3 Member Function Documentation . . . 976
12.81.3.1 applyChanges . . . 976
12.81.3.2 convertMacroBox . . . 978
12.81.3.3 helpStatus . . . 979
12.81.3.4 on_animBox_currentIndexChanged . . . 980
12.81.3.5 on_animButton_clicked . . . 981
12.81.3.6 on_applyButton_clicked . . . 981
12.81.3.7 on_btnClearMacro_clicked . . . 981
12.81.3.8 on_btnStartMacro_clicked . . . 982
12.81.3.9 on_btnStopMacro_clicked . . . 982
12.81.3.10 on_cancelButton_clicked . . . 983
12.81.3.11 on_dpiBox_currentIndexChanged . . . 984
12.81.3.12 on_dpiButton_clicked . . . 985
12.81.3.13 on_fnBox_currentIndexChanged . . . 985
12.81.3.14 on_fnButton_clicked . . . 985
12.81.3.15 on_lightBox_currentIndexChanged . . . 986
12.81.3.16 on_lightButton_clicked . . . 986
12.81.3.17 on_lockBox_currentIndexChanged . . . 986
12.81.3.18 on_lockButton_clicked . . . 987
12.81.3.19 on_mb2Box_currentIndexChanged . . . 987
12.81.3.20 on_mb2Button_clicked . . . 987
12.81.3.21 on_mbBox_currentIndexChanged . . . 987
12.81.3.22 on_mbButton_clicked . . . 988
12.81.3.23 on_mediaBox_currentIndexChanged . . . 988
12.81.3.24 on_mediaButton_clicked . . . 988
12.81.3.25 on_modBox_currentIndexChanged . . . 989
12.81.3.26 on_modButton_clicked . . . 989
12.81.3.27 on_modeBox_currentIndexChanged . . . 989
12.81.3.28 on_modeButton_clicked . . . 990
12.81.3.29 on_numBox_currentIndexChanged . . . 990
12.81.3.30 on_numButton_clicked . . . 990
12.81.3.31 on_programKpBox_textChanged . . . 990
12.81.3.32 on_programKpButton_clicked . . . 991
12.81.3.33 on_programKpSlBox_clicked . . . 991
12.81.3.34 on_programKrBox_textChanged . . . 991
12.81.3.35 on_programKrButton_clicked . . . 992
12.81.3.36 on_programKrSlBox_clicked . . . 992
12.81.3.37 on_pteMacroBox_textChanged . . . 992

12.81.3.38 on_rb_delay_asTyped_toggled . . . 993
12.81.3.39 on_rb_delay_default_toggled . . . 993
12.81.3.40 on_rb_delay_no_toggled . . . 994
12.81.3.41 on_resetButton_clicked . . . 994
12.81.3.42 on_typingBox_currentIndexChanged . . . 995
12.81.3.43 on_typingButton_clicked . . . 995
12.81.3.44 on_unbindButton_clicked . . . 995
12.81.3.45 on_wheelBox_currentIndexChanged . . . 996
12.81.3.46 on_wheelButton_clicked . . . 996
12.81.3.47 setBind . . . 996
12.81.3.48 setBox . . . 999
12.81.3.49 setCorrectRadioButton . . . 1000
12.81.3.50 setSelection . . . 1001
12.81.4 Field Documentation . . . 1004
12.81.4.1 bind . . . 1005
12.81.4.2 fnKeys . . . 1005
12.81.4.3 macReader . . . 1005
12.81.4.4 mediaKeys . . . 1005
12.81.4.5 modKeys . . . 1005
12.81.4.6 mouseExtKeys . . . 1005
12.81.4.7 mouseKeys . . . 1005
12.81.4.8 numKeys . . . 1005
12.81.4.9 profile . . . 1005
12.81.4.10 selection . . . 1006
12.81.4.11 TAB_ANIM . . . 1006
12.81.4.12 TAB_KB . . . 1006
12.81.4.13 TAB_MACRO . . . 1006
12.81.4.14 TAB_MOUSE . . . 1006
12.81.4.15 TAB_PROGRAM . . . 1006
12.81.4.16 TAB_SPECIAL . . . 1006
12.81.4.17 typingKeys . . . 1006
12.81.4.18 ui . . . 1006
12.81.4.19 wheelKeys . . . 1007
12.82 RListWidget Class Reference . . . 1007
12.82.1 Detailed Description . . . 1008
12.82.2 Constructor & Destructor Documentation . . . 1009
12.82.2.1 RListWidget . . . 1009
12.82.3 Member Function Documentation . . . 1009
12.82.3.1 change . . . 1009
12.82.3.2 enter . . . 1010

12.82.3.3 orderChanged . . . 1010
12.82.3.4 rescanItems . . . 1011
12.82.3.5 timerTick . . . 1011
12.82.4 Field Documentation . . . 1012
12.82.4.1 currentData . . . 1012
12.82.4.2 DATA_ROLE . . . 1012
12.82.4.3 dragged . . . 1012
12.82.4.4 previousItems . . . 1012
12.82.4.5 reorderTimer . . . 1012
12.83 SettingsWidget Class Reference . . . 1012
12.83.1 Detailed Description . . . 1015
12.83.2 Constructor & Destructor Documentation . . . 1015
12.83.2.1 SettingsWidget . . . 1015
12.83.2.2 ~SettingsWidget . . . 1016
12.83.3 Member Function Documentation . . . 1016
12.83.3.1 on_altBox_activated . . . 1017
12.83.3.2 on_autoFWBox_clicked . . . 1017
12.83.3.3 on_capsBox_activated . . . 1017
12.83.3.4 on_ctrlBox_activated . . . 1018
12.83.3.5 on_extraButton_clicked . . . 1018
12.83.3.6 on_layoutBox_activated . . . 1018
12.83.3.7 on_loginItemBox_clicked . . . 1019
12.83.3.8 on_pushButton_clicked . . . 1019
12.83.3.9 on_shiftBox_activated . . . 1019
12.83.3.10 on_winBox_activated . . . 1020
12.83.3.11 pollUpdates . . . 1020
12.83.3.12 setStatus . . . 1020
12.83.3.13 setVersion . . . 1021
12.83.3.14 showLayoutDialog . . . 1021
12.83.3.15 updateModifiers . . . 1022
12.83.4 Friends And Related Function Documentation . . . 1023
12.83.4.1 MainWindow . . . 1023
12.83.5 Field Documentation . . . 1023
12.83.5.1 extra . . . 1023
12.83.5.2 ui . . . 1023
12.84 Ui::SettingsWidget Class Reference . . . 1024
12.84.1 Detailed Description . . . 1026
12.84.2 Member Function Documentation . . . 1026
12.84.2.1 retranslateUi . . . 1026
12.84.2.2 setupUi . . . 1027

12.84.3 Field Documentation . . . 1031
12.84.3.1 altBox . . . 1031
12.84.3.2 altLabel . . . 1031
12.84.3.3 autoFWBox . . . 1031
12.84.3.4 capsBox . . . 1031
12.84.3.5 capsLabel . . . 1032
12.84.3.6 ctrlBox . . . 1032
12.84.3.7 ctrlLabel . . . 1032
12.84.3.8 devicesLabel . . . 1032
12.84.3.9 extraButton . . . 1032
12.84.3.10 gridLayout . . . 1032
12.84.3.11 gridLayout_3 . . . 1032
12.84.3.12 gridLayout_5 . . . 1032
12.84.3.13 horizontalLayout . . . 1032
12.84.3.14 horizontalLayout_2 . . . 1033
12.84.3.15 horizontalSpacer . . . 1033
12.84.3.16 horizontalSpacer_10 . . . 1033
12.84.3.17 horizontalSpacer_2 . . . 1033
12.84.3.18 horizontalSpacer_7 . . . 1033
12.84.3.19 horizontalSpacer_9 . . . 1033
12.84.3.20 label . . . 1033
12.84.3.21 label_2 . . . 1033
12.84.3.22 label_3 . . . 1033
12.84.3.23 label_5 . . . 1034
12.84.3.24 label_6 . . . 1034
12.84.3.25 label_8 . . . 1034
12.84.3.26 layoutBox . . . 1034
12.84.3.27 line . . . 1034
12.84.3.28 line_2 . . . 1034
12.84.3.29 line_3 . . . 1034
12.84.3.30 loginItemBox . . . 1034
12.84.3.31 pushButton . . . 1034
12.84.3.32 shiftBox . . . 1035
12.84.3.33 shiftLabel . . . 1035
12.84.3.34 versionLabel . . . 1035
12.84.3.35 verticalSpacer . . . 1035
12.84.3.36 verticalSpacer_10 . . . 1035
12.84.3.37 verticalSpacer_2 . . . 1035
12.84.3.38 verticalSpacer_3 . . . 1035
12.84.3.39 verticalSpacer_4 . . . 1035

12.84.3.40 verticalSpacer_5 . . . 1035
12.84.3.41 verticalSpacer_6 . . . 1036
12.84.3.42 verticalSpacer_7 . . . 1036
12.84.3.43 verticalSpacer_8 . . . 1036
12.84.3.44 winBox . . . 1036
12.84.3.45 winLabel . . . 1036
12.85 SGroup Class Reference . . . 1036
12.85.1 Detailed Description . . . 1038
12.85.2 Constructor & Destructor Documentation . . . 1038
12.85.2.1 SGroup . . . 1038
12.85.2.2 ~SGroup . . . 1038
12.85.3 Field Documentation . . . 1038
12.85.3.1 _settings . . . 1038
12.86 Ui_AnimAddDialog Class Reference . . . 1039
12.86.1 Detailed Description . . . 1040
12.86.2 Member Function Documentation . . . 1040
12.86.2.1 retranslateUi . . . 1040
12.86.2.2 setupUi . . . 1040
12.86.3 Field Documentation . . . 1042
12.86.3.1 animBox . . . 1042
12.86.3.2 buttonBox . . . 1042
12.86.3.3 gridLayout . . . 1042
12.86.3.4 horizontalLayout . . . 1042
12.86.3.5 horizontalSpacer . . . 1042
12.86.3.6 horizontalSpacer_2 . . . 1043
12.86.3.7 label . . . 1043
12.86.3.8 label_2 . . . 1043
12.86.3.9 presetBox . . . 1043
12.86.3.10 previewBox . . . 1043
12.86.3.11 verticalSpacer_2 . . . 1043
12.86.3.12 verticalSpacer_3 . . . 1043
12.87 Ui_AnimSettingDialog Class Reference . . . 1043
12.87.1 Detailed Description . . . 1046
12.87.2 Member Function Documentation . . . 1046
12.87.2.1 retranslateUi . . . 1046
12.87.2.2 setupUi . . . 1047
12.87.3 Field Documentation . . . 1050
12.87.3.1 animName . . . 1050
12.87.3.2 buttonBox . . . 1051
12.87.3.3 delayBox . . . 1051

12.87.3.4 gridLayout . . . 1051
12.87.3.5 gridLayout_2 . . . 1051
12.87.3.6 horizontalSpacer . . . 1051
12.87.3.7 horizontalSpacer_2 . . . 1051
12.87.3.8 horizontalSpacer_3 . . . 1051
12.87.3.9 horizontalSpacer_4 . . . 1051
12.87.3.10 horizontalSpacer_5 . . . 1051
12.87.3.11 horizontalSpacer_6 . . . 1052
12.87.3.12 infoTab . . . 1052
12.87.3.13 kpDelayBox . . . 1052
12.87.3.14 kpModeStopBox . . . 1052
12.87.3.15 kpReleaseBox . . . 1052
12.87.3.16 kpRepeatBox . . . 1052
12.87.3.17 kpRepeatLabel . . . 1052
12.87.3.18 kpRepeatLabel_2 . . . 1052
12.87.3.19 label . . . 1052
12.87.3.20 label_10 . . . 1053
12.87.3.21 label_11 . . . 1053
12.87.3.22 label_12 . . . 1053
12.87.3.23 label_13 . . . 1053
12.87.3.24 label_2 . . . 1053
12.87.3.25 label_3 . . . 1053
12.87.3.26 label_4 . . . 1053
12.87.3.27 label_5 . . . 1053
12.87.3.28 label_6 . . . 1053
12.87.3.29 label_7 . . . 1054
12.87.3.30 label_8 . . . 1054
12.87.3.31 label_9 . . . 1054
12.87.3.32 line . . . 1054
12.87.3.33 line_2 . . . 1054
12.87.3.34 repeatBox . . . 1054
12.87.3.35 repeatLabel . . . 1054
12.87.3.36 repeatLabel_2 . . . 1054
12.87.3.37 sAuthorLabel . . . 1054
12.87.3.38 sDescLabel . . . 1055
12.87.3.39 settingsGrid . . . 1055
12.87.3.40 settingTab . . . 1055
12.87.3.41 sLicenseLabel . . . 1055
12.87.3.42 sNameLabel . . . 1055
12.87.3.43 sVerLabel . . . 1055

12.87.3.44 sYearLabel . . . 1055
12.87.3.45 tabWidget . . . 1055
12.87.3.46 timeGrid . . . 1055
12.87.3.47 timeTab . . . 1056
12.87.3.48 verticalSpacer . . . 1056
12.87.3.49 verticalSpacer_2 . . . 1056
12.87.3.50 verticalSpacer_3 . . . 1056
12.88 Ui_ExtraSettingsWidget Class Reference . . . 1056
12.88.1 Detailed Description . . . 1058
12.88.2 Member Function Documentation . . . 1059
12.88.2.1 retranslateUi . . . 1059
12.88.2.2 setupUi . . . 1060
12.88.3 Field Documentation . . . 1063
12.88.3.1 animCountLabel . . . 1063
12.88.3.2 animPathLabel . . . 1064
12.88.3.3 animScanButton . . . 1064
12.88.3.4 brightnessBox . . . 1064
12.88.3.5 buttonBox . . . 1064
12.88.3.6 delayBox . . . 1064
12.88.3.7 ditherBox . . . 1064
12.88.3.8 fpsBox . . . 1064
12.88.3.9 fpsWarnLabel . . . 1064
12.88.3.10 gridLayout . . . 1064
12.88.3.11 horizontalLayout . . . 1065
12.88.3.12 horizontalLayout_2 . . . 1065
12.88.3.13 horizontalSpacer . . . 1065
12.88.3.14 horizontalSpacer_2 . . . 1065
12.88.3.15 horizontalSpacer_3 . . . 1065
12.88.3.16 label . . . 1065
12.88.3.17 label_10 . . . 1065
12.88.3.18 label_11 . . . 1065
12.88.3.19 label_2 . . . 1065
12.88.3.20 label_3 . . . 1066
12.88.3.21 label_5 . . . 1066
12.88.3.22 label_6 . . . 1066
12.88.3.23 line . . . 1066
12.88.3.24 line_2 . . . 1066
12.88.3.25 line_3 . . . 1066
12.88.3.26 mAccelBox . . . 1066
12.88.3.27 osxLabel . . . 1066

12.88.3.28 osxLine . . . 1066
12.88.3.29 sAccelBox . . . 1067
12.88.3.30 sSpeedBox . . . 1067
12.88.3.31 sSpeedWidget . . . 1067
12.88.3.32 trayBox . . . 1067
12.88.3.33 verticalSpacer . . . 1067
12.88.3.34 verticalSpacer_10 . . . 1067
12.88.3.35 verticalSpacer_11 . . . 1067
12.88.3.36 verticalSpacer_2 . . . 1067
12.88.3.37 verticalSpacer_3 . . . 1067
12.88.3.38 verticalSpacer_4 . . . 1068
12.88.3.39 verticalSpacer_5 . . . 1068
12.88.3.40 verticalSpacer_6 . . . 1068
12.88.3.41 verticalSpacer_7 . . . 1068
12.88.3.42 verticalSpacer_8 . . . 1068
12.88.3.43 verticalSpacer_9 . . . 1068
12.89 Ui_FwUpgradeDialog Class Reference . . . 1068
12.89.1 Detailed Description . . . 1070
12.89.2 Member Function Documentation . . . 1070
12.89.2.1 retranslateUi . . . 1070
12.89.2.2 setupUi . . . 1071
12.89.3 Field Documentation . . . 1072
12.89.3.1 actionButton . . . 1073
12.89.3.2 cancelButton . . . 1073
12.89.3.3 curLabel . . . 1073
12.89.3.4 devLabel . . . 1073
12.89.3.5 gridLayout . . . 1073
12.89.3.6 gridLayout_2 . . . 1073
12.89.3.7 horizontalSpacer . . . 1073
12.89.3.8 horizontalSpacer_2 . . . 1073
12.89.3.9 horizontalSpacer_3 . . . 1073
12.89.3.10 horizontalSpacer_4 . . . 1074
12.89.3.11 label_2 . . . 1074
12.89.3.12 label_4 . . . 1074
12.89.3.13 newLabel . . . 1074
12.89.3.14 progressBar . . . 1074
12.89.3.15 statusLabel . . . 1074
12.90 Ui_GradientDialog Class Reference . . . 1074
12.90.1 Detailed Description . . . 1076
12.90.2 Member Function Documentation . . . 1076

12.90.2.1 retranslateUi . . . 1076
12.90.2.2 setupUi . . . 1077
12.90.3 Field Documentation . . . 1079
12.90.3.1 buttonBox . . . 1079
12.90.3.2 gridLayout . . . 1079
12.90.3.3 gridLayout_2 . . . 1079
12.90.3.4 gridLayout_3 . . . 1080
12.90.3.5 groupBox_2 . . . 1080
12.90.3.6 horizontalLayout . . . 1080
12.90.3.7 horizontalLayout_2 . . . 1080
12.90.3.8 horizontalSpacer . . . 1080
12.90.3.9 horizontalSpacer_2 . . . 1080
12.90.3.10 horizontalSpacer_3 . . . 1080
12.90.3.11 label . . . 1080
12.90.3.12 label_2 . . . 1080
12.90.3.13 presetDelete . . . 1081
12.90.3.14 presetList . . . 1081
12.90.3.15 presetName . . . 1081
12.90.3.16 presetSave . . . 1081
12.90.3.17 stopBox . . . 1081
12.90.3.18 stopColor . . . 1081
12.90.3.19 stopOpacity . . . 1081
12.90.3.20 stopPos . . . 1081
12.90.3.21 verticalSpacer . . . 1081
12.90.3.22 verticalSpacer_2 . . . 1082
12.90.3.23 widget . . . 1082
12.91 Ui_KbAnimWidget Class Reference . . . 1082
12.91.1 Detailed Description . . . 1084
12.91.2 Member Function Documentation . . . 1084
12.91.2.1 retranslateUi . . . 1084
12.91.2.2 setupUi . . . 1084
12.91.3 Field Documentation . . . 1087
12.91.3.1 aCopyLabel . . . 1087
12.91.3.2 aMissingLabel . . . 1087
12.91.3.3 aMissingPage . . . 1087
12.91.3.4 aNameLabel . . . 1087
12.91.3.5 animList . . . 1087
12.91.3.6 animPage . . . 1088
12.91.3.7 aVerLabel . . . 1088
12.91.3.8 blendBox . . . 1088

12.91.3.9 deleteButton . . . 1088
12.91.3.10 gridLayout . . . 1088
12.91.3.11 gridLayout_2 . . . 1088
12.91.3.12 gridLayout_3 . . . 1088
12.91.3.13 gridLayout_4 . . . 1088
12.91.3.14 horizontalLayout . . . 1089
12.91.3.15 horizontalSpacer . . . 1089
12.91.3.16 horizontalSpacer_2 . . . 1089
12.91.3.17 horizontalSpacer_3 . . . 1089
12.91.3.18 horizontalSpacer_4 . . . 1089
12.91.3.19 horizontalSpacer_5 . . . 1089
12.91.3.20 keyButton . . . 1089
12.91.3.21 label_5 . . . 1089
12.91.3.22 label_6 . . . 1089
12.91.3.23 label_7 . . . 1090
12.91.3.24 nameBox . . . 1090
12.91.3.25 noAnimLabel . . . 1090
12.91.3.26 noSelPage . . . 1090
12.91.3.27 opacityBox . . . 1090
12.91.3.28 propertyButton . . . 1090
12.91.3.29 selectionStack . . . 1090
12.91.3.30 verticalLayout . . . 1090
12.91.3.31 verticalSpacer . . . 1090
12.91.3.32 verticalSpacer_2 . . . 1091
12.91.3.33 verticalSpacer_3 . . . 1091
12.92 Ui_KbBindWidget Class Reference . . . 1091
12.92.1 Detailed Description . . . 1092
12.92.2 Member Function Documentation . . . 1093
12.92.2.1 retranslateUi . . . 1093
12.92.2.2 setupUi . . . 1093
12.92.3 Field Documentation . . . 1095
12.92.3.1 copyButton . . . 1095
12.92.3.2 gridLayout . . . 1095
12.92.3.3 gridLayout_2 . . . 1095
12.92.3.4 horizontalLayout_2 . . . 1096
12.92.3.5 horizontalSpacer . . . 1096
12.92.3.6 horizontalSpacer_2 . . . 1096
12.92.3.7 keyWidget . . . 1096
12.92.3.8 label . . . 1096
12.92.3.9 label_2 . . . 1096

12.92.3.10 line . . . 1096
12.92.3.11 line_2 . . . 1096
12.92.3.12 rbWidget . . . 1096
12.92.3.13 resetButton . . . 1097
12.92.3.14 selectLabel . . . 1097
12.92.3.15 verticalLayout . . . 1097
12.92.3.16 verticalSpacer . . . 1097
12.92.3.17 widget . . . 1097
12.93 Ui_KbLightWidget Class Reference . . . 1097
12.93.1 Detailed Description . . . 1099
12.93.2 Member Function Documentation . . . 1099
12.93.2.1 retranslateUi . . . 1099
12.93.2.2 setupUi . . . 1100
12.93.3 Field Documentation . . . 1102
12.93.3.1 animButton . . . 1103
12.93.3.2 animWidget . . . 1103
12.93.3.3 bgButton . . . 1103
12.93.3.4 brightnessBox . . . 1103
12.93.3.5 gridLayout . . . 1103
12.93.3.6 gridLayout_2 . . . 1103
12.93.3.7 gridLayout_4 . . . 1103
12.93.3.8 horizontalLayout . . . 1103
12.93.3.9 horizontalLayout_2 . . . 1103
12.93.3.10 horizontalLayout_3 . . . 1104
12.93.3.11 horizontalSpacer_3 . . . 1104
12.93.3.12 horizontalSpacer_4 . . . 1104
12.93.3.13 horizontalSpacer_5 . . . 1104
12.93.3.14 horizontalSpacer_6 . . . 1104
12.93.3.15 keyWidget . . . 1104
12.93.3.16 label . . . 1104
12.93.3.17 label_2 . . . 1104
12.93.3.18 label_3 . . . 1104
12.93.3.19 line . . . 1105
12.93.3.20 line_2 . . . 1105
12.93.3.21 selLabel . . . 1105
12.93.3.22 showAnimBox . . . 1105
12.93.3.23 verticalLayout . . . 1105
12.93.3.24 verticalSpacer_2 . . . 1105
12.93.3.25 verticalSpacer_3 . . . 1105
12.93.3.26 widget . . . 1105

12.94 Ui_KbProfileDialog Class Reference . . . 1105
12.94.1 Detailed Description . . . 1107
12.94.2 Member Function Documentation . . . 1107
12.94.2.1 retranslateUi . . . 1107
12.94.2.2 setupUi . . . 1107
12.94.3 Field Documentation . . . 1108
12.94.3.1 buttonBox . . . 1108
12.94.3.2 gridLayout . . . 1108
12.94.3.3 horizontalSpacer . . . 1109
12.94.3.4 horizontalSpacer_2 . . . 1109
12.94.3.5 profileList . . . 1109
12.95 Ui_KbWidget Class Reference . . . 1109
12.95.1 Detailed Description . . . 1111
12.95.2 Member Function Documentation . . . 1111
12.95.2.1 retranslateUi . . . 1111
12.95.2.2 setupUi . . . 1112
12.95.3 Field Documentation . . . 1115
12.95.3.1 bindTab . . . 1115
12.95.3.2 bindWidget . . . 1115
12.95.3.3 devTab . . . 1115
12.95.3.4 fwLabel . . . 1115
12.95.3.5 fwUpdButton . . . 1115
12.95.3.6 fwUpdLabel . . . 1116
12.95.3.7 fwUpdLayout . . . 1116
12.95.3.8 gridLayout . . . 1116
12.95.3.9 gridLayout_2 . . . 1116
12.95.3.10 gridLayout_3 . . . 1116
12.95.3.11 gridLayout_4 . . . 1116
12.95.3.12 gridLayout_5 . . . 1116
12.95.3.13 gridLayout_7 . . . 1116
12.95.3.14 horizontalSpacer_2 . . . 1116
12.95.3.15 horizontalSpacer_3 . . . 1117
12.95.3.16 hwSaveButton . . . 1117
12.95.3.17 kPerfTab . . . 1117
12.95.3.18 kPerfWidget . . . 1117
12.95.3.19 label_3 . . . 1117
12.95.3.20 label_4 . . . 1117
12.95.3.21 label_9 . . . 1117
12.95.3.22 lightTab . . . 1117
12.95.3.23 lightWidget . . . 1117

12.95.3.24 modesList . . . 1118
12.95.3.25 mPerfTab . . . 1118
12.95.3.26 mPerfWidget . . . 1118
12.95.3.27 pollLabel . . . 1118
12.95.3.28 pollLabel2 . . . 1118
12.95.3.29 profileBox . . . 1118
12.95.3.30 serialLabel . . . 1118
12.95.3.31 tabWidget . . . 1118
12.95.3.32 verticalSpacer . . . 1118
12.96 Ui_KPerfWidget Class Reference . . . 1119
12.96.1 Detailed Description . . . 1122
12.96.2 Member Function Documentation . . . 1122
12.96.2.1 retranslateUi . . . 1122
12.96.2.2 setupUi . . . 1123
12.96.3 Field Documentation . . . 1130
12.96.3.1 capsBox . . . 1130
12.96.3.2 capsColorOff . . . 1130
12.96.3.3 capsColorOn . . . 1130
12.96.3.4 copyButton . . . 1130
12.96.3.5 gridLayout . . . 1130
12.96.3.6 horizontalLayout . . . 1130
12.96.3.7 horizontalLayout_2 . . . 1130
12.96.3.8 horizontalSpacer . . . 1131
12.96.3.9 horizontalSpacer_2 . . . 1131
12.96.3.10 horizontalSpacer_3 . . . 1131
12.96.3.11 horizontalSpacer_4 . . . 1131
12.96.3.12 horizontalSpacer_5 . . . 1131
12.96.3.13 horizontalSpacer_6 . . . 1131
12.96.3.14 horizontalSpacer_7 . . . 1131
12.96.3.15 horizontalSpacer_8 . . . 1131
12.96.3.16 intensityBox . . . 1131
12.96.3.17 k95Label1 . . . 1132
12.96.3.18 k95Label2 . . . 1132
12.96.3.19 k95Label3 . . . 1132
12.96.3.20 k95Label4 . . . 1132
12.96.3.21 k95Label5 . . . 1132
12.96.3.22 k95Label6 . . . 1132
12.96.3.23 k95Line . . . 1132
12.96.3.24 k95Spacer . . . 1132
12.96.3.25 label . . . 1132

12.96.3.26 label_10 . . . 1133
12.96.3.27 label_11 . . . 1133
12.96.3.28 label_12 . . . 1133
12.96.3.29 label_14 . . . 1133
12.96.3.30 label_15 . . . 1133
12.96.3.31 label_16 . . . 1133
12.96.3.32 label_17 . . . 1133
12.96.3.33 label_18 . . . 1133
12.96.3.34 label_19 . . . 1133
12.96.3.35 label_2 . . . 1134
12.96.3.36 label_20 . . . 1134
12.96.3.37 label_21 . . . 1134
12.96.3.38 label_22 . . . 1134
12.96.3.39 label_23 . . . 1134
12.96.3.40 label_24 . . . 1134
12.96.3.41 label_3 . . . 1134
12.96.3.42 label_4 . . . 1134
12.96.3.43 label_5 . . . 1134
12.96.3.44 label_6 . . . 1135
12.96.3.45 label_7 . . . 1135
12.96.3.46 label_8 . . . 1135
12.96.3.47 lightBox . . . 1135
12.96.3.48 lightColor1 . . . 1135
12.96.3.49 lightColor2 . . . 1135
12.96.3.50 lightColor3 . . . 1135
12.96.3.51 line . . . 1135
12.96.3.52 line_2 . . . 1135
12.96.3.53 line_3 . . . 1136
12.96.3.54 lockBox . . . 1136
12.96.3.55 lockColorOff . . . 1136
12.96.3.56 lockColorOn . . . 1136
12.96.3.57 macroBox . . . 1136
12.96.3.58 macroColorOff . . . 1136
12.96.3.59 macroColorOn . . . 1136
12.96.3.60 modeBox . . . 1136
12.96.3.61 modeColorOff . . . 1136
12.96.3.62 modeColorOn . . . 1137
12.96.3.63 muteBox . . . 1137
12.96.3.64 muteColorNA . . . 1137
12.96.3.65 muteColorOff . . . 1137

12.96.3.66 muteColorOn . . . 1137
12.96.3.67 numBox . . . 1137
12.96.3.68 numColorOff . . . 1137
12.96.3.69 numColorOn . . . 1137
12.96.3.70 scrollBox . . . 1137
12.96.3.71 scrollColorOff . . . 1138
12.96.3.72 scrollColorOn . . . 1138
12.96.3.73 verticalSpacer . . . 1138
12.96.3.74 verticalSpacer_2 . . . 1138
12.96.3.75 verticalSpacer_3 . . . 1138
12.96.3.76 verticalSpacer_4 . . . 1138
12.97 Ui_LayoutDialog Class Reference . . . 1138
12.97.1 Detailed Description . . . 1140
12.97.2 Member Function Documentation . . . 1140
12.97.2.1 retranslateUi . . . 1140
12.97.2.2 setupUi . . . 1140
12.97.3 Field Documentation . . . 1142
12.97.3.1 buttonBox . . . 1142
12.97.3.2 comboBox . . . 1142
12.97.3.3 gridLayout . . . 1142
12.97.3.4 horizontalSpacer . . . 1142
12.97.3.5 horizontalSpacer_2 . . . 1142
12.97.3.6 label . . . 1143
12.97.3.7 label_2 . . . 1143
12.97.3.8 verticalSpacer . . . 1143
12.97.3.9 verticalSpacer_2 . . . 1143
12.97.3.10 verticalSpacer_3 . . . 1143
12.97.3.11 verticalSpacer_4 . . . 1143
12.98 Ui_MainWindow Class Reference . . . 1143
12.98.1 Detailed Description . . . 1145
12.98.2 Member Function Documentation . . . 1145
12.98.2.1 retranslateUi . . . 1145
12.98.2.2 setupUi . . . 1145
12.98.3 Field Documentation . . . 1146
12.98.3.1 actionAbout . . . 1146
12.98.3.2 actionExit . . . 1146
12.98.3.3 centralWidget . . . 1146
12.98.3.4 gridLayout . . . 1147
12.98.3.5 tabWidget . . . 1147
12.99 Ui_ModeSelectDialog Class Reference . . . 1147

12.99.1 Detailed Description . . . 1148
12.99.2 Member Function Documentation . . . 1148
12.99.2.1 retranslateUi . . . 1148
12.99.2.2 setupUi . . . 1149
12.99.3 Field Documentation . . . 1150
12.99.3.1 buttonBox . . . 1150
12.99.3.2 gridLayout . . . 1150
12.99.3.3 label . . . 1150
12.99.3.4 modeList . . . 1150
12.99.3.5 selAllButton . . . 1150
12.99.3.6 selNoneButton . . . 1151
12.100 Ui_MPerfWidget Class Reference . . . 1151
12.100.1 Detailed Description . . . 1153
12.100.2 Member Function Documentation . . . 1154
12.100.2.1 retranslateUi . . . 1154
12.100.2.2 setupUi . . . 1155
12.100.3 Field Documentation . . . 1162
12.100.3.1 aSnapBox . . . 1162
12.100.3.2 copyButton . . . 1162
12.100.3.3 eBox1 . . . 1163
12.100.3.4 eBox2 . . . 1163
12.100.3.5 eBox3 . . . 1163
12.100.3.6 eBox4 . . . 1163
12.100.3.7 eBox5 . . . 1163
12.100.3.8 gridLayout . . . 1163
12.100.3.9 horizontalLayout . . . 1163
12.100.3.10 horizontalLayout_2 . . . 1163
12.100.3.11 horizontalLayout_4 . . . 1163
12.100.3.12 horizontalSpacer . . . 1164
12.100.3.13 horizontalSpacer_2 . . . 1164
12.100.3.14 horizontalSpacer_3 . . . 1164
12.100.3.15 horizontalSpacer_4 . . . 1164
12.100.3.16 horizontalSpacer_5 . . . 1164
12.100.3.17 horizontalSpacer_6 . . . 1164
12.100.3.18 horizontalSpacer_7 . . . 1164
12.100.3.19 mButton0 . . . 1164
12.100.3.20 mButton1 . . . 1164
12.100.3.21 mButton2 . . . 1165
12.100.3.22 mButton3 . . . 1165
12.100.3.23 mButton4 . . . 1165

12.100.3.24 Button5 . . . 1165
12.100.3.25 ButtonO . . . 1165
12.100.3.26 odicBox . . . 1165
12.100.3.27 abel . . . 1165
12.100.3.28 abel_11 . . . 1165
12.100.3.29 abel_12 . . . 1165
12.100.3.30 abel_13 . . . 1166
12.100.3.31 abel_14 . . . 1166
12.100.3.32 abel_15 . . . 1166
12.100.3.33 abel_2 . . . 1166
12.100.3.34 abel_25 . . . 1166
12.100.3.35 abel_3 . . . 1166
12.100.3.36 abel_4 . . . 1166
12.100.3.37 abel_5 . . . 1166
12.100.3.38 abel_6 . . . 1166
12.100.3.39 abel_7 . . . 1167
12.100.3.40 abel_8 . . . 1167
12.100.3.41 HeightBox . . . 1167
12.100.3.42 ne . . . 1167
12.100.3.43 ne_2 . . . 1167
12.100.3.44 ne_3 . . . 1167
12.100.3.45 pinBox . . . 1167
12.100.3.46 erticalLayout . . . 1167
12.100.3.47 erticalSpacer . . . 1167
12.100.3.48 erticalSpacer_2 . . . 1168
12.100.3.49 erticalSpacer_3 . . . 1168
12.100.3.50 Box0 . . . 1168
12.100.3.51 Box1 . . . 1168
12.100.3.52 Box2 . . . 1168
12.100.3.53 Box3 . . . 1168
12.100.3.54 Box4 . . . 1168
12.100.3.55 Box5 . . . 1168
12.100.3.56 Slider0 . . . 1168
12.100.3.57 Slider1 . . . 1169
12.100.3.58 Slider2 . . . 1169
12.100.3.59 Slider3 . . . 1169
12.100.3.60 Slider4 . . . 1169
12.100.3.61 Slider5 . . . 1169
12.100.3.62 yBox . . . 1169
12.100.3.63 Box0 . . . 1169

12.100.3.64 Box1 . . . 1169
12.100.3.65 Box2 . . . 1169
12.100.3.66 Box3 . . . 1170
12.100.3.67 Box4 . . . 1170
12.100.3.68 Box5 . . . 1170
12.100.3.69 Slider0 . . . 1170
12.100.3.70 Slider1 . . . 1170
12.100.3.71 Slider2 . . . 1170
12.100.3.72 Slider3 . . . 1170
12.100.3.73 Slider4 . . . 1170
12.100.3.74 Slider5 . . . 1170
12.101 Ui_RebindWidget Class Reference . . . 1171
12.101.1 Detailed Description . . . 1174
12.101.2 Member Function Documentation . . . 1174
12.101.2.1 retranslateUi . . . 1174
12.101.2.2 setupUi . . . 1177
12.101.3 Field Documentation . . . 1186
12.101.3.1 actionTab . . . 1186
12.101.3.2 animBox . . . 1186
12.101.3.3 animButton . . . 1187
12.101.3.4 animKrBox . . . 1187
12.101.3.5 animOnceBox . . . 1187
12.101.3.6 animTab . . . 1187
12.101.3.7 applyButton . . . 1187
12.101.3.8 btnClearMacro . . . 1187
12.101.3.9 btnStartMacro . . . 1187
12.101.3.10 btnStopMacro . . . 1187
12.101.3.11 cancelButton . . . 1188
12.101.3.12 delayButtons . . . 1188
12.101.3.13 dpiBox . . . 1188
12.101.3.14 dpiButton . . . 1188
12.101.3.15 dpiCustLabel . . . 1188
12.101.3.16 dpiCustXBox . . . 1188
12.101.3.17 dpiCustYBox . . . 1188
12.101.3.18 dpiWarning . . . 1188
12.101.3.19 fnBox . . . 1189
12.101.3.20 fnButton . . . 1189
12.101.3.21 gridLayout . . . 1189
12.101.3.22 gridLayout_10 . . . 1189
12.101.3.23 gridLayout_2 . . . 1189

12.101.3.24 gridLayout_3 . . . 1189
12.101.3.25 gridLayout_4 . . . 1189
12.101.3.26 gridLayout_5 . . . 1189
12.101.3.27 gridLayout_8 . . . 1189
12.101.3.28 groupBox_2 . . . 1190
12.101.3.29 groupBox_3 . . . 1190
12.101.3.30 groupBox_4 . . . 1190
12.101.3.31 horizontalLayout . . . 1190
12.101.3.32 horizontalLayout_2 . . . 1190
12.101.3.33 horizontalLayout_3 . . . 1190
12.101.3.34 horizontalSpacer . . . 1190
12.101.3.35 horizontalSpacer_10 . . . 1190
12.101.3.36 horizontalSpacer_11 . . . 1190
12.101.3.37 horizontalSpacer_12 . . . 1191
12.101.3.38 horizontalSpacer_13 . . . 1191
12.101.3.39 horizontalSpacer_14 . . . 1191
12.101.3.40 horizontalSpacer_15 . . . 1191
12.101.3.41 horizontalSpacer_16 . . . 1191
12.101.3.42 horizontalSpacer_17 . . . 1191
12.101.3.43 horizontalSpacer_18 . . . 1191
12.101.3.44 horizontalSpacer_19 . . . 1191
12.101.3.45 horizontalSpacer_2 . . . 1191
12.101.3.46 horizontalSpacer_20 . . . 1192
12.101.3.47 horizontalSpacer_21 . . . 1192
12.101.3.48 horizontalSpacer_22 . . . 1192
12.101.3.49 horizontalSpacer_23 . . . 1192
12.101.3.50 horizontalSpacer_24 . . . 1192
12.101.3.51 horizontalSpacer_25 . . . 1192
12.101.3.52 horizontalSpacer_26 . . . 1192
12.101.3.53 horizontalSpacer_3 . . . 1192
12.101.3.54 horizontalSpacer_4 . . . 1192
12.101.3.55 horizontalSpacer_5 . . . 1193
12.101.3.56 horizontalSpacer_6 . . . 1193
12.101.3.57 horizontalSpacer_7 . . . 1193
12.101.3.58 horizontalSpacer_8 . . . 1193
12.101.3.59 horizontalSpacer_9 . . . 1193
12.101.3.60 keyTab . . . 1193
12.101.3.61 label . . . 1193
12.101.3.62 layoutWidget . . . 1193
12.101.3.63 layoutWidget1 . . . 1193

12.101.3.64 layoutWidget2 . . . 1194
12.101.3.65 layoutWidget_2 . . . 1194
12.101.3.66 lbl_macro . . . 1194
12.101.3.67 lightBox . . . 1194
12.101.3.68 lightButton . . . 1194
12.101.3.69 lightWrapBox . . . 1194
12.101.3.70 lockBox . . . 1194
12.101.3.71 lockButton . . . 1194
12.101.3.72 macroTab . . . 1194
12.101.3.73 mb2Box . . . 1195
12.101.3.74 mb2Button . . . 1195
12.101.3.75 mbBox . . . 1195
12.101.3.76 mbButton . . . 1195
12.101.3.77 mediaBox . . . 1195
12.101.3.78 mediaButton . . . 1195
12.101.3.79 modBox . . . 1195
12.101.3.80 modButton . . . 1195
12.101.3.81 modeBox . . . 1196
12.101.3.82 modeButton . . . 1196
12.101.3.83 modeWrapBox . . . 1196
12.101.3.84 mouseTab . . . 1196
12.101.3.85 numBox . . . 1196
12.101.3.86 numButton . . . 1196
12.101.3.87 programKpBox . . . 1196
12.101.3.88 programKpButton . . . 1196
12.101.3.89 programKpExtra . . . 1197
12.101.3.90 programKpModeBox . . . 1197
12.101.3.91 programKpSIBox . . . 1197
12.101.3.92 programKrBox . . . 1197
12.101.3.93 programKrButton . . . 1197
12.101.3.94 programKrExtra . . . 1197
12.101.3.95 programKrModeBox . . . 1197
12.101.3.96 programKrSIBox . . . 1197
12.101.3.97 programTab . . . 1197
12.101.3.98 progTipLabel . . . 1198
12.101.3.99 pteMacroBox . . . 1198
12.101.3.100 pteMacroComment . . . 1198
12.101.3.101 pteMacroText . . . 1198
12.101.3.102 r_delay_asTyped . . . 1198
12.101.3.103 r_delay_default . . . 1198

12.101.3.104 delay_no . . . 1198
12.101.3.105 resetButton . . . 1198
12.101.3.106 tab . . . 1199
12.101.3.107 txtBuffer . . . 1199
12.101.3.108 typingBox . . . 1199
12.101.3.109 typingButton . . . 1199
12.101.3.110 unbindButton . . . 1199
12.101.3.111 verticalLayout . . . 1199
12.101.3.112 verticalLayout_2 . . . 1199
12.101.3.113 verticalSpacer . . . 1199
12.101.3.114 verticalSpacer_10 . . . 1199
12.101.3.115 verticalSpacer_11 . . . 1200
12.101.3.116 verticalSpacer_2 . . . 1200
12.101.3.117 verticalSpacer_3 . . . 1200
12.101.3.118 verticalSpacer_4 . . . 1200
12.101.3.119 verticalSpacer_5 . . . 1200
12.101.3.120 verticalSpacer_6 . . . 1200
12.101.3.121 verticalSpacer_7 . . . 1200
12.101.3.122 verticalSpacer_8 . . . 1200
12.101.3.123 wheelBox . . . 1200
12.101.3.124 wheelButton . . . 1201
12.102 Ui_SettingsWidget Class Reference . . . 1201
12.102.1 Detailed Description . . . 1203
12.102.2 Member Function Documentation . . . 1203
12.102.2.1 retranslateUi . . . 1203
12.102.2.2 setupUi . . . 1204
12.102.3 Field Documentation . . . 1208
12.102.3.1 altBox . . . 1208
12.102.3.2 altLabel . . . 1208
12.102.3.3 autoFWBox . . . 1208
12.102.3.4 capsBox . . . 1208
12.102.3.5 capsLabel . . . 1208
12.102.3.6 ctrlBox . . . 1209
12.102.3.7 ctrlLabel . . . 1209
12.102.3.8 devicesLabel . . . 1209
12.102.3.9 extraButton . . . 1209
12.102.3.10 gridLayout . . . 1209
12.102.3.11 gridLayout_3 . . . 1209
12.102.3.12 gridLayout_5 . . . 1209
12.102.3.13 horizontalLayout . . . 1209

12.102.3.14 horizontalLayout_2 . . . 1209
12.102.3.15 horizontalSpacer . . . 1210
12.102.3.16 horizontalSpacer_10 . . . 1210
12.102.3.17 horizontalSpacer_2 . . . 1210
12.102.3.18 horizontalSpacer_7 . . . 1210
12.102.3.19 horizontalSpacer_9 . . . 1210
12.102.3.20 label . . . 1210
12.102.3.21 label_2 . . . 1210
12.102.3.22 label_3 . . . 1210
12.102.3.23 label_5 . . . 1210
12.102.3.24 label_6 . . . 1211
12.102.3.25 label_8 . . . 1211
12.102.3.26 layoutBox . . . 1211
12.102.3.27 line . . . 1211
12.102.3.28 line_2 . . . 1211
12.102.3.29 line_3 . . . 1211
12.102.3.30 loginItemBox . . . 1211
12.102.3.31 pushButton . . . 1211
12.102.3.32 shiftBox . . . 1211
12.102.3.33 shiftLabel . . . 1212
12.102.3.34 versionLabel . . . 1212
12.102.3.35 verticalSpacer . . . 1212
12.102.3.36 verticalSpacer_10 . . . 1212
12.102.3.37 verticalSpacer_2 . . . 1212
12.102.3.38 verticalSpacer_3 . . . 1212
12.102.3.39 verticalSpacer_4 . . . 1212
12.102.3.40 verticalSpacer_5 . . . 1212
12.102.3.41 verticalSpacer_6 . . . 1212
12.102.3.42 verticalSpacer_7 . . . 1213
12.102.3.43 verticalSpacer_8 . . . 1213
12.102.3.44 winBox . . . 1213
12.102.3.45 winLabel . . . 1213
12.103 UsbId Struct Reference . . . 1213
12.103.1 Detailed Description . . . 1215
12.103.2 Constructor & Destructor Documentation . . . 1215
12.103.2.1 UsbId . . . 1215
12.103.2.2 UsbId . . . 1215
12.103.2.3 UsbId . . . 1215
12.103.3 Member Function Documentation . . . 1215
12.103.3.1 guidString . . . 1215

12.103.3.2 guidString . . . 1215
12.103.3.3 hwModifiedString . . . 1216
12.103.3.4 hwModifiedString . . . 1216
12.103.3.5 modifiedString . . . 1216
12.103.3.6 modifiedString . . . 1217
12.103.3.7 newGuid . . . 1217
12.103.3.8 newModified . . . 1217
12.103.4 Field Documentation . . . 1218
12.103.4.1 guid . . . 1218
12.103.4.2 hwModified . . . 1218
12.103.4.3 modified . . . 1218

13 File Documentation **1219**

13.1 BUILD.md File Reference . . . 1219
13.2 DAEMON.md File Reference . . . 1219
13.3 README.md File Reference . . . 1219
13.4 ROADMAP.md File Reference . . . 1219
13.5 src/ckb-daemon/command.c File Reference . . . 1219
13.5.1 Macro Definition Documentation . . . 1220
13.5.1.1 TRY_WITH_RESET . . . 1220
13.5.2 Function Documentation . . . 1220
13.5.2.1 readcmd . . . 1220
13.5.3 Variable Documentation . . . 1224
13.5.3.1 cmd_strings . . . 1224
13.6 src/ckb-daemon/command.h File Reference . . . 1224
13.6.1 Data Structure Documentation . . . 1226
13.6.1.1 union devcmd . . . 1226
13.6.2 Macro Definition Documentation . . . 1226
13.6.2.1 CMD_COUNT . . . 1226
13.6.2.2 CMD_DEV_COUNT . . . 1227
13.6.3 Typedef Documentation . . . 1227
13.6.3.1 cmdhandler . . . 1227
13.6.3.2 cmdhandler_io . . . 1227
13.6.3.3 cmdhandler_mac . . . 1227
13.6.3.4 devcmd . . . 1227
13.6.4 Enumeration Type Documentation . . . 1227
13.6.4.1 cmd . . . 1227
13.6.5 Function Documentation . . . 1229
13.6.5.1 readcmd . . . 1229
13.7 src/ckb-daemon/device.c File Reference . . . 1233

13.7.1 Function Documentation . . . 1234
13.7.1.1 _start_dev . . . 1234
13.7.1.2 start_dev . . . 1236
13.7.2 Variable Documentation . . . 1236
13.7.2.1 devlistmutex . . . 1236
13.7.2.2 devmutex . . . 1236
13.7.2.3 hwload_mode . . . 1236
13.7.2.4 inputmutex . . . 1236
13.7.2.5 keyboard . . . 1236
13.7.2.6 macromutex . . . 1237
13.7.2.7 macromutex2 . . . 1237
13.7.2.8 macrovar . . . 1237
13.8 src/ckb-daemon/device.h File Reference . . . 1237
13.8.1 Macro Definition Documentation . . . 1238
13.8.1.1 ACT_LIGHT . . . 1238
13.8.1.2 ACT_LOCK . . . 1238
13.8.1.3 ACT_M1 . . . 1239
13.8.1.4 ACT_M2 . . . 1239
13.8.1.5 ACT_M3 . . . 1239
13.8.1.6 ACT_MR_RING . . . 1239
13.8.1.7 ACT_NEXT . . . 1239
13.8.1.8 ACT_NEXT_NOWRAP . . . 1239
13.8.1.9 DEV_MAX . . . 1239
13.8.1.10 dmutex . . . 1239
13.8.1.11 imutex . . . 1239
13.8.1.12 IN_CORSAIR . . . 1239
13.8.1.13 IN_HID . . . 1240
13.8.1.14 IS_CONNECTED . . . 1240
13.8.1.15 mmutex . . . 1240
13.8.1.16 mmutex2 . . . 1240
13.8.1.17 mvar . . . 1240
13.8.1.18 setactive . . . 1240
13.8.2 Function Documentation . . . 1240
13.8.2.1 cmd_active_kb . . . 1240
13.8.2.2 cmd_active_mouse . . . 1241
13.8.2.3 cmd_idle_kb . . . 1241
13.8.2.4 cmd_idle_mouse . . . 1242
13.8.2.5 cmd_pollrate . . . 1242
13.8.2.6 setactive_kb . . . 1243
13.8.2.7 setactive_mouse . . . 1244

13.8.2.8 setmodeindex_nrgb . . . 1246
13.8.2.9 start_dev . . . 1246
13.8.2.10 start_kb_nrgb . . . 1246
13.8.3 Variable Documentation . . . 1247
13.8.3.1 devmutex . . . 1247
13.8.3.2 inputmutex . . . 1247
13.8.3.3 keyboard . . . 1247
13.8.3.4 macromutex . . . 1247
13.8.3.5 macromutex2 . . . 1247
13.8.3.6 macrovar . . . 1247
13.9 src/ckb-daemon/device_keyboard.c File Reference . . . 1247
13.9.1 Function Documentation . . . 1248
13.9.1.1 cmd_active_kb . . . 1248
13.9.1.2 cmd_idle_kb . . . 1249
13.9.1.3 setactive_kb . . . 1249
13.9.1.4 setmodeindex_nrgb . . . 1251
13.9.1.5 start_kb_nrgb . . . 1251
13.10 src/ckb-daemon/device_mouse.c File Reference . . . 1252
13.10.1 Function Documentation . . . 1252
13.10.1.1 cmd_active_mouse . . . 1252
13.10.1.2 cmd_idle_mouse . . . 1253
13.10.1.3 cmd_pollrate . . . 1253
13.10.1.4 setactive_mouse . . . 1254
13.11 src/ckb-daemon/device_vtable.c File Reference . . . 1255
13.11.1 Function Documentation . . . 1256
13.11.1.1 cmd_io_none . . . 1256
13.11.1.2 cmd_macro_none . . . 1256
13.11.1.3 cmd_none . . . 1256
13.11.1.4 int1_int_none . . . 1256
13.11.1.5 int1_void_none . . . 1257
13.11.1.6 loadprofile_none . . . 1257
13.11.2 Variable Documentation . . . 1257
13.11.2.1 vtable_keyboard . . . 1257
13.11.2.2 vtable_keyboard_nonrgb . . . 1257
13.11.2.3 vtable_mouse . . . 1257
13.12 src/ckb-daemon/devnode.c File Reference . . . 1257
13.12.1 Data Structure Documentation . . . 1259
13.12.1.1 struct _readlines_ctx . . . 1259
13.12.2 Macro Definition Documentation . . . 1259
13.12.2.1 MAX_BUFFER . . . 1259

13.12.2.2 S_GID_READ . . . 1259
13.12.3 Function Documentation . . . 1259
13.12.3.1 _mkdevpath . . . 1260
13.12.3.2 _mknotifynode . . . 1262
13.12.3.3 _rmnotifynode . . . 1263
13.12.3.4 _updateconnected . . . 1263
13.12.3.5 mkdevpath . . . 1264
13.12.3.6 mkfwnode . . . 1265
13.12.3.7 mknotifynode . . . 1266
13.12.3.8 readlines . . . 1266
13.12.3.9 readlines_ctx_free . . . 1267
13.12.3.10 readlines_ctx_init . . . 1268
13.12.3.11 rm_recursive . . . 1268
13.12.3.12 rmdevpath . . . 1269
13.12.3.13 rmnotifynode . . . 1270
13.12.3.14 updateconnected . . . 1270
13.12.4 Variable Documentation . . . 1271
13.12.4.1 devpath . . . 1271
13.12.4.2 gid . . . 1271
13.13 src/ckb-daemon/devnode.h File Reference . . . 1271
13.13.1 Macro Definition Documentation . . . 1273
13.13.1.1 S_CUSTOM . . . 1273
13.13.1.2 S_CUSTOM_R . . . 1273
13.13.1.3 S_READ . . . 1273
13.13.1.4 S_READDIR . . . 1273
13.13.1.5 S_READWRITE . . . 1273
13.13.2 Typedef Documentation . . . 1273
13.13.2.1 readlines_ctx . . . 1273
13.13.3 Function Documentation . . . 1273
13.13.3.1 mkdevpath . . . 1273
13.13.3.2 mkfwnode . . . 1274
13.13.3.3 mknotifynode . . . 1275
13.13.3.4 readlines . . . 1276
13.13.3.5 readlines_ctx_free . . . 1276
13.13.3.6 readlines_ctx_init . . . 1277
13.13.3.7 rmdevpath . . . 1277
13.13.3.8 rmnotifynode . . . 1278
13.13.3.9 updateconnected . . . 1279
13.13.4 Variable Documentation . . . 1279
13.13.4.1 devpath . . . 1279

13.13.4.2 gid . . . 1279
13.14 src/ckb-daemon/dpi.c File Reference . . . 1279
13.14.1 Function Documentation . . . 1280
13.14.1.1 cmd_dpi . . . 1280
13.14.1.2 cmd_dpisel . . . 1281
13.14.1.3 cmd_lift . . . 1281
13.14.1.4 cmd_snap . . . 1281
13.14.1.5 loaddpi . . . 1282
13.14.1.6 printdpi . . . 1283
13.14.1.7 savedpi . . . 1283
13.14.1.8 updatedpi . . . 1284
13.15 src/ckb-daemon/dpi.h File Reference . . . 1285
13.15.1 Function Documentation . . . 1286
13.15.1.1 cmd_dpi . . . 1286
13.15.1.2 cmd_dpisel . . . 1287
13.15.1.3 cmd_lift . . . 1287
13.15.1.4 cmd_snap . . . 1287
13.15.1.5 loaddpi . . . 1287
13.15.1.6 printdpi . . . 1288
13.15.1.7 savedpi . . . 1289
13.15.1.8 updatedpi . . . 1290
13.16 src/ckb-daemon/extra_mac.c File Reference . . . 1291
13.17 src/ckb-daemon/firmware.c File Reference . . . 1291
13.17.1 Macro Definition Documentation . . . 1292
13.17.1.1 FW_MAXSIZE . . . 1292
13.17.1.2 FW_NOFILE . . . 1292
13.17.1.3 FW_OK . . . 1292
13.17.1.4 FW_USBFAIL . . . 1292
13.17.1.5 FW_WRONGDEV . . . 1293
13.17.2 Function Documentation . . . 1293
13.17.2.1 cmd_fwupdate . . . 1293
13.17.2.2 fwupdate . . . 1293
13.17.2.3 getfwversion . . . 1295
13.18 src/ckb-daemon/firmware.h File Reference . . . 1296
13.18.1 Function Documentation . . . 1297
13.18.1.1 cmd_fwupdate . . . 1297
13.18.1.2 getfwversion . . . 1298
13.19 src/ckb-daemon/includes.h File Reference . . . 1299
13.19.1 Macro Definition Documentation . . . 1300
13.19.1.1 __FILE_NOPATH__ . . . 1300

13.19.1.2 ckb_err . . . 1300
13.19.1.3 ckb_err_fn . . . 1301
13.19.1.4 ckb_err_nofile . . . 1301
13.19.1.5 ckb_fatal . . . 1301
13.19.1.6 ckb_fatal_fn . . . 1301
13.19.1.7 ckb_fatal_nofile . . . 1301
13.19.1.8 ckb_info . . . 1301
13.19.1.9 ckb_info_fn . . . 1301
13.19.1.10 ckb_info_nofile . . . 1301
13.19.1.11 ckb_s_err . . . 1301
13.19.1.12 ckb_s_out . . . 1301
13.19.1.13 ckb_warn . . . 1302
13.19.1.14 ckb_warn_fn . . . 1302
13.19.1.15 ckb_warn_nofile . . . 1302
13.19.1.16 INDEX_OF . . . 1302
13.19.1.17 timespec_eq . . . 1302
13.19.1.18 timespec_ge . . . 1302
13.19.1.19 timespec_gt . . . 1302
13.19.1.20 timespec_le . . . 1302
13.19.1.21 timespec_lt . . . 1302
13.19.2 Typedef Documentation . . . 1303
13.19.2.1 uchar . . . 1303
13.19.2.2 ushort . . . 1303
13.19.3 Function Documentation . . . 1303
13.19.3.1 timespec_add . . . 1303
13.20 src/ckb-daemon/input.c File Reference . . . 1303
13.20.1 Macro Definition Documentation . . . 1304
13.20.1.1 IS_WHEEL . . . 1304
13.20.2 Function Documentation . . . 1304
13.20.2.1 _cmd_macro . . . 1304
13.20.2.2 cmd_bind . . . 1306
13.20.2.3 cmd_macro . . . 1307
13.20.2.4 cmd_rebind . . . 1307
13.20.2.5 cmd_unbind . . . 1307
13.20.2.6 freebind . . . 1308
13.20.2.7 initbind . . . 1308
13.20.2.8 inputupdate . . . 1309
13.20.2.9 inputupdate_keys . . . 1310
13.20.2.10 macro_pt_dequeue . . . 1312
13.20.2.11 macro_pt_enqueue . . . 1313

13.20.2.12 macro_pt_first . . . 1313
13.20.2.13 macromask . . . 1314
13.20.2.14 play_macro . . . 1314
13.20.2.15 updateindicators_kb . . . 1316
13.20.3 Variable Documentation . . . 1317
13.20.3.1 pt_head . . . 1317
13.20.3.2 pt_tail . . . 1317
13.21 src/ckb-daemon/input.h File Reference . . . 1317
13.21.1 Data Structure Documentation . . . 1319
13.21.1.1 struct parameter . . . 1319
13.21.1.2 struct ptlist . . . 1319
13.21.2 Macro Definition Documentation . . . 1320
13.21.2.1 IS_MOD . . . 1320
13.21.3 Typedef Documentation . . . 1320
13.21.3.1 parameter_t . . . 1320
13.21.3.2 ptlist_t . . . 1320
13.21.4 Function Documentation . . . 1320
13.21.4.1 cmd_bind . . . 1320
13.21.4.2 cmd_macro . . . 1321
13.21.4.3 cmd_rebind . . . 1321
13.21.4.4 cmd_unbind . . . 1322
13.21.4.5 freebind . . . 1322
13.21.4.6 initbind . . . 1322
13.21.4.7 inputupdate . . . 1323
13.21.4.8 os_inputclose . . . 1324
13.21.4.9 os_inputopen . . . 1324
13.21.4.10 os_keypress . . . 1325
13.21.4.11 os_mousemove . . . 1326
13.21.4.12 os_setupindicators . . . 1327
13.21.4.13 updateindicators_kb . . . 1328
13.22 src/ckb-daemon/input_linux.c File Reference . . . 1328
13.22.1 Function Documentation . . . 1329
13.22.1.1 _ledthread . . . 1329
13.22.1.2 isync . . . 1330
13.22.1.3 os_inputclose . . . 1330
13.22.1.4 os_inputopen . . . 1331
13.22.1.5 os_keypress . . . 1332
13.22.1.6 os_mousemove . . . 1333
13.22.1.7 os_setupindicators . . . 1333
13.22.1.8 uinputopen . . . 1334

13.23 src/ckb-daemon/input_mac.c File Reference . . . 1335
13.24 src/ckb-daemon/input_mac_mouse.c File Reference . . . 1335
13.25 src/ckb-daemon/keymap.c File Reference . . . 1335
13.25.1 Macro Definition Documentation . . . 1336
13.25.1.1 BUTTON_HID_COUNT . . . 1336
13.25.2 Function Documentation . . . 1336
13.25.2.1 corsair_kbcopy . . . 1336
13.25.2.2 corsair_mousecopy . . . 1337
13.25.2.3 hid_kb_translate . . . 1337
13.25.2.4 hid_mouse_translate . . . 1339
13.25.3 Variable Documentation . . . 1340
13.25.3.1 keymap . . . 1340
13.26 src/ckb-daemon/keymap_mac.h File Reference . . . 1340
13.27 src/ckb-daemon/led.c File Reference . . . 1341
13.27.1 Function Documentation . . . 1341
13.27.1.1 cmd_iauto . . . 1341
13.27.1.2 cmd_inotify . . . 1342
13.27.1.3 cmd_ioff . . . 1342
13.27.1.4 cmd_ion . . . 1343
13.27.1.5 cmd_rgb . . . 1343
13.27.1.6 has_key . . . 1344
13.27.1.7 iselect . . . 1344
13.27.1.8 printrgb . . . 1345
13.28 src/ckb-daemon/led.h File Reference . . . 1346
13.28.1 Function Documentation . . . 1347
13.28.1.1 cmd_iauto . . . 1347
13.28.1.2 cmd_inotify . . . 1348
13.28.1.3 cmd_ioff . . . 1348
13.28.1.4 cmd_ion . . . 1349
13.28.1.5 cmd_rgb . . . 1349
13.28.1.6 loadrgb_kb . . . 1350
13.28.1.7 loadrgb_mouse . . . 1352
13.28.1.8 printrgb . . . 1352
13.28.1.9 savergb_kb . . . 1354
13.28.1.10 savergb_mouse . . . 1355
13.28.1.11 updatergb_kb . . . 1356
13.28.1.12 updatergb_mouse . . . 1357
13.29 src/ckb-daemon/led_keyboard.c File Reference . . . 1358
13.29.1 Macro Definition Documentation . . . 1381
13.29.1.1 BR1 . . . 1381

13.29.1.2 BR2 . . . 1381
13.29.1.3 BR4 . . . 1381
13.29.1.4 O0 . . . 1381
13.29.1.5 O1 . . . 1381
13.29.1.6 O2 . . . 1382
13.29.1.7 O3 . . . 1382
13.29.1.8 O4 . . . 1382
13.29.1.9 O5 . . . 1382
13.29.1.10 O6 . . . 1382
13.29.1.11 O7 . . . 1382
13.29.1.12 O8 . . . 1382
13.29.2 Function Documentation . . . 1382
13.29.2.1 loadrgb_kb . . . 1382
13.29.2.2 makergb_512 . . . 1384
13.29.2.3 makergb_full . . . 1385
13.29.2.4 ordered8to3 . . . 1385
13.29.2.5 quantize8to3 . . . 1386
13.29.2.6 rgbcmp . . . 1386
13.29.2.7 savergb_kb . . . 1387
13.29.2.8 updatergb_kb . . . 1388
13.29.3 Variable Documentation . . . 1390
13.29.3.1 bit_reverse_table . . . 1391
13.30 src/ckb-daemon/led_mouse.c File Reference . . . 1392
13.30.1 Function Documentation . . . 1392
13.30.1.1 isblack . . . 1392
13.30.1.2 loadrgb_mouse . . . 1393
13.30.1.3 rgbcmp . . . 1393
13.30.1.4 savergb_mouse . . . 1394
13.30.1.5 updatergb_mouse . . . 1395
13.31 src/ckb-daemon/main.c File Reference . . . 1396
13.31.1 Function Documentation . . . 1396
13.31.1.1 localecase . . . 1396
13.31.1.2 main . . . 1397
13.31.1.3 quit . . . 1399
13.31.1.4 quitWithLock . . . 1400
13.31.1.5 restart . . . 1401
13.31.1.6 sighandler . . . 1402
13.31.1.7 sighandler2 . . . 1403
13.31.1.8 timespec_add . . . 1403
13.31.2 Variable Documentation . . . 1404

13.31.2.1 features_mask . . . 1404
13.31.2.2 hwload_mode . . . 1404
13.31.2.3 main_ac . . . 1404
13.31.2.4 main_av . . . 1404
13.31.2.5 reset_stop . . . 1404
13.32 src/ckb-gradient/main.c File Reference . . . 1404
13.32.1 Data Structure Documentation . . . 1405
13.32.1.1 struct keyAnim . . . 1405
13.32.2 Macro Definition Documentation . . . 1406
13.32.2.1 HOLD . . . 1406
13.32.2.2 NONE . . . 1406
13.32.3 Function Documentation . . . 1406
13.32.3.1 ckb_frame . . . 1407
13.32.3.2 ckb_info . . . 1407
13.32.3.3 ckb_init . . . 1408
13.32.3.4 ckb_keypress . . . 1408
13.32.3.5 ckb_parameter . . . 1409
13.32.3.6 ckb_start . . . 1409
13.32.3.7 ckb_time . . . 1410
13.32.4 Variable Documentation . . . 1410
13.32.4.1 anim . . . 1410
13.32.4.2 animcolor . . . 1410
13.32.4.3 kphold . . . 1410
13.32.4.4 kprelease . . . 1410
13.32.4.5 randomize . . . 1410
13.33 src/ckb-heat/main.c File Reference . . . 1411
13.33.1 Data Structure Documentation . . . 1411
13.33.1.1 struct keyanim . . . 1411
13.33.2 Function Documentation . . . 1412
13.33.2.1 anim_add . . . 1412
13.33.2.2 anim_remove . . . 1413
13.33.2.3 ckb_frame . . . 1413
13.33.2.4 ckb_info . . . 1414
13.33.2.5 ckb_init . . . 1414
13.33.2.6 ckb_keypress . . . 1415
13.33.2.7 ckb_parameter . . . 1415
13.33.2.8 ckb_start . . . 1415
13.33.2.9 ckb_time . . . 1416
13.33.3 Variable Documentation . . . 1416
13.33.3.1 animcolor . . . 1416

13.33.3.2 anims . . . 1416
13.33.3.3 ffade . . . 1416
13.33.3.4 pressestofull . . . 1416
13.33.3.5 randomBright . . . 1416
13.34 src/ckb-mviz/main.c File Reference . . . 1416
13.34.1 Function Documentation . . . 1417
13.34.1.1 anim_add . . . 1417
13.34.1.2 anim_remove . . . 1418
13.34.1.3 ckb_frame . . . 1418
13.34.1.4 ckb_info . . . 1418
13.34.1.5 ckb_init . . . 1419
13.34.1.6 ckb_keypress . . . 1419
13.34.1.7 ckb_parameter . . . 1419
13.34.1.8 ckb_start . . . 1419
13.34.1.9 ckb_time . . . 1420
13.34.1.10 getFreqDec . . . 1420
13.34.1.11 max . . . 1421
13.34.1.12 min . . . 1421
13.34.2 Variable Documentation . . . 1422
13.34.2.1 animcolor . . . 1422
13.34.2.2 gcounter . . . 1422
13.34.2.3 inbuf . . . 1422
13.34.2.4 outbuf . . . 1422
13.34.2.5 pas . . . 1422
13.34.2.6 power . . . 1422
13.34.2.7 powers . . . 1422
13.35 src/ckb-pinwheel/main.c File Reference . . . 1422
13.35.1 Macro Definition Documentation . . . 1423
13.35.1.1 ANGLE . . . 1423
13.35.2 Function Documentation . . . 1424
13.35.2.1 ckb_frame . . . 1424
13.35.2.2 ckb_info . . . 1424
13.35.2.3 ckb_init . . . 1425
13.35.2.4 ckb_keypress . . . 1425
13.35.2.5 ckb_parameter . . . 1425
13.35.2.6 ckb_start . . . 1426
13.35.2.7 ckb_time . . . 1426
13.35.3 Variable Documentation . . . 1426
13.35.3.1 animcolor . . . 1426
13.35.3.2 animlength . . . 1426

13.35.3.3 counter_clock . . . 1426
13.35.3.4 frame . . . 1426
13.35.3.5 symmetric . . . 1426
13.35.3.6 x . . . 1427
13.35.3.7 x_offset . . . 1427
13.35.3.8 y . . . 1427
13.35.3.9 y_offset . . . 1427
13.36 src/ckb-rain/main.c File Reference . . . 1427
13.36.1 Macro Definition Documentation . . . 1428
13.36.1.1 DROP_MAX . . . 1428
13.36.2 Function Documentation . . . 1428
13.36.2.1 ckb_frame . . . 1428
13.36.2.2 ckb_info . . . 1429
13.36.2.3 ckb_init . . . 1430
13.36.2.4 ckb_keypress . . . 1430
13.36.2.5 ckb_parameter . . . 1430
13.36.2.6 ckb_start . . . 1431
13.36.2.7 ckb_time . . . 1431
13.36.2.8 drop_add . . . 1431
13.36.3 Variable Documentation . . . 1432
13.36.3.1 aa . . . 1432
13.36.3.2 ab . . . 1432
13.36.3.3 ag . . . 1432
13.36.3.4 ar . . . 1432
13.36.3.5 drop . . . 1432
13.36.3.6 maxsize . . . 1433
13.36.3.7 period . . . 1433
13.36.3.8 spawn . . . 1433
13.36.3.9 speed . . . 1433
13.36.3.10 tick . . . 1433
13.37 src/ckb-random/main.c File Reference . . . 1433
13.37.1 Data Structure Documentation . . . 1434
13.37.1.1 struct rgb . . . 1434
13.37.2 Function Documentation . . . 1435
13.37.2.1 ckb_frame . . . 1435
13.37.2.2 ckb_info . . . 1435
13.37.2.3 ckb_init . . . 1436
13.37.2.4 ckb_keypress . . . 1436
13.37.2.5 ckb_parameter . . . 1436
13.37.2.6 ckb_start . . . 1436

13.37.2.7 ckb_time . . . 1437
13.37.2.8 newtarget . . . 1437
13.37.3 Variable Documentation . . . 1438
13.37.3.1 current . . . 1438
13.37.3.2 fadein . . . 1438
13.37.3.3 phase . . . 1438
13.37.3.4 target . . . 1438
13.37.3.5 useopacity . . . 1438
13.38 src/ckb-ripple/main.c File Reference . . . 1438
13.38.1 Macro Definition Documentation . . . 1440
13.38.1.1 ANIM_MAX . . . 1440
13.38.2 Function Documentation . . . 1440
13.38.2.1 anim_add . . . 1440
13.38.2.2 anim_remove . . . 1440
13.38.2.3 ckb_frame . . . 1440
13.38.2.4 ckb_info . . . 1441
13.38.2.5 ckb_init . . . 1442
13.38.2.6 ckb_keypress . . . 1442
13.38.2.7 ckb_parameter . . . 1442
13.38.2.8 ckb_start . . . 1443
13.38.2.9 ckb_time . . . 1443
13.38.3 Variable Documentation . . . 1443
13.38.3.1 anim . . . 1444
13.38.3.2 animcolor . . . 1444
13.38.3.3 animlength . . . 1444
13.38.3.4 kbsize . . . 1444
13.38.3.5 kprelease . . . 1444
13.38.3.6 randomize . . . 1444
13.38.3.7 symmetric . . . 1444
13.38.3.8 x_offset . . . 1444
13.38.3.9 y_offset . . . 1444
13.39 src/ckb-wave/main.c File Reference . . . 1444
13.39.1 Macro Definition Documentation . . . 1446
13.39.1.1 ANIM_MAX . . . 1446
13.39.2 Function Documentation . . . 1446
13.39.2.1 anim_add . . . 1446
13.39.2.2 anim_remove . . . 1446
13.39.2.3 ckb_frame . . . 1446
13.39.2.4 ckb_info . . . 1447
13.39.2.5 ckb_init . . . 1448

13.39.2.6 ckb_keypress . . . 1449
13.39.2.7 ckb_parameter . . . 1449
13.39.2.8 ckb_start . . . 1450
13.39.2.9 ckb_time . . . 1451
13.39.3 Variable Documentation . . . 1452
13.39.3.1 angle . . . 1452
13.39.3.2 anim . . . 1452
13.39.3.3 animcolor . . . 1452
13.39.3.4 animlength . . . 1452
13.39.3.5 kprelease . . . 1452
13.39.3.6 left . . . 1452
13.39.3.7 symmetric . . . 1452
13.39.3.8 top . . . 1452
13.39.3.9 width . . . 1452
13.40 src/ckb-daemon/notify.c File Reference . . . 1452
13.40.1 Macro Definition Documentation . . . 1453
13.40.1.1 HW_STANDARD . . . 1453
13.40.1.2 HWMODE_OR_RETURN . . . 1454
13.40.2 Function Documentation . . . 1454
13.40.2.1 _cmd_get . . . 1454
13.40.2.2 cmd_get . . . 1456
13.40.2.3 cmd_notify . . . 1457
13.40.2.4 cmd_restart . . . 1457
13.40.2.5 nprintf . . . 1458
13.40.2.6 nprintind . . . 1459
13.40.2.7 nprintkey . . . 1459
13.40.2.8 restart . . . 1460
13.41 src/ckb-daemon/notify.h File Reference . . . 1461
13.41.1 Function Documentation . . . 1462
13.41.1.1 cmd_get . . . 1462
13.41.1.2 cmd_notify . . . 1463
13.41.1.3 cmd_restart . . . 1463
13.41.1.4 nprintf . . . 1463
13.41.1.5 nprintind . . . 1464
13.41.1.6 nprintkey . . . 1465
13.42 src/ckb-daemon/os.h File Reference . . . 1466
13.42.1 Macro Definition Documentation . . . 1466
13.42.1.1 _DEFAULT_SOURCE . . . 1466
13.42.1.2 _GNU_SOURCE . . . 1467
13.42.1.3 euid_guard_start . . . 1467

13.42.1.4 euid_guard_stop . . . 1467
13.42.1.5 UINPUT_VERSION . . . 1467
13.43 src/ckb-daemon/profile.c File Reference . . . 1467
13.43.1 Function Documentation . . . 1468
13.43.1.1 _freeprofile . . . 1468
13.43.1.2 allocprofile . . . 1469
13.43.1.3 cmd_erase . . . 1470
13.43.1.4 cmd_eraseprofile . . . 1470
13.43.1.5 cmd_id . . . 1471
13.43.1.6 cmd_name . . . 1471
13.43.1.7 cmd_profileid . . . 1472
13.43.1.8 cmd_profilename . . . 1472
13.43.1.9 freemode . . . 1473
13.43.1.10 freeprofile . . . 1473
13.43.1.11 gethwmodename . . . 1474
13.43.1.12 gethwprofilename . . . 1474
13.43.1.13 getid . . . 1475
13.43.1.14 getmodename . . . 1476
13.43.1.15 getprofilename . . . 1476
13.43.1.16 hwtonative . . . 1477
13.43.1.17 initmode . . . 1478
13.43.1.18 loadprofile . . . 1479
13.43.1.19 nativetohw . . . 1479
13.43.1.20 printname . . . 1479
13.43.1.21 setid . . . 1480
13.43.1.22 u16dec . . . 1481
13.43.1.23 u16enc . . . 1481
13.43.1.24 urldecode2 . . . 1482
13.43.1.25 urlencode2 . . . 1483
13.43.2 Variable Documentation . . . 1483
13.43.2.1 utf16to8 . . . 1484
13.43.2.2 utf8to16 . . . 1484
13.44 src/ckb-daemon/profile.h File Reference . . . 1484
13.44.1 Macro Definition Documentation . . . 1485
13.44.1.1 hwloadprofile . . . 1485
13.44.2 Function Documentation . . . 1485
13.44.2.1 allocprofile . . . 1485
13.44.2.2 cmd_erase . . . 1486
13.44.2.3 cmd_eraseprofile . . . 1486
13.44.2.4 cmd_hwload_kb . . . 1487

13.44.2.5 cmd_hwload_mouse . . . 1488
13.44.2.6 cmd_hwsave_kb . . . 1489
13.44.2.7 cmd_hwsave_mouse . . . 1490
13.44.2.8 cmd_id . . . 1491
13.44.2.9 cmd_name . . . 1491
13.44.2.10 cmd_profileid . . . 1492
13.44.2.11 cmd_profilename . . . 1492
13.44.2.12 freeprofile . . . 1493
13.44.2.13 gethwmodename . . . 1493
13.44.2.14 gethwprofilename . . . 1494
13.44.2.15 getid . . . 1495
13.44.2.16 getmodename . . . 1495
13.44.2.17 getprofilename . . . 1496
13.44.2.18 hwtonative . . . 1497
13.44.2.19 loadprofile . . . 1497
13.44.2.20 nativetohw . . . 1498
13.44.2.21 setid . . . 1498
13.45 src/ckb-daemon/profile_keyboard.c File Reference . . . 1499
13.45.1 Function Documentation . . . 1499
13.45.1.1 cmd_hwload_kb . . . 1499
13.45.1.2 cmd_hwsave_kb . . . 1500
13.45.1.3 hwloadmode . . . 1501
13.46 src/ckb-daemon/profile_mouse.c File Reference . . . 1502
13.46.1 Function Documentation . . . 1502
13.46.1.1 cmd_hwload_mouse . . . 1503
13.46.1.2 cmd_hwsave_mouse . . . 1504
13.47 src/ckb-daemon/structures.h File Reference . . . 1504
13.47.1 Data Structure Documentation . . . 1506
13.47.1.1 struct usbid . . . 1506
13.47.1.2 struct macroaction . . . 1507
13.47.1.3 struct keymacro . . . 1508
13.47.1.4 struct binding . . . 1509
13.47.1.5 struct dpiset . . . 1510
13.47.1.6 struct lighting . . . 1511
13.47.1.7 struct usbmode . . . 1512
13.47.1.8 struct usbprofile . . . 1513
13.47.1.9 struct hwprofile . . . 1514
13.47.1.10 struct usbinput . . . 1515
13.47.1.11 struct usbdevice . . . 1516
13.47.2 Macro Definition Documentation . . . 1518

13.47.2.1 CLEAR_KEYBIT . . . 1518
13.47.2.2 DPI_COUNT . . . 1518
13.47.2.3 FEAT_ADJRATE . . . 1518
13.47.2.4 FEAT_ANSI . . . 1518
13.47.2.5 FEAT_BIND . . . 1518
13.47.2.6 FEAT_COMMON . . . 1518
13.47.2.7 FEAT_FWUPDATE . . . 1518
13.47.2.8 FEAT_FWVERSION . . . 1519
13.47.2.9 FEAT_HWLOAD . . . 1519
13.47.2.10 FEAT_ISO . . . 1519
13.47.2.11 FEAT_LMASK . . . 1519
13.47.2.12 FEAT_MONOCHROME . . . 1519
13.47.2.13 FEAT_MOUSEACCEL . . . 1519
13.47.2.14 FEAT_NOTIFY . . . 1519
13.47.2.15 FEAT_POLLRATE . . . 1519
13.47.2.16 FEAT_RGB . . . 1519
13.47.2.17 FEAT_STD_NRGB . . . 1520
13.47.2.18 FEAT_STD_RGB . . . 1520
13.47.2.19 HAS_ANY_FEATURE . . . 1520
13.47.2.20 HAS_FEATURES . . . 1520
13.47.2.21 HWMODE_K70 . . . 1520
13.47.2.22 HWMODE_K95 . . . 1520
13.47.2.23 HWMODE_MAX . . . 1520
13.47.2.24 I_CAPS . . . 1520
13.47.2.25 I_NUM . . . 1520
13.47.2.26 I_SCROLL . . . 1520
13.47.2.27 IFACE_MAX . . . 1521
13.47.2.28 KB_NAME_LEN . . . 1521
13.47.2.29 LIFT_MAX . . . 1521
13.47.2.30 LIFT_MIN . . . 1521
13.47.2.31 MACRO_MAX . . . 1521
13.47.2.32 MD_NAME_LEN . . . 1521
13.47.2.33 MODE_COUNT . . . 1521
13.47.2.34 MSG_SIZE . . . 1521
13.47.2.35 NEEDS_FW_UPDATE . . . 1521
13.47.2.36 OUTFIFO_MAX . . . 1522
13.47.2.37 PR_NAME_LEN . . . 1522
13.47.2.38 SCROLL_ACCELERATED . . . 1522
13.47.2.39 SCROLL_MAX . . . 1522
13.47.2.40 SCROLL_MIN . . . 1522

13.47.2.41 SERIAL_LEN . . . 1522
13.47.2.42 SET_KEYBIT . . . 1522
13.47.3 Variable Documentation . . . 1522
13.47.3.1 vtable_keyboard . . . 1522
13.47.3.2 vtable_keyboard_nonrgb . . . 1522
13.47.3.3 vtable_mouse . . . 1523
13.48 src/ckb-daemon/usb.c File Reference . . . 1523
13.48.1 Function Documentation . . . 1524
13.48.1.1 _resetusb . . . 1524
13.48.1.2 _setupusb . . . 1524
13.48.1.3 _usbrecv . . . 1529
13.48.1.4 _usbsend . . . 1530
13.48.1.5 closeusb . . . 1531
13.48.1.6 devmain . . . 1533
13.48.1.7 get_vtable . . . 1535
13.48.1.8 product_str . . . 1536
13.48.1.9 revertusb . . . 1537
13.48.1.10 setupusb . . . 1538
13.48.1.11 usb_tryreset . . . 1539
13.48.1.12 vendor_str . . . 1540
13.48.2 Variable Documentation . . . 1541
13.48.2.1 features_mask . . . 1541
13.48.2.2 hwload_mode . . . 1541
13.48.2.3 reset_stop . . . 1541
13.48.2.4 usbmutex . . . 1541
13.49 src/ckb-daemon/usb.h File Reference . . . 1541
13.49.1 Detailed Description . . . 1545
13.49.2 Macro Definition Documentation . . . 1546
13.49.2.1 DELAY_LONG . . . 1546
13.49.2.2 DELAY_MEDIUM . . . 1546
13.49.2.3 DELAY_SHORT . . . 1546
13.49.2.4 IS_FULLRANGE . . . 1546
13.49.2.5 IS_GLAIVE . . . 1546
13.49.2.6 IS_HARPOON . . . 1546
13.49.2.7 IS_K63 . . . 1547
13.49.2.8 IS_K65 . . . 1547
13.49.2.9 IS_K68 . . . 1547
13.49.2.10 IS_K70 . . . 1547
13.49.2.11 IS_K95 . . . 1547
13.49.2.12 IS_M65 . . . 1547

13.49.2.13 IS_MONOCHROME . . . 1547
13.49.2.14 IS_MONOCHROME_DEV . . . 1547
13.49.2.15 IS_MOUSE . . . 1547
13.49.2.16 IS_MOUSE_DEV . . . 1548
13.49.2.17 IS_PLATINUM . . . 1548
13.49.2.18 IS_RGB . . . 1548
13.49.2.19 IS_RGB_DEV . . . 1548
13.49.2.20 IS_SABRE . . . 1548
13.49.2.21 IS_SCIMITAR . . . 1548
13.49.2.22 IS_STRAFE . . . 1548
13.49.2.23 IS_V2_OVERRIDE . . . 1548
13.49.2.24 NK95_HWOFF . . . 1548
13.49.2.25 NK95_HWON . . . 1549
13.49.2.26 NK95_M1 . . . 1549
13.49.2.27 NK95_M2 . . . 1549
13.49.2.28 NK95_M3 . . . 1549
13.49.2.29 nk95cmd . . . 1549
13.49.2.30 P_GLAIVE . . . 1549
13.49.2.31 P_GLAIVE_STR . . . 1549
13.49.2.32 P_HARPOON . . . 1549
13.49.2.33 P_HARPOON_STR . . . 1550
13.49.2.34 P_K63_NRGB . . . 1550
13.49.2.35 P_K63_NRGB_STR . . . 1550
13.49.2.36 P_K65 . . . 1550
13.49.2.37 P_K65_LUX . . . 1550
13.49.2.38 P_K65_LUX_STR . . . 1550
13.49.2.39 P_K65_NRGB . . . 1550
13.49.2.40 P_K65_NRGB_STR . . . 1550
13.49.2.41 P_K65_RFIRE . . . 1550
13.49.2.42 P_K65_RFIRE_STR . . . 1550
13.49.2.43 P_K65_STR . . . 1550
13.49.2.44 P_K68 . . . 1551
13.49.2.45 P_K68_STR . . . 1551
13.49.2.46 P_K70 . . . 1551
13.49.2.47 P_K70_LUX . . . 1551
13.49.2.48 P_K70_LUX_NRGB . . . 1551
13.49.2.49 P_K70_LUX_NRGB_STR . . . 1551
13.49.2.50 P_K70_LUX_STR . . . 1551
13.49.2.51 P_K70_NRGB . . . 1551
13.49.2.52 P_K70_NRGB_STR . . . 1551

13.49.2.53 P_K70_RFIRE . . . 1551
13.49.2.54 P_K70_RFIRE_NRGB . . . 1552
13.49.2.55 P_K70_RFIRE_NRGB_STR . . . 1552
13.49.2.56 P_K70_RFIRE_STR . . . 1552
13.49.2.57 P_K70_STR . . . 1552
13.49.2.58 P_K95 . . . 1552
13.49.2.59 P_K95_NRGB . . . 1552
13.49.2.60 P_K95_NRGB_STR . . . 1552
13.49.2.61 P_K95_PLATINUM . . . 1552
13.49.2.62 P_K95_PLATINUM_STR . . . 1552
13.49.2.63 P_K95_STR . . . 1552
13.49.2.64 P_M65 . . . 1552
13.49.2.65 P_M65_PRO . . . 1553
13.49.2.66 P_M65_PRO_STR . . . 1553
13.49.2.67 P_M65_STR . . . 1553
13.49.2.68 P_SABRE_L . . . 1553
13.49.2.69 P_SABRE_L_STR . . . 1553
13.49.2.70 P_SABRE_N . . . 1553
13.49.2.71 P_SABRE_N_STR . . . 1553
13.49.2.72 P_SABRE_O . . . 1553
13.49.2.73 P_SABRE_O2 . . . 1553
13.49.2.74 P_SABRE_O2_STR . . . 1553
13.49.2.75 P_SABRE_O_STR . . . 1553
13.49.2.76 P_SCIMITAR . . . 1554
13.49.2.77 P_SCIMITAR_PRO . . . 1554
13.49.2.78 P_SCIMITAR_PRO_STR . . . 1554
13.49.2.79 P_SCIMITAR_STR . . . 1554
13.49.2.80 P_STRAFE . . . 1554
13.49.2.81 P_STRAFE_NRGB . . . 1554
13.49.2.82 P_STRAFE_NRGB_2 . . . 1554
13.49.2.83 P_STRAFE_NRGB_2_STR . . . 1554
13.49.2.84 P_STRAFE_NRGB_STR . . . 1554
13.49.2.85 P_STRAFE_STR . . . 1554
13.49.2.86 resetusb . . . 1555
13.49.2.87 USB_DELAY_DEFAULT . . . 1555
13.49.2.88 usbrecv . . . 1555
13.49.2.89 usbsend . . . 1555
13.49.2.90 V_CORSAIR . . . 1555
13.49.2.91 V_CORSAIR_STR . . . 1555
13.49.3 Function Documentation . . . 1556

13.49.3.1 _nk95cmd . . . 1556
13.49.3.2 _resetusb . . . 1557
13.49.3.3 _usbrecv . . . 1558
13.49.3.4 _usbsend . . . 1559
13.49.3.5 closeusb . . . 1561
13.49.3.6 os_closeusb . . . 1563
13.49.3.7 os_inputmain . . . 1564
13.49.3.8 os_resetusb . . . 1567
13.49.3.9 os_sendindicators . . . 1568
13.49.3.10 os_setupusb . . . 1570
13.49.3.11 os_usbrecv . . . 1572
13.49.3.12 os_usbsend . . . 1574
13.49.3.13 product_str . . . 1576
13.49.3.14 revertusb . . . 1577
13.49.3.15 setupusb . . . 1578
13.49.3.16 usb_tryreset . . . 1579
13.49.3.17 usbkill . . . 1580
13.49.3.18 usbmain . . . 1581
13.49.3.19 vendor_str . . . 1582
13.50 src/ckb-daemon/usb_linux.c File Reference . . . 1583
13.50.1 Data Structure Documentation . . . 1585
13.50.1.1 struct _model . . . 1585
13.50.2 Macro Definition Documentation . . . 1585
13.50.2.1 DEBUG . . . 1585
13.50.2.2 N_MODELS . . . 1585
13.50.2.3 TEST_RESET . . . 1585
13.50.3 Function Documentation . . . 1586
13.50.3.1 _nk95cmd . . . 1586
13.50.3.2 os_closeusb . . . 1587
13.50.3.3 os_inputmain . . . 1588
13.50.3.4 os_resetusb . . . 1591
13.50.3.5 os_sendindicators . . . 1592
13.50.3.6 os_setupusb . . . 1593
13.50.3.7 os_usbrecv . . . 1595
13.50.3.8 os_usbsend . . . 1596
13.50.3.9 strtrim . . . 1598
13.50.3.10 udev_enum . . . 1599
13.50.3.11 usb_add_device . . . 1600
13.50.3.12 usb_rm_device . . . 1601
13.50.3.13 usbadd . . . 1602

13.50.3.14 usbclaim . . . 1604
13.50.3.15 usbkill . . . 1605
13.50.3.16 usbmain . . . 1605
13.50.3.17 usbunclaim . . . 1607
13.50.4 Variable Documentation . . . 1608
13.50.4.1 kbsyspath . . . 1608
13.50.4.2 models . . . 1608
13.50.4.3 udev . . . 1608
13.50.4.4 udevthread . . . 1609
13.50.4.5 usbthread . . . 1609
13.51 src/ckb-daemon/usb_mac.c File Reference . . . 1609
13.52 src/ckb-mviz/_kiss_fft_guts.h File Reference . . . 1609
13.52.1 Data Structure Documentation . . . 1611
13.52.1.1 struct kiss_fft_state . . . 1611
13.52.2 Macro Definition Documentation . . . 1612
13.52.2.1 C_ADD . . . 1612
13.52.2.2 C_ADDTO . . . 1612
13.52.2.3 C_FIXDIV . . . 1612
13.52.2.4 C_MUL . . . 1612
13.52.2.5 C_MULBYSCALAR . . . 1612
13.52.2.6 C_SUB . . . 1613
13.52.2.7 C_SUBFROM . . . 1613
13.52.2.8 CHECK_OVERFLOW_OP . . . 1613
13.52.2.9 HALF_OF . . . 1613
13.52.2.10 kf_cexp . . . 1613
13.52.2.11 KISS_FFT_COS . . . 1613
13.52.2.12 KISS_FFT_SIN . . . 1613
13.52.2.13 KISS_FFT_TMP_ALLOC . . . 1614
13.52.2.14 KISS_FFT_TMP_FREE . . . 1614
13.52.2.15 MAXFACTORS . . . 1614
13.52.2.16 pcpx . . . 1614
13.52.2.17 S_MUL . . . 1614
13.53 src/ckb-mviz/kiss_fft.c File Reference . . . 1614
13.53.1 Function Documentation . . . 1615
13.53.1.1 kf_bfly2 . . . 1615
13.53.1.2 kf_bfly3 . . . 1615
13.53.1.3 kf_bfly4 . . . 1616
13.53.1.4 kf_bfly5 . . . 1617
13.53.1.5 kf_bfly_generic . . . 1618
13.53.1.6 kf_factor . . . 1619

13.53.1.7 kf_work . . . 1620
13.53.1.8 kiss_fft . . . 1621
13.53.1.9 kiss_fft_alloc . . . 1622
13.53.1.10 kiss_fft_cleanup . . . 1623
13.53.1.11 kiss_fft_next_fast_size . . . 1624
13.53.1.12 kiss_fft_stride . . . 1624
13.54 src/ckb-mviz/kiss_fft.h File Reference . . . 1625
13.54.1 Data Structure Documentation . . . 1627
13.54.1.1 struct kiss_fft_cpx . . . 1627
13.54.2 Macro Definition Documentation . . . 1627
13.54.2.1 KISS_FFT_FREE . . . 1627
13.54.2.2 kiss_fft_free . . . 1627
13.54.2.3 KISS_FFT_MALLOC . . . 1627
13.54.2.4 kiss_fft_scalar . . . 1628
13.54.2.5 kiss_fftr_next_fast_size_real . . . 1628
13.54.3 Typedef Documentation . . . 1628
13.54.3.1 kiss_fft_cfg . . . 1628
13.54.4 Function Documentation . . . 1628
13.54.4.1 kiss_fft . . . 1628
13.54.4.2 kiss_fft_alloc . . . 1629
13.54.4.3 kiss_fft_cleanup . . . 1630
13.54.4.4 kiss_fft_next_fast_size . . . 1630
13.54.4.5 kiss_fft_stride . . . 1631
13.55 src/ckb-mviz/kiss_fftr.c File Reference . . . 1632
13.55.1 Data Structure Documentation . . . 1633
13.55.1.1 struct kiss_fftr_state . . . 1633
13.55.2 Function Documentation . . . 1634
13.55.2.1 kiss_fftr . . . 1634
13.55.2.2 kiss_fftr_alloc . . . 1635
13.55.2.3 kiss_fftri . . . 1636
13.56 src/ckb-mviz/kiss_fftr.h File Reference . . . 1637
13.56.1 Macro Definition Documentation . . . 1638
13.56.1.1 kiss_fftr_free . . . 1638
13.56.2 Typedef Documentation . . . 1638
13.56.2.1 kiss_fftr_cfg . . . 1638
13.56.3 Function Documentation . . . 1638
13.56.3.1 kiss_fftr . . . 1638
13.56.3.2 kiss_fftr_alloc . . . 1640
13.56.3.3 kiss_fftri . . . 1640
13.57 src/ckb/animadddialog.cpp File Reference . . . 1641

13.58 src/ckb/animadddialog.h File Reference . . . 1642
13.59 src/ckb/animscript.cpp File Reference . . . 1642
13.59.1 Function Documentation . . . 1643
13.59.1.1 urlParam . . . 1643
13.59.2 Variable Documentation . . . 1643
13.59.2.1 ONE_DAY . . . 1643
13.60 src/ckb/animscript.h File Reference . . . 1644
13.60.1 Data Structure Documentation . . . 1644
13.60.1.1 struct AnimScript::Param . . . 1644
13.61 src/ckb/animsettingdialog.cpp File Reference . . . 1645
13.61.1 Function Documentation . . . 1646
13.61.1.1 angleFlip . . . 1646
13.62 src/ckb/animsettingdialog.h File Reference . . . 1647
13.63 src/ckb/autorun.cpp File Reference . . . 1648
13.63.1 Variable Documentation . . . 1648
13.63.1.1 settingPath . . . 1648
13.64 src/ckb/autorun.h File Reference . . . 1649
13.65 src/ckb/ckb-anim.h File Reference . . . 1649
13.65.1 Data Structure Documentation . . . 1652
13.65.1.1 struct ckb_key . . . 1652
13.65.1.2 struct ckb_runctx . . . 1653
13.65.1.3 struct ckb_gradient . . . 1654
13.65.2 Macro Definition Documentation . . . 1654
13.65.2.1 CKB_CONTAINER . . . 1654
13.65.2.2 CKB_COPYRIGHT . . . 1654
13.65.2.3 CKB_DESCRIPTION . . . 1655
13.65.2.4 CKB_DOWN . . . 1655
13.65.2.5 CKB_GRAD_MAX . . . 1655
13.65.2.6 CKB_GUID . . . 1655
13.65.2.7 CKB_KEYCLEAR . . . 1655
13.65.2.8 CKB_KEYNAME_MAX . . . 1655
13.65.2.9 CKB_KP_NAME . . . 1655
13.65.2.10 CKB_KP_NONE . . . 1655
13.65.2.11 CKB_KP_POSITION . . . 1655
13.65.2.12 CKB_KPMODE . . . 1656
13.65.2.13 CKB_LICENSE . . . 1656
13.65.2.14 CKB_LIVEPARAMS . . . 1656
13.65.2.15 CKB_MAX_WORD . . . 1656
13.65.2.16 CKB_NAME . . . 1656
13.65.2.17 CKB_PARAM . . . 1656

13.65.2.18 CKB_PARAM_AGRADIENT . . . 1656
13.65.2.19 CKB_PARAM_ANGLE . . . 1656
13.65.2.20 CKB_PARAM_ARGB . . . 1656
13.65.2.21 CKB_PARAM_BOOL . . . 1657
13.65.2.22 CKB_PARAM_DOUBLE . . . 1657
13.65.2.23 CKB_PARAM_GRADIENT . . . 1657
13.65.2.24 CKB_PARAM_LABEL . . . 1657
13.65.2.25 CKB_PARAM_LONG . . . 1657
13.65.2.26 CKB_PARAM_RGB . . . 1657
13.65.2.27 CKB_PARAM_STRING . . . 1657
13.65.2.28 CKB_PARSE_AGRADIENT . . . 1657
13.65.2.29 CKB_PARSE_ANGLE . . . 1657
13.65.2.30 CKB_PARSE_ARGB . . . 1658
13.65.2.31 CKB_PARSE_BOOL . . . 1658
13.65.2.32 CKB_PARSE_DOUBLE . . . 1658
13.65.2.33 CKB_PARSE_GRADIENT . . . 1658
13.65.2.34 CKB_PARSE_LONG . . . 1658
13.65.2.35 CKB_PARSE_RGB . . . 1658
13.65.2.36 CKB_PARSE_STRING . . . 1658
13.65.2.37 CKB_PREEMPT . . . 1658
13.65.2.38 CKB_PRESET_END . . . 1658
13.65.2.39 CKB_PRESET_PARAM . . . 1659
13.65.2.40 CKB_PRESET_START . . . 1659
13.65.2.41 CKB_REAL_ANGLE . . . 1659
13.65.2.42 CKB_REPEAT . . . 1659
13.65.2.43 CKB_TIME_ABSOLUTE . . . 1659
13.65.2.44 CKB_TIME_DURATION . . . 1659
13.65.2.45 CKB_TIMEMODE . . . 1659
13.65.2.46 CKB_UP . . . 1659
13.65.2.47 CKB_VERSION . . . 1659
13.65.2.48 FALSE . . . 1660
13.65.2.49 M_PI . . . 1660
13.65.2.50 TRUE . . . 1660
13.65.3 Function Documentation . . . 1660
13.65.3.1 ckb_alpha_blend . . . 1660
13.65.3.2 ckb_frame . . . 1660
13.65.3.3 ckb_getline . . . 1661
13.65.3.4 ckb_grad_color . . . 1662
13.65.3.5 ckb_info . . . 1663
13.65.3.6 ckb_init . . . 1664

13.65.3.7 ckb_keypress . . . 1664
13.65.3.8 ckb_parameter . . . 1666
13.65.3.9 ckb_read_params . . . 1666
13.65.3.10 ckb_scan_grad . . . 1667
13.65.3.11 ckb_start . . . 1668
13.65.3.12 ckb_time . . . 1669
13.65.3.13 main . . . 1669
13.65.3.14 printurl . . . 1672
13.65.3.15 urldecode . . . 1672
13.66 src/ckb/ckbsettings.cpp File Reference . . . 1673
13.66.1 Macro Definition Documentation . . . 1674
13.66.1.1 lockMutex . . . 1674
13.66.1.2 lockMutexCache . . . 1674
13.66.1.3 lockMutexStatic . . . 1674
13.66.1.4 lockMutexStatic2 . . . 1674
13.66.2 Function Documentation . . . 1674
13.66.2.1 cacheWritesInProgress . . . 1674
13.66.2.2 globalSettings . . . 1674
13.66.2.3 settingsCacheMutex . . . 1675
13.66.2.4 settingsMutex . . . 1675
13.66.3 Variable Documentation . . . 1675
13.66.3.1 _globalSettings . . . 1675
13.66.3.2 globalCache . . . 1675
13.66.3.3 globalThread . . . 1675
13.67 src/ckb/ckbsettings.h File Reference . . . 1676
13.68 src/ckb/ckbsettingswriter.cpp File Reference . . . 1676
13.68.1 Macro Definition Documentation . . . 1677
13.68.1.1 lockMutex . . . 1677
13.68.1.2 lockMutexCache . . . 1677
13.68.1.3 lockMutexStatic . . . 1677
13.68.1.4 lockMutexStatic2 . . . 1677
13.68.2 Variable Documentation . . . 1678
13.68.2.1 cacheWritesInProgress . . . 1678
13.68.2.2 settingsCacheMutex . . . 1678
13.68.2.3 settingsMutex . . . 1678
13.69 src/ckb/ckbsettingswriter.h File Reference . . . 1678
13.70 src/ckb/colorbutton.cpp File Reference . . . 1678
13.71 src/ckb/colorbutton.h File Reference . . . 1679
13.72 src/ckb/colormap.cpp File Reference . . . 1680
13.72.1 Function Documentation . . . 1680

13.72.1.1 qs_strcmp . . . 1680
13.73 src/ckb/colormap.h File Reference . . . 1681
13.73.1 Typedef Documentation . . . 1682
13.73.1.1 QColorMap . . . 1682
13.73.1.2 QColorMapIterator . . . 1682
13.73.1.3 QMutableColorMapIterator . . . 1682
13.74 src/ckb/extrasettingswidget.cpp File Reference . . . 1682
13.74.1 Variable Documentation . . . 1682
13.74.1.1 lastSharedDimming . . . 1682
13.75 src/ckb/extrasettingswidget.h File Reference . . . 1683
13.76 src/ckb/fwupgradedialog.cpp File Reference . . . 1683
13.76.1 Data Structure Documentation . . . 1684
13.76.1.1 struct KbId . . . 1684
13.76.2 Function Documentation . . . 1685
13.76.2.1 DIALOG_HEIGHT_MAX . . . 1685
13.76.2.2 verifyFw . . . 1685
13.76.3 Variable Documentation . . . 1686
13.76.3.1 DIALOG_HEIGHT_MIN . . . 1686
13.76.3.2 DIALOG_WIDTH . . . 1686
13.76.3.3 ids . . . 1686
13.77 src/ckb/fwupgradedialog.h File Reference . . . 1686
13.78 src/ckb/gradientbutton.cpp File Reference . . . 1687
13.79 src/ckb/gradientbutton.h File Reference . . . 1688
13.80 src/ckb/gradientdialog.cpp File Reference . . . 1689
13.80.1 Variable Documentation . . . 1689
13.80.1.1 prefsPath . . . 1689
13.81 src/ckb/gradientdialog.h File Reference . . . 1689
13.82 src/ckb/gradientdialogwidget.cpp File Reference . . . 1690
13.83 src/ckb/gradientdialogwidget.h File Reference . . . 1690
13.84 src/ckb/kb.cpp File Reference . . . 1691
13.84.1 Variable Documentation . . . 1692
13.84.1.1 activeDevices . . . 1692
13.84.1.2 notifyPathMutex . . . 1692
13.84.1.3 notifyPaths . . . 1692
13.85 src/ckb/kb.h File Reference . . . 1692
13.86 src/ckb/kbanim.cpp File Reference . . . 1693
13.86.1 Typedef Documentation . . . 1694
13.86.1.1 blendFunc . . . 1694
13.86.2 Function Documentation . . . 1694
13.86.2.1 blendAdd . . . 1694

13.86.2.2 blendDivide . . . 1694
13.86.2.3 blendMultiply . . . 1694
13.86.2.4 blendNormal . . . 1695
13.86.2.5 blendSubtract . . . 1695
13.86.3 Variable Documentation . . . 1695
13.86.3.1 functions . . . 1695
13.87 src/ckb/kbanim.h File Reference . . . 1695
13.88 src/ckb/kbanimwidget.cpp File Reference . . . 1696
13.89 src/ckb/kbanimwidget.h File Reference . . . 1696
13.90 src/ckb/kbbind.cpp File Reference . . . 1697
13.91 src/ckb/kbbind.h File Reference . . . 1698
13.92 src/ckb/kbbindwidget.cpp File Reference . . . 1699
13.93 src/ckb/kbbindwidget.h File Reference . . . 1699
13.94 src/ckb/kbfirmware.cpp File Reference . . . 1700
13.94.1 Function Documentation . . . 1701
13.94.1.1 tableName . . . 1701
13.94.2 Variable Documentation . . . 1701
13.94.2.1 AUTO_CHECK_TIME . . . 1701
13.95 src/ckb/kbfirmware.h File Reference . . . 1701
13.96 src/ckb/kblight.cpp File Reference . . . 1702
13.96.1 Function Documentation . . . 1703
13.96.1.1 lToS . . . 1703
13.96.1.2 monoRgb . . . 1704
13.96.1.3 sToL . . . 1704
13.96.2 Variable Documentation . . . 1705
13.96.2.1 _shareDimming . . . 1705
13.96.2.2 activeLights . . . 1705
13.97 src/ckb/kblight.h File Reference . . . 1705
13.98 src/ckb/kblightwidget.cpp File Reference . . . 1706
13.99 src/ckb/kblightwidget.h File Reference . . . 1706
13.100 src/ckb/kbmanager.cpp File Reference . . . 1707
13.100.1 Variable Documentation . . . 1708
13.100.1.1 devpath . . . 1708
13.101 src/ckb/kbmanager.h File Reference . . . 1708
13.101.1 Macro Definition Documentation . . . 1709
13.101.1.1 DAEMON_UNAVAILABLE_STR . . . 1709
13.102 src/ckb/kbmode.cpp File Reference . . . 1709
13.103 src/ckb/kbmode.h File Reference . . . 1709
13.104 src/ckb/kbperf.cpp File Reference . . . 1710
13.104.1 Macro Definition Documentation . . . 1710

13.104.1.1 map_last . . . 1710
13.105 src/ckb/kbperf.h File Reference . . . 1710
13.106 src/ckb/kbprofile.cpp File Reference . . . 1711
13.107 src/ckb/kbprofile.h File Reference . . . 1712
13.108 src/ckb/kbprofiledialog.cpp File Reference . . . 1712
13.109 src/ckb/kbprofiledialog.h File Reference . . . 1713
13.110 src/ckb/kbwidget.cpp File Reference . . . 1714
13.111 src/ckb/kbwidget.h File Reference . . . 1714
13.112 src/ckb/keyaction.cpp File Reference . . . 1715
13.112.1 Variable Documentation . . . 1716
13.112.1.1 ANIM_KRSTOP . . . 1716
13.112.1.2 ANIM_ONCE . . . 1716
13.113 src/ckb/keyaction.h File Reference . . . 1716
13.114 src/ckb/keymap.cpp File Reference . . . 1717
13.114.1 Data Structure Documentation . . . 1719
13.114.1.1 struct KeyPatch . . . 1719
13.114.2 Macro Definition Documentation . . . 1719
13.114.2.1 ANSI_ENTER_H . . . 1719
13.114.2.2 ANSI_ENTER_W . . . 1720
13.114.2.3 ANSI_ENTER_X . . . 1720
13.114.2.4 ANSI_ENTER_Y . . . 1720
13.114.2.5 ANSI_LSHIFT_W . . . 1720
13.114.2.6 ANSI_LSHIFT_X . . . 1720
13.114.2.7 GLAIVE_HEIGHT . . . 1720
13.114.2.8 GLAIVE_WIDTH . . . 1720
13.114.2.9 HARPOON_HEIGHT . . . 1720
13.114.2.10 HARPOON_WIDTH . . . 1720
13.114.2.11 K63_HEIGHT . . . 1720
13.114.2.12 K63_TOP_COUNT . . . 1721
13.114.2.13 K63_WIDTH . . . 1721
13.114.2.14 K65_HEIGHT . . . 1721
13.114.2.15 K65_TOP_COUNT . . . 1721
13.114.2.16 K65_WIDTH . . . 1721
13.114.2.17 K68_HEIGHT . . . 1721
13.114.2.18 K68_TOP_COUNT . . . 1721
13.114.2.19 K68_WIDTH . . . 1721
13.114.2.20 K70_HEIGHT . . . 1721
13.114.2.21 K70_WIDTH . . . 1721
13.114.2.22 K70_X_START . . . 1722
13.114.2.23 K95_HEIGHT . . . 1722

13.114.2.24 K95_WIDTH . . . 1722
13.114.2.25 K95P_HEIGHT . . . 1722
13.114.2.26 K95P_WIDTH . . . 1722
13.114.2.27 K95P_X_START . . . 1722
13.114.2.28 KEYCOUNT_GLAIVE . . . 1722
13.114.2.29 KEYCOUNT_HARPOON . . . 1722
13.114.2.30 KEYCOUNT_K95 . . . 1722
13.114.2.31 KEYCOUNT_M65 . . . 1723
13.114.2.32 KEYCOUNT_SABRE . . . 1723
13.114.2.33 KEYCOUNT_SCIM . . . 1723
13.114.2.34 KSTRAFE_HEIGHT . . . 1723
13.114.2.35 KSTRAFE_WIDTH . . . 1723
13.114.2.36 KSTRAFE_X_START . . . 1723
13.114.2.37 LBARCOUNT_K95P . . . 1723
13.114.2.38 LBS . . . 1723
13.114.2.39 M65_HEIGHT . . . 1723
13.114.2.40 M65_WIDTH . . . 1723
13.114.2.41 N_LAYOUTS . . . 1724
13.114.2.42 N_MODELS . . . 1724
13.114.2.43 NS . . . 1724
13.114.2.44 patch . . . 1724
13.114.2.45 PATCH_COUNT . . . 1724
13.114.2.46 SABRE_HEIGHT . . . 1724
13.114.2.47 SABRE_WIDTH . . . 1724
13.114.2.48 SCIM_HEIGHT . . . 1724
13.114.2.49 SCIM_WIDTH . . . 1724
13.114.3 Function Documentation . . . 1724
13.114.3.1 _patch . . . 1724
13.114.3.2 getMap . . . 1725
13.114.3.3 patchABNT2 . . . 1729
13.114.3.4 patchANSI . . . 1730
13.114.3.5 patchISO . . . 1731
13.114.3.6 patchJP106 . . . 1731
13.114.3.7 patchJP106fn . . . 1732
13.114.3.8 patchnonJP106 . . . 1732
13.114.4 Variable Documentation . . . 1733
13.114.4.1 GlaiveKeys . . . 1733
13.114.4.2 HarpoonKeys . . . 1733
13.114.4.3 K63TopRow . . . 1733
13.114.4.4 K65TopRow . . . 1734

13.114.4.5 K68TopRow . . . 1734
13.114.4.6 K95BaseMap . . . 1734
13.114.4.7 K95Keys . . . 1734
13.114.4.8 K95PLbar . . . 1735
13.114.4.9 KStrafeKeys . . . 1735
13.114.4.10 M65Keys . . . 1736
13.114.4.11 patchDE . . . 1736
13.114.4.12 patchDK . . . 1736
13.114.4.13 patchDvorak . . . 1736
13.114.4.14 patchES . . . 1737
13.114.4.15 patchEU . . . 1737
13.114.4.16 patchFR . . . 1737
13.114.4.17 patchIT . . . 1737
13.114.4.18 patchJP . . . 1737
13.114.4.19 patchMX . . . 1738
13.114.4.20 patchNO . . . 1738
13.114.4.21 patchPTBR . . . 1738
13.114.4.22 patchSE . . . 1738
13.114.4.23 SabreKeys . . . 1738
13.114.4.24 scimKeys . . . 1739
13.114.4.25 standardMaps . . . 1739
13.115 src/ckb/keymap.h File Reference . . . 1739
13.116 src/ckb-daemon/keymap.h File Reference . . . 1740
13.116.1 Data Structure Documentation . . . 1741
13.116.1.1 struct key . . . 1741
13.116.2 Macro Definition Documentation . . . 1742
13.116.2.1 BTN_WHEELDOWN . . . 1742
13.116.2.2 BTN_WHEELUP . . . 1742
13.116.2.3 KEY_BACKSLASH_ISO . . . 1742
13.116.2.4 KEY_CORSAIR . . . 1742
13.116.2.5 KEY_NONE . . . 1743
13.116.2.6 KEY_UNBOUND . . . 1743
13.116.2.7 LED_DPI . . . 1743
13.116.2.8 LED_MOUSE . . . 1743
13.116.2.9 MOUSE_BUTTON_FIRST . . . 1743
13.116.2.10 MOUSE_EXTRA_FIRST . . . 1743
13.116.2.11 N_BUTTONS_EXTENDED . . . 1743
13.116.2.12 N_BUTTONS_HW . . . 1743
13.116.2.13 N_KEY_ZONES . . . 1743
13.116.2.14 N_KEYBYTES_EXTENDED . . . 1743

13.116.2.15 N_KEYBYTES_HW . . . 1744
13.116.2.16 N_KEYBYTES_INPUT . . . 1744
13.116.2.17 N_KEYS_EXTENDED . . . 1744
13.116.2.18 N_KEYS_EXTRA . . . 1744
13.116.2.19 N_KEYS_HW . . . 1744
13.116.2.20 N_KEYS_INPUT . . . 1744
13.116.2.21 N_MOUSE_ZONES . . . 1744
13.116.2.22 N_MOUSE_ZONES_EXTENDED . . . 1744
13.116.2.23 SCAN_KBD . . . 1744
13.116.2.24 SCAN_MOUSE . . . 1744
13.116.2.25 SCAN_SILENT . . . 1745
13.116.3 Function Documentation . . . 1745
13.116.3.1 corsair_kbcopy . . . 1745
13.116.3.2 corsair_mousecopy . . . 1745
13.116.3.3 hid_kb_translate . . . 1746
13.116.3.4 hid_mouse_translate . . . 1747
13.116.4 Variable Documentation . . . 1748
13.116.4.1 keymap . . . 1748
13.117 src/ckb/keywidget.cpp File Reference . . . 1748
13.117.1 Function Documentation . . . 1749
13.117.1.1 monoRgb . . . 1749
13.117.2 Variable Documentation . . . 1750
13.117.2.1 glaiveOverlay . . . 1750
13.117.2.2 harpOverlay . . . 1750
13.117.2.3 KEY_SIZE . . . 1750
13.117.2.4 m65Overlay . . . 1750
13.117.2.5 sabOverlay . . . 1751
13.117.2.6 scimOverlay . . . 1751
13.118 src/ckb/keywidget.h File Reference . . . 1751
13.119 src/ckb/kperfwidget.cpp File Reference . . . 1752
13.120 src/ckb/kperfwidget.h File Reference . . . 1752
13.120.1 Data Structure Documentation . . . 1753
13.120.1.1 struct KPerfWidget::IndicatorUi . . . 1753
13.121 src/ckb/layoutdialog.cpp File Reference . . . 1754
13.122 src/ckb/layoutdialog.h File Reference . . . 1755
13.123 src/ckb/macroreader.cpp File Reference . . . 1756
13.124 src/ckb/macroreader.h File Reference . . . 1756
13.125 src/ckb/main.cpp File Reference . . . 1757
13.125.1 Macro Definition Documentation . . . 1758
13.125.1.1 SHMEM_SIZE . . . 1758

13.125.1.2 SHMEM_SIZE_V015 . . . 1758
13.125.2 Enumeration Type Documentation . . . 1758
13.125.2.1 CommandLineParseResults . . . 1758
13.125.3 Function Documentation . . . 1759
13.125.3.1 appShare . . . 1759
13.125.3.2 isRunning . . . 1759
13.125.3.3 main . . . 1760
13.125.3.4 parseCommandLine . . . 1761
13.125.3.5 pidActive . . . 1763
13.126 src/ckb/mainwindow.cpp File Reference . . . 1763
13.126.1 Variable Documentation . . . 1764
13.126.1.1 appShare . . . 1764
13.126.1.2 configLabel . . . 1764
13.126.1.3 daemonDialogText . . . 1764
13.126.1.4 devpath . . . 1764
13.127 src/ckb/mainwindow.h File Reference . . . 1764
13.128 src/ckb/media.h File Reference . . . 1766
13.128.1 Macro Definition Documentation . . . 1766
13.128.1.1 ENUM_C . . . 1766
13.128.1.2 ENUM_END_C . . . 1766
13.128.1.3 EXTERN_C . . . 1766
13.128.2 Enumeration Type Documentation . . . 1766
13.128.2.1 muteState . . . 1766
13.128.3 Function Documentation . . . 1767
13.128.3.1 getMuteState . . . 1767
13.129 src/ckb/media_linux.cpp File Reference . . . 1768
13.129.1 Function Documentation . . . 1768
13.129.1.1 getMuteState . . . 1768
13.129.2 Variable Documentation . . . 1769
13.129.2.1 lastKnown . . . 1769
13.130 src/ckb/media_mac.m File Reference . . . 1769
13.130.1 Function Documentation . . . 1769
13.130.1.1 disableAppNap . . . 1769
13.130.1.2 getMuteState . . . 1770
13.131 src/ckb/modeselectdialog.cpp File Reference . . . 1770
13.132 src/ckb/modeselectdialog.h File Reference . . . 1771
13.133 src/ckb/mperfwidget.cpp File Reference . . . 1772
13.133.1 Macro Definition Documentation . . . 1772
13.133.1.1 SET_END . . . 1772
13.133.1.2 SET_START . . . 1772

13.133.2 Function Documentation . . . 1772
13.133.2.1 dpiExp . . . 1772
13.133.2.2 dpiLog . . . 1773
13.133.2.3 dpiRound . . . 1774
13.133.3 Variable Documentation . . . 1774
13.133.3.1 DPI_BASE . . . 1774
13.133.3.2 xyLinkPath . . . 1774
13.134 src/ckb/mperfwidget.h File Reference . . . 1774
13.134.1 Data Structure Documentation . . . 1775
13.134.1.1 struct MPerfWidget::DpiUi . . . 1775
13.135 src/ckb/quazip/crypt.h File Reference . . . 1776
13.135.1 Macro Definition Documentation . . . 1777
13.135.1.1 CRC32 . . . 1778
13.135.1.2 zdecode . . . 1778
13.135.1.3 zencode . . . 1778
13.135.2 Function Documentation . . . 1778
13.135.2.1 decrypt_byte . . . 1778
13.135.2.2 init_keys . . . 1778
13.135.2.3 update_keys . . . 1779
13.136 src/ckb/quazip/ioapi.h File Reference . . . 1780
13.136.1 Data Structure Documentation . . . 1782
13.136.1.1 struct zlib_filefunc_def_s . . . 1782
13.136.1.2 struct zlib_filefunc64_def_s . . . 1782
13.136.1.3 struct zlib_filefunc64_32_def_s . . . 1783
13.136.2 Macro Definition Documentation . . . 1783
13.136.2.1 __USE_FILE_OFFSET64 . . . 1783
13.136.2.2 __USE_LARGEFILE64 . . . 1783
13.136.2.3 _FILE_OFFSET_BIT . . . 1784
13.136.2.4 _LARGEFILE64_SOURCE . . . 1784
13.136.2.5 OF . . . 1784
13.136.2.6 ZCALLBACK . . . 1784
13.136.2.7 ZCLOSE64 . . . 1784
13.136.2.8 ZERROR64 . . . 1784
13.136.2.9 ZFAKECLOSE64 . . . 1784
13.136.2.10 ZLIB_FILEFUNC_MODE_CREATE . . . 1784
13.136.2.11 ZLIB_FILEFUNC_MODE_EXISTING . . . 1784
13.136.2.12 ZLIB_FILEFUNC_MODE_READ . . . 1784
13.136.2.13 ZLIB_FILEFUNC_MODE_READWRITEFILTER . . . 1785
13.136.2.14 ZLIB_FILEFUNC_MODE_WRITE . . . 1785
13.136.2.15 ZLIB_FILEFUNC_SEEK_CUR . . . 1785

13.136.2.16 ZLIB_FILEFUNC_SEEK_END . . . 1785
13.136.2.17 ZLIB_FILEFUNC_SEEK_SET . . . 1785
13.136.2.18 ZOPEN64 . . . 1785
13.136.2.19 ZREAD64 . . . 1785
13.136.2.20 ZSEEK64 . . . 1785
13.136.2.21 ZTELL64 . . . 1786
13.136.2.22 ZWRITE64 . . . 1786
13.136.3 Typedef Documentation . . . 1786
13.136.3.1 _Z_OF . . . 1786
13.136.3.2 zlib_filefunc64_32_def . . . 1786
13.136.3.3 zlib_filefunc64_def . . . 1786
13.136.3.4 zlib_filefunc_def . . . 1786
13.136.3.5 ZPOS64_T . . . 1786
13.136.4 Function Documentation . . . 1786
13.136.4.1 _Z_OF . . . 1786
13.136.4.2 _Z_OF . . . 1786
13.136.4.3 _Z_OF . . . 1786
13.136.4.4 _Z_OF . . . 1786
13.136.4.5 _Z_OF . . . 1786
13.136.4.6 fill_zlib_filefunc64_32_def_from_filefunc32 . . . 1787
13.137 src/ckb/quazip/JlCompress.cpp File Reference . . . 1787
13.137.1 Function Documentation . . . 1788
13.137.1.1 copyData . . . 1788
13.138 src/ckb/quazip/JlCompress.h File Reference . . . 1789
13.139 src/ckb/quazip/qioapi.cpp File Reference . . . 1789
13.139.1 Macro Definition Documentation . . . 1791
13.139.1.1 SEEK_CUR . . . 1791
13.139.1.2 SEEK_END . . . 1791
13.139.1.3 SEEK_SET . . . 1791
13.139.2 Function Documentation . . . 1791
13.139.2.1 call_zopen64 . . . 1791
13.139.2.2 call_zseek64 . . . 1791
13.139.2.3 call_ztell64 . . . 1791
13.139.2.4 fill_qiodevice64_filefunc . . . 1792
13.139.2.5 fill_qiodevice_filefunc . . . 1793
13.139.2.6 fill_zlib_filefunc64_32_def_from_filefunc32 . . . 1794
13.139.2.7 qiodevice64_seek_file_func . . . 1795
13.139.2.8 qiodevice64_tell_file_func . . . 1796
13.139.2.9 qiodevice_close_file_func . . . 1796
13.139.2.10 qiodevice_error_file_func . . . 1797

13.139.2.1 qiodevice_fakeclose_file_func . . . 1797
13.139.2.2 qiodevice_open_file_func . . . 1798
13.139.2.3 qiodevice_read_file_func . . . 1799
13.139.2.4 qiodevice_seek_file_func . . . 1799
13.139.2.5 qiodevice_tell_file_func . . . 1800
13.139.2.6 qiodevice_write_file_func . . . 1801
13.140 src/ckb/quazip/quaadler32.cpp File Reference . . . 1801
13.141 src/ckb/quazip/quaadler32.h File Reference . . . 1802
13.142 src/ckb/quazip/quachecksum32.h File Reference . . . 1803
13.143 src/ckb/quazip/quacrc32.cpp File Reference . . . 1804
13.144 src/ckb/quazip/quacrc32.h File Reference . . . 1805
13.145 src/ckb/quazip/quagzipfile.cpp File Reference . . . 1806
13.146 src/ckb/quazip/quagzipfile.h File Reference . . . 1807
13.147 src/ckb/quazip/quaziodevice.cpp File Reference . . . 1808
13.147.1 Macro Definition Documentation . . . 1809
13.147.1.1 QUAZIO_INBUFSIZE . . . 1809
13.147.1.2 QUAZIO_OUTBUFSIZE . . . 1809
13.148 src/ckb/quazip/quaziodevice.h File Reference . . . 1809
13.149 src/ckb/quazip/quazip.cpp File Reference . . . 1810
13.149.1 Function Documentation . . . 1811
13.149.1.1 QuaZip_getFileInfo . . . 1811
13.149.1.2 QuaZip_getFileInfo . . . 1811
13.149.1.3 QuaZip_getFileInfo . . . 1812
13.149.1.4 QuaZip_getFileInfo . . . 1812
13.150 src/ckb/quazip/quazip.h File Reference . . . 1812
13.150.1 Macro Definition Documentation . . . 1813
13.150.1.1 UNZ_OPENERROR . . . 1813
13.151 src/ckb/quazip/quazip_global.h File Reference . . . 1813
13.151.1 Macro Definition Documentation . . . 1814
13.151.1.1 QUAZIP_EXPORT . . . 1814
13.151.1.2 QUAZIP_EXTRA_NTFS_MAGIC . . . 1814
13.151.1.3 QUAZIP_EXTRA_NTFS_TIME_MAGIC . . . 1814
13.151.1.4 UNUSED . . . 1815
13.152 src/ckb/quazip/quazipdir.cpp File Reference . . . 1815
13.152.1 Function Documentation . . . 1815
13.152.1.1 QuaZipDir_convertInfoList . . . 1815
13.152.1.2 QuaZipDir_convertInfoList . . . 1815
13.152.1.3 QuaZipDir_convertInfoList . . . 1816
13.152.1.4 QuaZipDir_getFileInfo . . . 1816
13.153 src/ckb/quazip/quazipdir.h File Reference . . . 1817

13.154 src/ckb/quazip/quazipfile.cpp File Reference . . . 1817
13.155 src/ckb/quazip/quazipfile.h File Reference . . . 1818
13.156 src/ckb/quazip/quazipfileinfo.cpp File Reference . . . 1819
13.156.1 Function Documentation . . . 1819
13.156.1.1 getNTFSTime . . . 1820
13.156.1.2 permissionsFromExternalAttr . . . 1821
13.157 src/ckb/quazip/quazipfileinfo.h File Reference . . . 1821
13.158 src/ckb/quazip/quazipnewinfo.cpp File Reference . . . 1822
13.158.1 Function Documentation . . . 1823
13.158.1.1 QuaZipNewInfo_init . . . 1823
13.158.1.2 QuaZipNewInfo_setPermissions . . . 1824
13.158.1.3 setNTFSTime . . . 1824
13.159 src/ckb/quazip/quazipnewinfo.h File Reference . . . 1826
13.160 src/ckb/quazip/unzip.c File Reference . . . 1827
13.160.1 Data Structure Documentation . . . 1830
13.160.1.1 struct unz_file_info64_internal_s . . . 1830
13.160.1.2 struct file_in_zip64_read_info_s . . . 1831
13.160.1.3 struct unz64_s . . . 1831
13.160.2 Macro Definition Documentation . . . 1832
13.160.2.1 ALLOC . . . 1832
13.160.2.2 BUFREADCOMMENT . . . 1833
13.160.2.3 CASESENSITIVITYDEFAULT_NO . . . 1833
13.160.2.4 CASESENSITIVITYDEFAULTVALUE . . . 1833
13.160.2.5 local . . . 1833
13.160.2.6 SIZECENTRALDIRITEM . . . 1833
13.160.2.7 SIZEZIPLOCALHEADER . . . 1833
13.160.2.8 STRCMPCASENOSENTIVEFUNCTION . . . 1833
13.160.2.9 TRYFREE . . . 1833
13.160.2.10 UNZ_BUFSIZE . . . 1833
13.160.2.11 UNZ_MAXFILENAMEINZIP . . . 1834
13.160.3 Typedef Documentation . . . 1834
13.160.3.1 unz_file_info64_internal . . . 1834
13.160.3.2 z_crc_t . . . 1834
13.160.4 Function Documentation . . . 1834
13.160.4.1 _Z_OF . . . 1834
13.160.4.2 _Z_OF . . . 1834
13.160.4.3 _Z_OF . . . 1834
13.160.4.4 _Z_OF . . . 1834
13.160.4.5 _Z_OF . . . 1834
13.160.4.6 strcmpcasenosensitive_internal . . . 1834

13.160.4.7 unz64local_CheckCurrentFileCoherencyHeader . . . 1834
13.160.4.8 unz64local_DosDateToTmuDate . . . 1836
13.160.4.9 unz64local_getByte . . . 1837
13.160.4.10 unz64local_GetCurrentFileInfoInternal . . . 1838
13.160.4.11 unz64local_getLong . . . 1841
13.160.4.12 unz64local_getLong64 . . . 1842
13.160.4.13 unz64local_getShort . . . 1844
13.160.4.14 unz64local_SearchCentralDir . . . 1845
13.160.4.15 unz64local_SearchCentralDir64 . . . 1846
13.160.4.16 unzClearFlags . . . 1847
13.160.4.17 unzClose . . . 1848
13.160.4.18 unzCloseCurrentFile . . . 1849
13.160.4.19 unzeof . . . 1850
13.160.4.20 unzGetCurrentFileInfo . . . 1850
13.160.4.21 unzGetCurrentFileInfo64 . . . 1851
13.160.4.22 unzGetCurrentFileZStreamPos64 . . . 1852
13.160.4.23 unzGetFilePos . . . 1852
13.160.4.24 unzGetFilePos64 . . . 1853
13.160.4.25 unzGetGlobalComment . . . 1853
13.160.4.26 unzGetGlobalInfo . . . 1854
13.160.4.27 unzGetGlobalInfo64 . . . 1854
13.160.4.28 unzGetLocalExtrafield . . . 1855
13.160.4.29 unzGetOffset . . . 1856
13.160.4.30 unzGetOffset64 . . . 1856
13.160.4.31 unzGoToFilePos . . . 1857
13.160.4.32 unzGoToFilePos64 . . . 1857
13.160.4.33 unzGoToFirstFile . . . 1858
13.160.4.34 unzGoToNextFile . . . 1859
13.160.4.35 unzLocateFile . . . 1860
13.160.4.36 unzOpen . . . 1861
13.160.4.37 unzOpen2 . . . 1861
13.160.4.38 unzOpen2_64 . . . 1862
13.160.4.39 unzOpen64 . . . 1863
13.160.4.40 unzOpenCurrentFile . . . 1864
13.160.4.41 unzOpenCurrentFile2 . . . 1864
13.160.4.42 unzOpenCurrentFile3 . . . 1864
13.160.4.43 unzOpenCurrentFilePassword . . . 1867
13.160.4.44 unzOpenInternal . . . 1868
13.160.4.45 unzReadCurrentFile . . . 1871
13.160.4.46 unzSetFlags . . . 1874

13.160.4.47 unzSetOffset . . . 1875
13.160.4.48 unzSetOffset64 . . . 1875
13.160.4.49 unzStringFileNameCompare . . . 1876
13.160.4.50 unztell . . . 1876
13.160.4.51 unztell64 . . . 1877
13.160.5 Variable Documentation . . . 1877
13.160.5.1 unz_copyright . . . 1877
13.161 src/ckb/quazip/unzip.h File Reference . . . 1878
13.161.1 Data Structure Documentation . . . 1880
13.161.1.1 struct tm_unz_s . . . 1880
13.161.1.2 struct unz_global_info64_s . . . 1880
13.161.1.3 struct unz_global_info_s . . . 1881
13.161.1.4 struct unz_file_info64_s . . . 1882
13.161.1.5 struct unz_file_info_s . . . 1883
13.161.1.6 struct unz_file_pos_s . . . 1885
13.161.1.7 struct unz64_file_pos_s . . . 1886
13.161.2 Macro Definition Documentation . . . 1886
13.161.2.1 UNZ_AUTO_CLOSE . . . 1886
13.161.2.2 UNZ_BADZIPFILE . . . 1886
13.161.2.3 UNZ_CRCERROR . . . 1886
13.161.2.4 UNZ_DEFAULT_FLAGS . . . 1887
13.161.2.5 UNZ_END_OF_LIST_OF_FILE . . . 1887
13.161.2.6 UNZ_EOF . . . 1887
13.161.2.7 UNZ_ERRNO . . . 1887
13.161.2.8 UNZ_INTERNALERROR . . . 1887
13.161.2.9 UNZ_OK . . . 1887
13.161.2.10 UNZ_PARAMERROR . . . 1887
13.161.2.11 Z_BZIP2ED . . . 1888
13.161.3 Typedef Documentation . . . 1888
13.161.3.1 tm_unz . . . 1888
13.161.3.2 unz64_file_pos . . . 1888
13.161.3.3 unz_file_info . . . 1888
13.161.3.4 unz_file_info64 . . . 1888
13.161.3.5 unz_file_pos . . . 1888
13.161.3.6 unz_global_info . . . 1888
13.161.3.7 unz_global_info64 . . . 1888
13.161.3.8 unzFile . . . 1888
13.161.4 Function Documentation . . . 1888
13.161.4.1 _Z_OF . . . 1888
13.161.4.2 Z_OF . . . 1888

13.161.4.3 _Z_OF . . . 1888
13.161.4.4 _Z_OF . . . 1888
13.161.4.5 _Z_OF . . . 1888
13.161.4.6 _Z_OF . . . 1888
13.161.4.7 _Z_OF . . . 1888
13.161.4.8 _Z_OF . . . 1888
13.161.4.9 _Z_OF . . . 1888
13.161.4.10 _Z_OF . . . 1888
13.161.4.11 _Z_OF . . . 1889
13.161.4.12 _Z_OF . . . 1889
13.161.4.13 _Z_OF . . . 1889
13.161.4.14 _Z_OF . . . 1889
13.161.4.15 _Z_OF . . . 1889
13.161.4.16 unzClearFlags . . . 1889
13.161.4.17 unzGetFilePos . . . 1889
13.161.4.18 unzGetFilePos64 . . . 1890
13.161.4.19 unzGetOffset . . . 1890
13.161.4.20 unzGetOffset64 . . . 1891
13.161.4.21 unzGoToFilePos . . . 1891
13.161.4.22 unzGoToFilePos64 . . . 1892
13.161.4.23 unzOpenInternal . . . 1893
13.161.4.24 unzSetFlags . . . 1896
13.161.4.25 unzSetOffset . . . 1897
13.161.4.26 unzSetOffset64 . . . 1897
13.162 src/ckb/quazip/zip.c File Reference . . . 1898
13.162.1 Data Structure Documentation . . . 1901
13.162.1.1 struct linkedlist_datablock_internal_s . . . 1901
13.162.1.2 struct linkedlist_data_s . . . 1902
13.162.1.3 struct curfile64_info . . . 1903
13.162.1.4 struct zip64_internal . . . 1904
13.162.2 Macro Definition Documentation . . . 1904
13.162.2.1 ALLOC . . . 1905
13.162.2.2 BUFREADCOMMENT . . . 1905
13.162.2.3 CENTRALHEADERMAGIC . . . 1905
13.162.2.4 CRC_LOCALHEADER_OFFSET . . . 1905
13.162.2.5 DESCRIPTORHEADERMAGIC . . . 1905
13.162.2.6 ENDHEADERMAGIC . . . 1905
13.162.2.7 FLAG_LOCALHEADER_OFFSET . . . 1905
13.162.2.8 INCLUDECRYPTINGCODE_IFCRYPTALLOWED . . . 1905
13.162.2.9 local . . . 1905

13.162.2.10 LOCALHEADERMAGIC . . . 1905
13.162.2.11 MAKEULONG64 . . . 1906
13.162.2.12 SEEK_CUR . . . 1906
13.162.2.13 SEEK_END . . . 1906
13.162.2.14 SEEK_SET . . . 1906
13.162.2.15 SIZECENTRALHEADER . . . 1906
13.162.2.16 SIZEDATA_INDATABLOCK . . . 1906
13.162.2.17 TRYFREE . . . 1906
13.162.2.18 VERSIONMADEBY . . . 1906
13.162.2.19 Z_BUFSIZE . . . 1906
13.162.2.20 Z_MAXFILENAMEINZIP . . . 1906
13.162.2.21 ZIP64ENDHEADERMAGIC . . . 1907
13.162.2.22 ZIP64ENDLOCHEADERMAGIC . . . 1907
13.162.3 Typedef Documentation . . . 1907
13.162.3.1 linkedlist_data . . . 1907
13.162.3.2 linkedlist_datablock_internal . . . 1907
13.162.3.3 z_crc_t . . . 1907
13.162.4 Function Documentation . . . 1907
13.162.4.1 _Z_OF . . . 1907
13.162.4.2 _Z_OF . . . 1907
13.162.4.3 _Z_OF . . . 1907
13.162.4.4 _Z_OF . . . 1907
13.162.4.5 _Z_OF . . . 1907
13.162.4.6 _Z_OF . . . 1907
13.162.4.7 add_data_in_datablock . . . 1907
13.162.4.8 allocate_new_datablock . . . 1909
13.162.4.9 free_datablock . . . 1909
13.162.4.10 free_linkedlist . . . 1910
13.162.4.11 init_linkedlist . . . 1910
13.162.4.12 LoadCentralDirectoryRecord . . . 1911
13.162.4.13 Write_EndOfCentralDirectoryRecord . . . 1914
13.162.4.14 Write_GlobalComment . . . 1916
13.162.4.15 Write_LocalFileHeader . . . 1916
13.162.4.16 Write_Zip64EndOfCentralDirectoryLocator . . . 1918
13.162.4.17 Write_Zip64EndOfCentralDirectoryRecord . . . 1919
13.162.4.18 zip64FlushWriteBuffer . . . 1921
13.162.4.19 zip64local_getByte . . . 1921
13.162.4.20 zip64local_getLong . . . 1922
13.162.4.21 zip64local_getLong64 . . . 1923
13.162.4.22 zip64local_getShort . . . 1924

13.162.4.23 zip64local_putValue . . . 1925
13.162.4.24 zip64local_putValue_inmemory . . . 1926
13.162.4.25 zip64local_SearchCentralDir . . . 1926
13.162.4.26 zip64local_SearchCentralDir64 . . . 1927
13.162.4.27 zip64local_TmzDateToDosDate . . . 1929
13.162.4.28 zipClearFlags . . . 1930
13.162.4.29 zipClose . . . 1930
13.162.4.30 zipCloseFileInZip . . . 1932
13.162.4.31 zipCloseFileInZipRaw . . . 1933
13.162.4.32 zipCloseFileInZipRaw64 . . . 1934
13.162.4.33 zipOpen . . . 1938
13.162.4.34 zipOpen2 . . . 1938
13.162.4.35 zipOpen2_64 . . . 1939
13.162.4.36 zipOpen3 . . . 1940
13.162.4.37 zipOpen64 . . . 1942
13.162.4.38 zipOpenNewFileInZip . . . 1943
13.162.4.39 zipOpenNewFileInZip2 . . . 1943
13.162.4.40 zipOpenNewFileInZip2_64 . . . 1944
13.162.4.41 zipOpenNewFileInZip3 . . . 1944
13.162.4.42 zipOpenNewFileInZip3_64 . . . 1945
13.162.4.43 zipOpenNewFileInZip4 . . . 1945
13.162.4.44 zipOpenNewFileInZip4_64 . . . 1946
13.162.4.45 zipOpenNewFileInZip64 . . . 1950
13.162.4.46 zipRemoveExtraInfoBlock . . . 1950
13.162.4.47 zipSetFlags . . . 1951
13.162.4.48 zipWriteInFileInZip . . . 1952
13.162.5 Variable Documentation . . . 1954
13.162.5.1 zip_copyright . . . 1954
13.163 src/ckb/quazip/zip.h File Reference . . . 1954
13.163.1 Data Structure Documentation . . . 1956
13.163.1.1 struct tm_zip_s . . . 1956
13.163.1.2 struct zip_fileinfo . . . 1957
13.163.2 Macro Definition Documentation . . . 1958
13.163.2.1 APPEND_STATUS_ADDINZIP . . . 1958
13.163.2.2 APPEND_STATUS_CREATE . . . 1958
13.163.2.3 APPEND_STATUS_CREATEAFTER . . . 1959
13.163.2.4 DEF_MEM_LEVEL . . . 1959
13.163.2.5 Z_BZIP2ED . . . 1959
13.163.2.6 ZIP_AUTO_CLOSE . . . 1959
13.163.2.7 ZIP_BADZIPFILE . . . 1959

13.163.2.8 ZIP_DEFAULT_FLAGS . . . 1959
13.163.2.9 ZIP_EOF . . . 1959
13.163.2.10 ZIP_ERRNO . . . 1959
13.163.2.11 ZIP_INTERNALERROR . . . 1959
13.163.2.12 ZIP_OK . . . 1960
13.163.2.13 ZIP_PARAMERROR . . . 1960
13.163.2.14 ZIP_SEQUENTIAL . . . 1960
13.163.2.15 ZIP_WRITE_DATA_DESCRIPTOR . . . 1960
13.163.3 Typedef Documentation . . . 1960
13.163.3.1 tm_zip . . . 1960
13.163.3.2 zipcharpc . . . 1960
13.163.3.3 zipFile . . . 1960
13.163.4 Function Documentation . . . 1960
13.163.4.1 _Z_OF . . . 1960
13.163.4.2 _Z_OF . . . 1960
13.163.4.3 _Z_OF . . . 1960
13.163.4.4 _Z_OF . . . 1961
13.163.4.5 _Z_OF . . . 1961
13.163.4.6 _Z_OF . . . 1961
13.163.4.7 _Z_OF . . . 1961
13.163.4.8 _Z_OF . . . 1961
13.163.4.9 _Z_OF . . . 1961
13.163.4.10 _Z_OF . . . 1961
13.163.4.11 _Z_OF . . . 1961
13.163.4.12 _Z_OF . . . 1961
13.163.4.13 _Z_OF . . . 1961
13.163.4.14 _Z_OF . . . 1961
13.163.4.15 _Z_OF . . . 1961
13.163.4.16 _Z_OF . . . 1961
13.163.4.17 _Z_OF . . . 1961
13.163.4.18 zipClearFlags . . . 1961
13.163.4.19 zipOpen3 . . . 1962
13.163.4.20 zipSetFlags . . . 1964
13.164 src/ckb/rebindwidget.cpp File Reference . . . 1965
13.164.1 Variable Documentation . . . 1965
13.164.1.1 DPI_CUST_IDX . . . 1965
13.164.1.2 DPI_OFFSET . . . 1965
13.165 src/ckb/rebindwidget.h File Reference . . . 1965
13.166 src/ckb/release/moc_animadddialog.cpp File Reference . . . 1966
13.166.1 Data Structure Documentation . . . 1967

13.166.1.1 struct qt_meta_stringdata_AnimAddDialog_t . . . 1967
13.166.2 Macro Definition Documentation . . . 1968
13.166.2.1 QT_MOC_LITERAL . . . 1968
13.166.3 Variable Documentation . . . 1968
13.166.3.1 qt_meta_data_AnimAddDialog . . . 1968
13.166.3.2 qt_meta_stringdata_AnimAddDialog . . . 1969
13.167 src/ckb/release/moc_animscript.cpp File Reference . . . 1969
13.167.1 Data Structure Documentation . . . 1970
13.167.1.1 struct qt_meta_stringdata_AnimScript_t . . . 1970
13.167.2 Macro Definition Documentation . . . 1971
13.167.2.1 QT_MOC_LITERAL . . . 1971
13.167.3 Variable Documentation . . . 1971
13.167.3.1 qt_meta_data_AnimScript . . . 1971
13.167.3.2 qt_meta_stringdata_AnimScript . . . 1972
13.168 src/ckb/release/moc_animsettingdialog.cpp File Reference . . . 1972
13.168.1 Data Structure Documentation . . . 1973
13.168.1.1 struct qt_meta_stringdata_AnimSettingDialog_t . . . 1973
13.168.2 Macro Definition Documentation . . . 1973
13.168.2.1 QT_MOC_LITERAL . . . 1973
13.168.3 Variable Documentation . . . 1974
13.168.3.1 qt_meta_data_AnimSettingDialog . . . 1974
13.168.3.2 qt_meta_stringdata_AnimSettingDialog . . . 1974
13.169 src/ckb/release/moc_ckbsettingswriter.cpp File Reference . . . 1975
13.169.1 Data Structure Documentation . . . 1975
13.169.1.1 struct qt_meta_stringdata_CkbSettingsWriter_t . . . 1975
13.169.2 Macro Definition Documentation . . . 1976
13.169.2.1 QT_MOC_LITERAL . . . 1976
13.169.3 Variable Documentation . . . 1976
13.169.3.1 qt_meta_data_CkbSettingsWriter . . . 1976
13.169.3.2 qt_meta_stringdata_CkbSettingsWriter . . . 1977
13.170 src/ckb/release/moc_colorbutton.cpp File Reference . . . 1977
13.170.1 Data Structure Documentation . . . 1978
13.170.1.1 struct qt_meta_stringdata_ColorButton_t . . . 1978
13.170.2 Macro Definition Documentation . . . 1978
13.170.2.1 QT_MOC_LITERAL . . . 1978
13.170.3 Variable Documentation . . . 1979
13.170.3.1 qt_meta_data_ColorButton . . . 1979
13.170.3.2 qt_meta_stringdata_ColorButton . . . 1979
13.171 src/ckb/release/moc_extrasettingswidget.cpp File Reference . . . 1979
13.171.1 Data Structure Documentation . . . 1980

13.171.1.1 struct qt_meta_stringdata_ExtraSettingsWidget_t . . . 1980
13.171.2 Macro Definition Documentation . . . 1981
13.171.2.1 QT_MOC_LITERAL . . . 1981
13.171.3 Variable Documentation . . . 1981
13.171.3.1 qt_meta_data_ExtraSettingsWidget . . . 1981
13.171.3.2 qt_meta_stringdata_ExtraSettingsWidget . . . 1981
13.172 src/ckb/release/moc_fwupgradedialog.cpp File Reference . . . 1982
13.172.1 Data Structure Documentation . . . 1983
13.172.1.1 struct qt_meta_stringdata_FwUpgradeDialog_t . . . 1983
13.172.2 Macro Definition Documentation . . . 1984
13.172.2.1 QT_MOC_LITERAL . . . 1984
13.172.3 Variable Documentation . . . 1984
13.172.3.1 qt_meta_data_FwUpgradeDialog . . . 1984
13.172.3.2 qt_meta_stringdata_FwUpgradeDialog . . . 1985
13.173 src/ckb/release/moc_gradientbutton.cpp File Reference . . . 1985
13.173.1 Data Structure Documentation . . . 1986
13.173.1.1 struct qt_meta_stringdata_GradientButton_t . . . 1986
13.173.2 Macro Definition Documentation . . . 1987
13.173.2.1 QT_MOC_LITERAL . . . 1987
13.173.3 Variable Documentation . . . 1987
13.173.3.1 qt_meta_data_GradientButton . . . 1987
13.173.3.2 qt_meta_stringdata_GradientButton . . . 1988
13.174 src/ckb/release/moc_gradientdialog.cpp File Reference . . . 1988
13.174.1 Data Structure Documentation . . . 1989
13.174.1.1 struct qt_meta_stringdata_GradientDialog_t . . . 1989
13.174.2 Macro Definition Documentation . . . 1989
13.174.2.1 QT_MOC_LITERAL . . . 1989
13.174.3 Variable Documentation . . . 1990
13.174.3.1 qt_meta_data_GradientDialog . . . 1990
13.174.3.2 qt_meta_stringdata_GradientDialog . . . 1990
13.175 src/ckb/release/moc_gradientdialogwidget.cpp File Reference . . . 1991
13.175.1 Data Structure Documentation . . . 1991
13.175.1.1 struct qt_meta_stringdata_GradientDialogWidget_t . . . 1991
13.175.2 Macro Definition Documentation . . . 1992
13.175.2.1 QT_MOC_LITERAL . . . 1992
13.175.3 Variable Documentation . . . 1992
13.175.3.1 qt_meta_data_GradientDialogWidget . . . 1992
13.175.3.2 qt_meta_stringdata_GradientDialogWidget . . . 1993
13.176 src/ckb/release/moc_kb.cpp File Reference . . . 1993
13.176.1 Data Structure Documentation . . . 1994

13.176.1.1 struct qt_meta_stringdata_Kb_t . . . 1994
13.176.2 Macro Definition Documentation . . . 1995
13.176.2.1 QT_MOC_LITERAL . . . 1995
13.176.3 Variable Documentation . . . 1995
13.176.3.1 qt_meta_data_Kb . . . 1995
13.176.3.2 qt_meta_stringdata_Kb . . . 1995
13.177 src/ckb/release/moc_kbanim.cpp File Reference . . . 1996
13.177.1 Data Structure Documentation . . . 1996
13.177.1.1 struct qt_meta_stringdata_KbAnim_t . . . 1996
13.177.2 Macro Definition Documentation . . . 1997
13.177.2.1 QT_MOC_LITERAL . . . 1997
13.177.3 Variable Documentation . . . 1997
13.177.3.1 qt_meta_data_KbAnim . . . 1997
13.177.3.2 qt_meta_stringdata_KbAnim . . . 1998
13.178 src/ckb/release/moc_kbanimwidget.cpp File Reference . . . 1998
13.178.1 Data Structure Documentation . . . 1999
13.178.1.1 struct qt_meta_stringdata_KbAnimWidget_t . . . 1999
13.178.2 Macro Definition Documentation . . . 2000
13.178.2.1 QT_MOC_LITERAL . . . 2000
13.178.3 Variable Documentation . . . 2000
13.178.3.1 qt_meta_data_KbAnimWidget . . . 2000
13.178.3.2 qt_meta_stringdata_KbAnimWidget . . . 2000
13.179 src/ckb/release/moc_kbbind.cpp File Reference . . . 2001
13.179.1 Data Structure Documentation . . . 2001
13.179.1.1 struct qt_meta_stringdata_KbBind_t . . . 2001
13.179.2 Macro Definition Documentation . . . 2002
13.179.2.1 QT_MOC_LITERAL . . . 2002
13.179.3 Variable Documentation . . . 2002
13.179.3.1 qt_meta_data_KbBind . . . 2002
13.179.3.2 qt_meta_stringdata_KbBind . . . 2002
13.180 src/ckb/release/moc_kbbindwidget.cpp File Reference . . . 2003
13.180.1 Data Structure Documentation . . . 2004
13.180.1.1 struct qt_meta_stringdata_KbBindWidget_t . . . 2004
13.180.2 Macro Definition Documentation . . . 2004
13.180.2.1 QT_MOC_LITERAL . . . 2004
13.180.3 Variable Documentation . . . 2005
13.180.3.1 qt_meta_data_KbBindWidget . . . 2005
13.180.3.2 qt_meta_stringdata_KbBindWidget . . . 2005
13.181 src/ckb/release/moc_kbfirmware.cpp File Reference . . . 2005
13.181.1 Data Structure Documentation . . . 2006

13.181.1.1 struct qt_meta_stringdata_KbFirmware_t . . . 2006
13.181.2 Macro Definition Documentation . . . 2007
13.181.2.1 QT_MOC_LITERAL . . . 2007
13.181.3 Variable Documentation . . . 2007
13.181.3.1 qt_meta_data_KbFirmware . . . 2007
13.181.3.2 qt_meta_stringdata_KbFirmware . . . 2008
13.182 src/ckb/release/moc_kblight.cpp File Reference . . . 2008
13.182.1 Data Structure Documentation . . . 2009
13.182.1.1 struct qt_meta_stringdata_KbLight_t . . . 2009
13.182.2 Macro Definition Documentation . . . 2010
13.182.2.1 QT_MOC_LITERAL . . . 2010
13.182.3 Variable Documentation . . . 2010
13.182.3.1 qt_meta_data_KbLight . . . 2010
13.182.3.2 qt_meta_stringdata_KbLight . . . 2011
13.183 src/ckb/release/moc_kblightwidget.cpp File Reference . . . 2011
13.183.1 Data Structure Documentation . . . 2012
13.183.1.1 struct qt_meta_stringdata_KbLightWidget_t . . . 2012
13.183.2 Macro Definition Documentation . . . 2013
13.183.2.1 QT_MOC_LITERAL . . . 2013
13.183.3 Variable Documentation . . . 2013
13.183.3.1 qt_meta_data_KbLightWidget . . . 2013
13.183.3.2 qt_meta_stringdata_KbLightWidget . . . 2013
13.184 src/ckb/release/moc_kbmanager.cpp File Reference . . . 2014
13.184.1 Data Structure Documentation . . . 2014
13.184.1.1 struct qt_meta_stringdata_KbManager_t . . . 2014
13.184.2 Macro Definition Documentation . . . 2015
13.184.2.1 QT_MOC_LITERAL . . . 2015
13.184.3 Variable Documentation . . . 2015
13.184.3.1 qt_meta_data_KbManager . . . 2015
13.184.3.2 qt_meta_stringdata_KbManager . . . 2015
13.185 src/ckb/release/moc_kbmode.cpp File Reference . . . 2016
13.185.1 Data Structure Documentation . . . 2017
13.185.1.1 struct qt_meta_stringdata_KbMode_t . . . 2017
13.185.2 Macro Definition Documentation . . . 2017
13.185.2.1 QT_MOC_LITERAL . . . 2017
13.185.3 Variable Documentation . . . 2018
13.185.3.1 qt_meta_data_KbMode . . . 2018
13.185.3.2 qt_meta_stringdata_KbMode . . . 2018
13.186 src/ckb/release/moc_kbperf.cpp File Reference . . . 2018
13.186.1 Data Structure Documentation . . . 2019

13.186.1.1 struct qt_meta_stringdata_KbPerf_t . . . 2019
13.186.2 Macro Definition Documentation . . . 2020
13.186.2.1 QT_MOC_LITERAL . . . 2020
13.186.3 Variable Documentation . . . 2020
13.186.3.1 qt_meta_data_KbPerf . . . 2020
13.186.3.2 qt_meta_stringdata_KbPerf . . . 2021
13.187 src/ckb/release/moc_kbprofile.cpp File Reference . . . 2021
13.187.1 Data Structure Documentation . . . 2022
13.187.1.1 struct qt_meta_stringdata_KbProfile_t . . . 2022
13.187.2 Macro Definition Documentation . . . 2022
13.187.2.1 QT_MOC_LITERAL . . . 2022
13.187.3 Variable Documentation . . . 2023
13.187.3.1 qt_meta_data_KbProfile . . . 2023
13.187.3.2 qt_meta_stringdata_KbProfile . . . 2023
13.188 src/ckb/release/moc_kbprofiledialog.cpp File Reference . . . 2023
13.188.1 Data Structure Documentation . . . 2024
13.188.1.1 struct qt_meta_stringdata_KbProfileDialog_t . . . 2024
13.188.2 Macro Definition Documentation . . . 2025
13.188.2.1 QT_MOC_LITERAL . . . 2025
13.188.3 Variable Documentation . . . 2025
13.188.3.1 qt_meta_data_KbProfileDialog . . . 2025
13.188.3.2 qt_meta_stringdata_KbProfileDialog . . . 2026
13.189 src/ckb/release/moc_kbwidget.cpp File Reference . . . 2027
13.189.1 Data Structure Documentation . . . 2027
13.189.1.1 struct qt_meta_stringdata_KbWidget_t . . . 2027
13.189.2 Macro Definition Documentation . . . 2028
13.189.2.1 QT_MOC_LITERAL . . . 2028
13.189.3 Variable Documentation . . . 2028
13.189.3.1 qt_meta_data_KbWidget . . . 2028
13.189.3.2 qt_meta_stringdata_KbWidget . . . 2028
13.190 src/ckb/release/moc_keyaction.cpp File Reference . . . 2029
13.190.1 Data Structure Documentation . . . 2029
13.190.1.1 struct qt_meta_stringdata_KeyAction_t . . . 2029
13.190.2 Macro Definition Documentation . . . 2030
13.190.2.1 QT_MOC_LITERAL . . . 2030
13.190.3 Variable Documentation . . . 2030
13.190.3.1 qt_meta_data_KeyAction . . . 2030
13.190.3.2 qt_meta_stringdata_KeyAction . . . 2031
13.191 src/ckb/release/moc_keywidget.cpp File Reference . . . 2031
13.191.1 Data Structure Documentation . . . 2032

13.191.1.1 struct qt_meta_stringdata_KeyWidget_t . . . 2032
13.191.2 Macro Definition Documentation . . . 2032
13.191.2.1 QT_MOC_LITERAL . . . 2032
13.191.3 Variable Documentation . . . 2033
13.191.3.1 qt_meta_data_KeyWidget . . . 2033
13.191.3.2 qt_meta_stringdata_KeyWidget . . . 2033
13.192 src/ckb/release/moc_kperfwidget.cpp File Reference . . . 2033
13.192.1 Data Structure Documentation . . . 2034
13.192.1.1 struct qt_meta_stringdata_KPerfWidget_t . . . 2034
13.192.2 Macro Definition Documentation . . . 2035
13.192.2.1 QT_MOC_LITERAL . . . 2035
13.192.3 Variable Documentation . . . 2035
13.192.3.1 qt_meta_data_KPerfWidget . . . 2035
13.192.3.2 qt_meta_stringdata_KPerfWidget . . . 2036
13.193 src/ckb/release/moc_layoutdialog.cpp File Reference . . . 2036
13.193.1 Data Structure Documentation . . . 2037
13.193.1.1 struct qt_meta_stringdata_LayoutDialog_t . . . 2037
13.193.2 Macro Definition Documentation . . . 2038
13.193.2.1 QT_MOC_LITERAL . . . 2038
13.193.3 Variable Documentation . . . 2038
13.193.3.1 qt_meta_data_LayoutDialog . . . 2038
13.193.3.2 qt_meta_stringdata_LayoutDialog . . . 2039
13.194 src/ckb/release/moc_macroreader.cpp File Reference . . . 2039
13.194.1 Data Structure Documentation . . . 2040
13.194.1.1 struct qt_meta_stringdata_MacroReaderThread_t . . . 2040
13.194.1.2 struct qt_meta_stringdata_MacroReader_t . . . 2040
13.194.2 Macro Definition Documentation . . . 2041
13.194.2.1 QT_MOC_LITERAL . . . 2041
13.194.2.2 QT_MOC_LITERAL . . . 2041
13.194.3 Variable Documentation . . . 2041
13.194.3.1 qt_meta_data_MacroReader . . . 2042
13.194.3.2 qt_meta_data_MacroReaderThread . . . 2042
13.194.3.3 qt_meta_stringdata_MacroReader . . . 2042
13.194.3.4 qt_meta_stringdata_MacroReaderThread . . . 2042
13.195 src/ckb/release/moc_mainwindow.cpp File Reference . . . 2043
13.195.1 Data Structure Documentation . . . 2044
13.195.1.1 struct qt_meta_stringdata_MainWindow_t . . . 2044
13.195.2 Macro Definition Documentation . . . 2044
13.195.2.1 QT_MOC_LITERAL . . . 2044
13.195.3 Variable Documentation . . . 2044

13.195.3.1 qt_meta_data_MainWindow . . . 2044
13.195.3.2 qt_meta_stringdata_MainWindow . . . 2045
13.196 src/ckb/release/moc_modeselectdialog.cpp File Reference . . . 2045
13.196.1 Data Structure Documentation . . . 2045
13.196.1.1 struct qt_meta_stringdata_ModeSelectDialog_t . . . 2045
13.196.2 Macro Definition Documentation . . . 2046
13.196.2.1 QT_MOC_LITERAL . . . 2046
13.196.3 Variable Documentation . . . 2046
13.196.3.1 qt_meta_data_ModeSelectDialog . . . 2046
13.196.3.2 qt_meta_stringdata_ModeSelectDialog . . . 2047
13.197 src/ckb/release/moc_mperfwidget.cpp File Reference . . . 2047
13.197.1 Data Structure Documentation . . . 2048
13.197.1.1 struct qt_meta_stringdata_MPerfWidget_t . . . 2048
13.197.2 Macro Definition Documentation . . . 2048
13.197.2.1 QT_MOC_LITERAL . . . 2048
13.197.3 Variable Documentation . . . 2049
13.197.3.1 qt_meta_data_MPerfWidget . . . 2049
13.197.3.2 qt_meta_stringdata_MPerfWidget . . . 2049
13.198 src/ckb/release/moc_quagzipfile.cpp File Reference . . . 2050
13.198.1 Data Structure Documentation . . . 2050
13.198.1.1 struct qt_meta_stringdata_QuaGzipFile_t . . . 2050
13.198.2 Macro Definition Documentation . . . 2051
13.198.2.1 QT_MOC_LITERAL . . . 2051
13.198.3 Variable Documentation . . . 2051
13.198.3.1 qt_meta_data_QuaGzipFile . . . 2051
13.198.3.2 qt_meta_stringdata_QuaGzipFile . . . 2052
13.199 src/ckb/release/moc_quaziodevice.cpp File Reference . . . 2052
13.199.1 Data Structure Documentation . . . 2053
13.199.1.1 struct qt_meta_stringdata_QuaZIODevice_t . . . 2053
13.199.2 Macro Definition Documentation . . . 2053
13.199.2.1 QT_MOC_LITERAL . . . 2053
13.199.3 Variable Documentation . . . 2054
13.199.3.1 qt_meta_data_QuaZIODevice . . . 2054
13.199.3.2 qt_meta_stringdata_QuaZIODevice . . . 2054
13.200 src/ckb/release/moc_quazipfile.cpp File Reference . . . 2054
13.200.1 Data Structure Documentation . . . 2055
13.200.1.1 struct qt_meta_stringdata_QuaZipFile_t . . . 2055
13.200.2 Macro Definition Documentation . . . 2055
13.200.2.1 QT_MOC_LITERAL . . . 2055
13.200.3 Variable Documentation . . . 2056

13.200.3.1 qt_meta_data_QuaZipFile . . . 2056
13.200.3.2 qt_meta_stringdata_QuaZipFile . . . 2056
13.201 src/ckb/release/moc_rebindwidget.cpp File Reference . . . 2056
13.201.1 Data Structure Documentation . . . 2057
13.201.1.1 struct qt_meta_stringdata_RebindWidget_t . . . 2057
13.201.2 Macro Definition Documentation . . . 2058
13.201.2.1 QT_MOC_LITERAL . . . 2058
13.201.3 Variable Documentation . . . 2058
13.201.3.1 qt_meta_data_RebindWidget . . . 2058
13.201.3.2 qt_meta_stringdata_RebindWidget . . . 2058
13.202 src/ckb/release/moc_rlistwidget.cpp File Reference . . . 2059
13.202.1 Data Structure Documentation . . . 2059
13.202.1.1 struct qt_meta_stringdata_RListWidget_t . . . 2059
13.202.2 Macro Definition Documentation . . . 2060
13.202.2.1 QT_MOC_LITERAL . . . 2060
13.202.3 Variable Documentation . . . 2060
13.202.3.1 qt_meta_data_RListWidget . . . 2060
13.202.3.2 qt_meta_stringdata_RListWidget . . . 2060
13.203 src/ckb/release/moc_settingswidget.cpp File Reference . . . 2061
13.203.1 Data Structure Documentation . . . 2062
13.203.1.1 struct qt_meta_stringdata_SettingsWidget_t . . . 2062
13.203.2 Macro Definition Documentation . . . 2062
13.203.2.1 QT_MOC_LITERAL . . . 2062
13.203.3 Variable Documentation . . . 2063
13.203.3.1 qt_meta_data_SettingsWidget . . . 2063
13.203.3.2 qt_meta_stringdata_SettingsWidget . . . 2063
13.204 src/ckb/release/qrc_binary.cpp File Reference . . . 2063
13.204.1 Macro Definition Documentation . . . 2064
13.204.1.1 QT_RCC_MANGLE_NAMESPACE . . . 2064
13.204.1.2 QT_RCC_PREPEND_NAMESPACE . . . 2064
13.204.2 Function Documentation . . . 2064
13.204.2.1 qCleanupResources_binary . . . 2064
13.204.2.2 qInitResources_binary . . . 2065
13.204.2.3 qRegisterResourceData . . . 2065
13.204.2.4 qUnregisterResourceData . . . 2065
13.204.3 Variable Documentation . . . 2066
13.204.3.1 qt_resource_data . . . 2066
13.204.3.2 qt_resource_name . . . 2066
13.204.3.3 qt_resource_struct . . . 2066
13.205 src/ckb/release/qrc_image.cpp File Reference . . . 2066

13.205.1 Macro Definition Documentation . . . 2067
13.205.1.1 QT_RCC_MANGLE_NAMESPACE . . . 2067
13.205.1.2 QT_RCC_PREPEND_NAMESPACE . . . 2067
13.205.2 Function Documentation . . . 2067
13.205.2.1 qCleanupResources_image . . . 2067
13.205.2.2 qInitResources_image . . . 2067
13.205.2.3 qRegisterResourceData . . . 2068
13.205.2.4 qUnregisterResourceData . . . 2068
13.205.3 Variable Documentation . . . 2068
13.205.3.1 qt_resource_data . . . 2068
13.205.3.2 qt_resource_name . . . 2068
13.205.3.3 qt_resource_struct . . . 2069
13.206 src/ckb/release/qrc_text.cpp File Reference . . . 2069
13.206.1 Macro Definition Documentation . . . 2069
13.206.1.1 QT_RCC_MANGLE_NAMESPACE . . . 2069
13.206.1.2 QT_RCC_PREPEND_NAMESPACE . . . 2069
13.206.2 Function Documentation . . . 2069
13.206.2.1 qCleanupResources_text . . . 2069
13.206.2.2 qInitResources_text . . . 2070
13.206.2.3 qRegisterResourceData . . . 2070
13.206.2.4 qUnregisterResourceData . . . 2071
13.206.3 Variable Documentation . . . 2071
13.206.3.1 qt_resource_data . . . 2071
13.206.3.2 qt_resource_name . . . 2071
13.206.3.3 qt_resource_struct . . . 2071
13.207 src/ckb/rlistwidget.cpp File Reference . . . 2072
13.208 src/ckb/rlistwidget.h File Reference . . . 2072
13.209 src/ckb/settingswidget.cpp File Reference . . . 2073
13.209.1 Function Documentation . . . 2074
13.209.1.1 right . . . 2074
13.209.2 Variable Documentation . . . 2074
13.209.2.1 devpath . . . 2074
13.209.2.2 modKeys . . . 2074
13.209.2.3 modNames . . . 2075
13.210 src/ckb/settingswidget.h File Reference . . . 2075
13.211 src/ckb/ui_animadddialog.h File Reference . . . 2076
13.212 src/ckb/ui_animsettingdialog.h File Reference . . . 2076
13.213 src/ckb/ui_extrasettingswidget.h File Reference . . . 2077
13.214 src/ckb/ui_fwupgradedialog.h File Reference . . . 2078
13.215 src/ckb/ui_gradientdialog.h File Reference . . . 2079

13.216 src/ckb/ui_kbanimwidget.h File Reference 2080
13.217 src/ckb/ui_kbbindwidget.h File Reference 2081
13.218 src/ckb/ui_kblightwidget.h File Reference 2082
13.219 src/ckb/ui_kbprofiledialog.h File Reference 2083
13.220 src/ckb/ui_kbwidget.h File Reference 2084
13.221 src/ckb/ui_kperfwidget.h File Reference 2085
13.222 src/ckb/ui_layoutdialog.h File Reference 2086
13.223 src/ckb/ui_mainwindow.h File Reference 2087
13.224 src/ckb/ui_modeselectdialog.h File Reference 2088
13.225 src/ckb/ui_mperfwidget.h File Reference 2089
13.226 src/ckb/ui_rebindwidget.h File Reference 2090
13.227 src/ckb/ui_settingswidget.h File Reference 2091

Index **2093**

Chapter 1

README

ckb-next: RGB Driver for Linux and macOS

ckb-next is an open-source driver for Corsair keyboards and mice. It aims to bring the features of their proprietary CUE software to the Linux and Mac operating systems. This project is currently a work in progress, but it already supports much of the same functionality, including full RGB animations. More features are coming soon. Testing and bug reports are appreciated!

Disclaimer: ckb-next is not an official Corsair product. It is licensed under the GNU General Public License (version 2) in the hope that it will be useful, but with NO WARRANTY of any kind.

- Device Support
  - Keyboards
  - Mice
- Linux Installation
  - Pre-made packages
  - Preparation
  - Installing
  - Upgrading
  - Uninstalling
- OS X/macOS Installation
  - Binary download
  - Building from source
  - Upgrading (binary)
  - Upgrading (source)
  - Uninstalling
- Usage
  - Major features
  - Roadmap
- Troubleshooting
  - Linux

    - OS X/macOS
    - General
    - Reporting issues
- Known issues
- Contributing
- Contact us
- What happened to the original ckb

See also:

- https://github.com/mattanger/ckb-next/blob/master/DAEMON.md "Manual for the driver daemon"

Device Support

Keyboards

- K63
- K65:
    - RGB
    - non-RGB
    - LUX RGB
    - RGB RAPIDFIRE
- K68
- K70:
    - RGB
    - non-RGB
    - LUX RGB
    - LUX non-RGB
    - RGB RAPIDFIRE
    - non-RGB RAPIDFIRE
- K95:
    - RGB
    - non-RGB∗
    - Platinum∗∗
- Strafe:
    - RGB
    - non-RGB
- = hardware playback not supported. Settings will be saved to software only.

∗∗ = partial support, static hardware playback only.

Mice

- M65:
  - non-RGB
  - PRO RGB
- Sabre:
  - Optical RGB
  - Laser RGB
- Scimitar:
  - RGB
  - PRO RGB
- Harpoon
- Glaive

Linux Installation

Pre-made packages

- Fedora 24/25, CentOS/RHEL 7 (maintained by):
  - `'johanh/ckb'` - based on `master` branch
- Arch Linux (maintained by ,):
  - `'aur/ckb-next'` - based on GitHub releases
  - `'aur/ckb-next-git'` - based on `master` branch
  - `'aur/ckb-next-latest-git'` - based on `newdev` branch
  - `'aur/ckb-next-newfw-git'` - based on `newfw` branch

If you are a package maintainer or want to discuss something with package maintainers let us know in #5, so we can have an accountable and centralized communication about this. *If you would like to maintain a package for your favorite distro/OS, please let us know as well.*

Preparation

ckb-next requires Qt5 (Qt 5.9 is recommended), libudev, zlib, gcc, g++, and glibc.

- Ubuntu: `sudo apt-get install build-essential libudev-dev qt5-default zlib1g-dev libappindicator-dev`
- Fedora: `sudo dnf install zlib-devel qt5-qtbase-devel libgudev-devel libappindicator-de systemd-devel gcc-c++`
- Arch: `sudo pacman -S base-devel qt5-base zlib`
- Other distros: Look for `qt5` or `libqt5*-devel`

Note: If you build your own kernels, ckb-next requires the `CONFIG_INPUT_UINPUT` flag to be enabled. It is located in `Device Drivers -> Input Device Support -> Miscellaneous devices -> User level driver support`. If you don't know what this means, you can ignore this.

Installing

You can download ckb-next using the "Download zip" option on the right or clone it using `git clone`. Extract it and open the ckb-master directory in a terminal. Run `./quickinstall`. It will attempt to build ckb and then ask if you'd like to install/run the application. If the build doesn't succeed, or if you'd like to hand-tune the compilation of ckb, see https://github.com/mattanger/ckb-next/blob/master/BUILD.md "'BUILD.md'" for instructions.

Upgrading

To install a new version of ckb, or to reinstall the same version, first delete the ckb-master directory and the zip file from your previous download. Then download the source code again and re-run `./quickinstall`. The script will automatically replace the previous installation. You may need to reboot afterward.

Uninstalling

First, stop the ckb-daemon service and remove the service file.

- If you have systemd (Ubuntu versions starting with 15.04): "' sudo systemctl stop ckb-daemon sudo rm -f /usr/lib/systemd/system/ckb-daemon.service "'
- If you have Upstart (Ubuntu versions earlier than 15.04): "' sudo service ckb-daemon stop sudo rm -f /etc/init/ckb-daemon.conf "'
- If you have OpenRC: "' sudo rc-service ckb-daemon stop sudo rc-update del ckb-daemon default sudo rm -f /etc/init.d/ckb-daemon "
- `If you're not sure, re-run the`quickinstall`script and proceed to the service installation. The script will say`System service: Upstart detected`or`System service: systemd detected'. Please be aware that OpenRC is currently not detected automatically.

Afterward, remove the applications and related files: "' sudo rm -f /usr/bin/ckb /usr/bin/ckb-daemon /usr/share/applications/ckb.desktop /usr/share/icons/hicolor/512x512/apps/ckb.png sudo rm -rf /usr/lib/ckb-animations "'

Before https://github.com/mattanger/ckb-next/commit/f347e60df211c60452f95084b6c46dc4ec5f42 animations were located elsewhere, try removing them as well: "' sudo rm -rf /usr/bin/ckb-animations "'

OS X/macOS Installation

Binary download

macOS `pkg` can be downloaded from `GitHub Releases`. It is always built with the last available stable Qt version and tagrets 10.10 SDK. If you run 10.9.x, you'll need to build the project from source and comment out `src/ckb-heat` (and the backslash above it) inside `ckb.pro`.

Building from source

Install the latest version of Xcode from the App Store. While it's downloading, open the Terminal and execute `xcode-select --install` to install Command Line Tools. Then open Xcode, accept the license agreement and wait for it to install any additional components (if necessary). When you see the "Welcome to Xcode" screen, from the top bar choose `Xcode -> Preferences -> Locations -> Command Line Tools` and select an SDK version. Afterwards install `Homebrew` and execute `brew install qt5` in the Terminal.

> **Note**: If you decide to use the official Qt5 package from Qt website instead, you will have to edit the installation script and provide installation paths manually due to a qmake bug.

The easiest way to build the driver is with the `quickinstall` script, which is present in the ckb-master folder. Double-click on `quickinstall` and it will compile the app for you, then ask if you'd like to install it system-wide. If the build fails for any reason, or if you'd like to compile and install manually, see https://github.com/ccMS-C/ckb/blob/master/BUILD.md "'BUILD.md'".

Upgrading (binary)

Download the latest `ckb.pkg`, run the installer, and reboot. The newly-installed driver will replace the old one.

Upgrading (source)

Remove the existing ckb-master directory and zip file. Re-download the source code and run the `quickinstall` script again. The script will automatically replace the previous installation. You may need to reboot afterward.

Uninstalling

Drag `ckb.app` into the trash. Then stop and remove the agent:

"'sh sudo unload /Library/LaunchDaemons/com.ckb.daemon.plist sudo rm /Library/LaunchDaemons/com.ckb.-daemon.plist "'

Usage

The user interface is still a work in progress.

Major features

- Control multiple devices independently
- United States and European keyboard layouts
- Customizable key bindings
- Per-key lighting and animation
- Reactive lighting
- Multiple profiles/modes with hardware save function
- Adjustable mouse DPI with ability to change DPI on button press

Closing ckb will actually minimize it to the system tray. Use the Quit option from the tray icon or the settings screen to exit the application.

Roadmap

- **v0.3 release:**
- Ability to store profiles separately from devices, import/export them
- More functions for the Win Lock key
- Key macros
- **v0.4 release:**
- Ability to import CUE profiles

- Ability to tie profiles to which application has focus
- **v0.5 release:**
- Key combos
- Timers?
- **v1.0 release:**
- OSD? (Not sure if this can actually be done)
- Extra settings?
- ????

Troubleshooting

Linux

If you have problems connecting the device to your system (device doesn't respond, ckb-daemon doesn't recognize or can't connect it) and/or you experience long boot times when using the keyboard, try adding the following to your kernel's `cmdline`:

- K65 RGB: `usbhid.quirks=0x1B1C:0x1B17:0x20000408`
- K65 LUX: `usbhid.quirks=0x1B1C:0x1B37:0x20000408`
- K68: `usbhid.quirks=0x1B1C:0x1B3F:0x20000408`
- K70: `usbhid.quirks=0x1B1C:0x1B09:0x20000408`
- K70 LUX: `usbhid.quirks=0x1B1C:0x1B36:0x20000408`
- K70 RGB: `usbhid.quirks=0x1B1C:0x1B13:0x20000408`
- K95: `usbhid.quirks=0x1B1C:0x1B08:0x20000408`
- K95 RGB: `usbhid.quirks=0x1B1C:0x1B11:0x20000408`
- Strafe: `usbhid.quirks=0x1B1C:0x1B15:0x20000408`
- Strafe RGB: `usbhid.quirks=0x1B1C:0x1B20:0x20000408`
- M65 RGB: `usbhid.quirks=0x1B1C:0x1B12:0x20000408`
- Sabre RGB Optical: `usbhid.quirks=0x1B1C:0x1B14:0x20000408`
- Sabre RGB Laser: `usbhid.quirks=0x1B1C:0x1B19:0x20000408`
- Scimitar RGB: `usbhid.quirks=0x1B1C:0x1B1E:0x20000408`

For instructions on adding `cmdline` parameters in Ubuntu, see `https://wiki.ubuntu.com/Kernel/KernelBootParameters`

If you have multiple devices, combine them with commas, starting after the `=`. For instance, for K70 RGB + M65 RGB: `usbhid.quirks=0x1B1C:0x1B13:0x20000408,0x1B1C:0x1B12:0x20000408`

If it still doesn't work, try replacing `0x20000408` with `0x4`. Note that this will cause the kernel driver to ignore the device(s) completely, so you need to ensure ckb-daemon is running at boot or else you'll have no input. This will not work if you are using full-disk encryption.

If you see **GLib** critical errors like "' GLib-GObject-CRITICAL **: g_type_add_interface_static: assertion 'G_TYPE_IS_INSTANTIATABLE (instance_type)' failed "' read `this Arch Linux thread` and try different combinations from it. If it doesn't help, you might want get support from your distribution community and tell them you cannot solve the problem in this thread.

If you're using **Unity** and the tray icon doesn't appear correctly, run `sudo apt-get install libappindicator-dev`. Then reinstall ckb.

Fedora 26 Color Changer Freeze Fix

If you're running Fedora 26, a working solution for the color changer freezing issue is to install qt5ct `dnf install qt5ct` then modify your /etc/environment file to contain the line `QT_QPA_PLATFORMTHEME=qt5ct`

OS X/macOS

- ∗∗"ckb.pkg" can't be opened because it is from an unidentified developer∗∗ Right-click (control-click) on ckb.pkg and select Open. This new dialog box will give you the option to open anyway, without changing your system preferences.
- **Modifier keys (Shift, Ctrl, etc.) are not rebound correctly** ckb does not recognize modifier keys rebound from System Preferences. You can rebind them again within the application.
- ∗∗∼ key prints `§±`∗∗ Check your keyboard layout on ckb's Settings screen. Choose the layout that matches your physical keyboard.
- **Compile problems** Can usually be resolved by rebooting your computer and/or reinstalling Qt. Make sure that Xcode works on its own. If a compile fails, delete the `ckb-master` directory as well as any automatically generated `build-ckb` folders and try again from a new download.
- **Scroll wheel does not scroll** As of #c3474d2 it's now possible to **disable scroll acceleration** from the GUI. You can access it under "OSX tweaks" in the "More settings" screen. Once disabled, the scroll wheel should behave consistently.

General

Please ensure your keyboard firmware is up to date. If you've just bought the keyboard, connect it to a Windows computer first and update the firmware from Corsair's official utility.

Before reporting an issue, connect your keyboard to a Windows computer and see if the problem still occurs. If it does, contact Corsair. Additionally, please check the Corsair user forums to see if your issue has been reported by other users. If so, try their solutions first.

Common issues:

- **Problem:** ckb says "No devices connected" or "Driver inactive"
- **Solution:** Try rebooting the computer and/or reinstalling ckb. Try removing the keyboard and plugging it back in. If the error doesn't go away, try the following:
- **Problem:** Keyboard doesn't work in BIOS, doesn't work at boot
- **Solution:** Some BIOSes have trouble communicating with the keyboard. They may prevent the keyboard from working correctly in the operating system as well. First, try booting the OS *without* the keyboard attached, and plug the keyboard in after logging in. If the keyboard works after the computer is running but does not work at boot, you may need to use the keyboard's BIOS mode option.
- BIOS mode can be activated using the poll rate switch at the back of the keyboard. Slide it all the way to the position marked "BIOS". You should see the scroll lock light blinking to indicate that it is on. (Note: Unfortunately, this has its own problems - see Known Issues. You may need to activate BIOS mode when booting the computer and deactivate it after logging in).
- **Problem:** Keyboard isn't detected when plugged in, even if driver is already running
- **Solution:** Try moving to a different USB port. Be sure to follow `Corsair's USB connection requirements`. Note that the keyboard does not work with some USB3 controllers - if you have problems with USB3 ports, try USB2 instead. If you have any USB hubs on hand, try those as well. You may also have success sliding the poll switch back and forth a few times.

Reporting issues

If you have a problem that you can't solve (and it isn't mentioned in the Known Issues section below), you can report it on the GitHub issue tracker. Before opening a new issue, please check to see if someone else has reported your problem already - if so, feel free to leave a comment there.

Known issues

- Using the keyboard in BIOS mode prevents the media keys (including mute and volume wheel), as well as the K95's G-keys from working. This is a hardware limitation.
- The tray icon doesn't appear in some desktop environments. This is a known Qt bug. If you can't see the icon, reopen ckb to bring the window back.
- When starting the driver manually, the Terminal window sometimes gets spammed with enter keys. You can stop it by unplugging and replugging the keyboard or by moving the poll rate switch.
- When stopping the driver manually, the keyboard sometimes stops working completely. You can reconnect it by moving the poll rate switch.
- On newer versions of macOS (i.e. 10.12 and up) CMD/Shift+select does not work, yet. Stopping the daemon and GUI for `ckb` will fix this issue temporarily.

Contributing

You can contribute to the project by opening a pull request. It's best if you base your changes off of the `testing` branch as opposed to the `master`, because the pull request will be merged there first. If you'd like to contribute but don't know what you can do, take a look at the issue tracker and see if any features/problems are still unresolved. Feel free to ask if you'd like some ideas.

Contact us

There are multiple ways you can get in touch with us:

- join `ckb-next` mailing list
- open a GitHub Issue
- hop on `#ckb-next` on Freenode to chat

What happened to the original ckb

Due to time restrictions, the original author of **ckb** ccMSC hasn't been able to further develop the software. So the community around it decided to take the project over and continue its development. That's how **ckb-next** was created. Currently it's not rock solid and not very easy to set up on newer systems but we are actively working on this. Nevertheless the project already incorporates a notable amount of fixes and patches in comparison to the original ckb.

Chapter 2

Building ckb

Linux

You can build the project by running `./qmake-auto && make` in a Terminal inside the ckb-master directory. The binaries will be placed in a new `bin` directory assuming they compile successfully. If you get a `No suitable qmake found` error, make sure Qt5 is installed and up to date. You may have to invoke qmake manually, then run `make` on its own. If you have Qt Creator installed, you can open `ckb.pro` (when asked to configure the project, make sure "Desktop" is checked) and use `Build > Build Project "ckb" (Ctrl+B)` to build the application instead.

Running as a service:

First copy the binary and the service files to their system directories:

- Upstart (Ubuntu, prior to 15.04): `sudo cp -R bin/* /usr/bin && sudo cp service/upstart/ckb-daemon.conf /etc/init`
- Systemd (Ubuntu 15.04 and later): `sudo cp -R bin/* /usr/bin && sudo cp service/systemd/ckb-daemon.service /usr/lib/systemd/system`
- OpenRC: `sudo cp -R bin/* /usr/bin && sudo cp service/openrc/ckb-daemon /etc/init.d/`

To launch the driver and enable it at start-up:

- Upstart: `sudo service ckb-daemon start`
- Systemd: `sudo systemctl start ckb-daemon && sudo systemctl enable ckb-daemon`
- OpenRC: `sudo rc-service ckb-daemon start && sudo rc-update add ckb-daemon default`

Open the `bin` directory and double-click on `ckb` to launch the user interface. If you want to run it at login, add `ckb --background` to your Startup Applications.

Running manually:

Open the `bin` directory in a Terminal and run `sudo ./ckb-daemon` to start the driver. To start the user interface, run `./ckb`. Running the driver manually may be useful for testing/debugging purposes, but you must leave the terminal window open and you'll have to re-run it at every reboot, so installing it as a service is the best long-term solution.

OSX

Open ckb.pro in Qt Creator. You should be prompted to configure the project (make sure the "Desktop" configuration is selected and not iOS). Once it's finished loading, press `Cmd+B` or select `Build > Build Project "ckb"` from the menu bar. When it's done, you should see a newly-created `ckb.app` in the project directory. Exit Qt Creator.

Alternatively, open a Terminal in the ckb-master directory and run `./qmake-auto && make`. It will detect Qt automatically if you installed it to one of the standard locations. You should see a newly created `ckb.app` if the build is successful.

Running as a service:

Copy `ckb.app` to your Applications folder. Copy the file `'service/launchd/com.ckb.daemon.plist'` to your computer's `/Library/LaunchDaemons` folder (you can get to it by pressing `Cmd+Shift+G` in Finder and typing the location). Then open a Terminal and run the following commands to launch the driver:

"' sudo chown root:wheel /Library/LaunchDaemons/com.ckb.daemon.plist sudo chmod 0700 /Library/LaunchDaemons/com.ckb.daemon.plist sudo launchctl load /Library/LaunchDaemons/com.ckb.daemon.plist "'

After you're done, open `ckb.app` to launch the user interface.

Running manually:

Open a Terminal in the ckb directory and run `sudo ckb.app/Contents/Resources/ckb-daemon` to start the driver. Open `ckb.app` to start the user interface. Note that you must leave the terminal window open and must re-launch the driver at every boot if you choose this; installing as a service is the better long term solution.

Chapter 3

cbk Improvements Roadmap

Short term plan

- merge existing PR submitted to original ckb repo
- Contact other developers interested in collaboration on a new and improved version of ckb
- Figure out the issues relating to MacOS Sierra and other version
- Device support:
  - Determine which will need support other than just USB id additions
- Address existing bugs. Not help requests.

Chapter 4

DAEMON

The daemon provides devices at `/dev/input/ckb*`, where `*` is the device number, starting at 1. Up to 9 devices may be connected at once and controlled independently. The daemon additionally provides `/dev/input/ckb0`, which stores driver information.

Mac note: The devices on OSX are located at `/var/run/ckb*` and not `/dev/input/ckb*`. So wherever you see `/dev/input` in this document, replace it with `/var/run`.

`/dev/input/ckb0` contains the following files:

- `connected`: A list of all connected devices, one per line. Each line contains a device path followed by the device's serial number and its description.
- `pid`: The process identifier of the daemon.
- `version`: The daemon version.

Other `ckb*` devices contain the following:

- `cmd`: Keyboard controller.
- `notify0`: Keyboard- or mouse notifications.
- `notify1`: Keyboard- or mouse notifications, used for macro recording.
- `features`: Device features.
- `fwversion`: Device firmware version (not present on all devices).
- `model`: Device description/model.
- `pollrate`: Poll rate in milliseconds (not present on all devices).
- `productid`: Contains the USB productID of the hardware
- `serial`: Device serial number. `model` and `serial` will match the info found in `ckb0/connected`

Commands

The `/dev/input/ckb*/cmd` nodes accept input in the form of text commands. They are normally accessible to all users on the system (see Security section). Commands should be given in the following format: `[mode <n>] command1 [paramter1] [command2] [parameter2] [command3] [parameter3] ...`

In a terminal shell, you can do this like `echo mycommand > /dev/input/ckb1/cmd`. Programmatically, you can open and write them as regular files. When programming, you must append a newline character and flush the output before your command(s) will actually be read.

The mode parameter is used to group settings. Most (but not all) settings are mode-specific; that is, changing mode 1 will not affect mode 2. By default, all commands affect the current mode. Use mode <n> switch to change the current mode.

When plugged in, all devices start in hardware-controlled mode (also known as idle mode) and will not respond to commands. Before issuing any other commands, write active to the command node, like echo active > /dev/input/ckb1/cmd. To put the device back into hardware mode, issue the idle command.

Features

The features node describes features supported by the device, which may not be present on all devices. The first two words in the features node are always <vendor> <model>, like corsair k70. After that, any of the following features may appear:

- adjrate: Device supports adjustable poll rate.
- bind: Device supports key rebinding.
- fwupdate: Device supports firmware updates.
- fwversion: Device has a detectable firmware version (stored in the fwversion node).
- notify: Device supports key notifications.
- pollrate: Device has a detectable poll rate (stored in the pollrate node).
- rgb: Device supports RGB lighting.

Keyboard layout

The driver has no concept of keyboard layouts; all keys are referred to by their English names regardless of the underlying hardware. This means that, for instance, in an AZERTY layout the q key in ckb-daemon corresponds to A on the physical keyboard. Note that on UK/european (ISO) layouts, the backslash key (beside left shift) is called bslash_iso, while bslash refers to the backslash on the US keyboard. The key next to Enter on the ISO keyboard is known as hash. See src/ckb-daemon/keymap.c for the full table of supported keys.

For technical reasons, the OSX driver may swap the bslash_iso and grave keys if the keyboard layout is not set correctly. To compensate for this, write layout iso or layout ansi to the command node.

Poll rate

A device's current poll rate can be read from its pollrate node, assuming it has one. Keyboards have a hardware switch to control poll rate and cannot be adjusted via software. However, mice have a software-controlled poll rate. You can change it by issuing pollrate <interval> to the command node, where interval is the time in milliseconds. Valid poll rates are 1, 2, 4, and 8.

Profiles and modes

Each mode has its own independent binding and lighting setup. When the daemon starts or a keyboard is plugged in, the profile will be loaded from the hardware. By default, all commands will update the currently selected mode. The mode <n> command may be used to change the settings for a different mode. Up to 6 modes are available. Each keyboard has one profile, which may be given a name. Mode 1 may be saved to the device hardware, or modes 1-3 in the case of the K95. Modes 4 through 6 are software-only. Profile management commands are as follows:

- profilename <name> sets the profile's name. The name must be written without spaces; to add a space, use %20.

- `name <name>` sets the current mode's name. Use `mode <n> name <name>` to set a different mode's name.
- `profileid <guid> [<modification>]` sets a profile's ID. The GUID must be written in registry format, like `{12345678-ABCD-EF01-2345-6789ABCDEF01}`. The optional modification number must be written with 8 hex digits, like `ABCDEF01`.
- `id <guid> [<modification>]` sets a mode's ID.
- `mode <n> switch` switches the keyboard to mode N. If the mode does not exist, it will be created with a blank ID, black lighting, and default bindings.
- `hwload` loads the RGB profile from the hardware. Key bindings and non-hardware RGB modes are unaffected.
- `hwsave` saves the RGB profile to the hardware.
- `erase` erases the current mode, resetting its lighting and bindings. Use `mode <n> erase` to erase a different mode.
- `eraseprofile` erases the entire profile, deleting its name, ID, and all of its modes.

Examples:

- `profilename My%20Profile mode 1 name Mode%201 mode 2 name Mode%202 mode 3 name Mode%203` will name the profile "My Profile" and name modes 1-3 "Mode 1", "Mode 2", and "Mode 3".
- `eraseprofile hwload` resets the entire profile to its hardware settings.

LED commands

The backlighting is controlled by the `rgb` commands.

- `rgb <RRGGBB>` sets the entire keyboard to the color specified by the hex constant RRGGBB.
- `rgb <key>:<RRGGBB>` sets the specified key to the specified hex color.

Examples:

- `rgb ffffff` makes the whole keyboard white.
- `rgb 000000` makes the whole keyboard black.
- `rgb esc:ff0000` sets the Esc key red but leaves the rest of the keyboard unchanged.

Multiple keys may be changed to one color when separated with commas, for instance:

- `rgb w,a,s,d:0000ff` sets the WASD keys to blue.

Additionally, multiple commands may be combined into one, for instance:

- `rgb ffffff esc:ff0000 w,a,s,d:0000ff` sets the Esc key red, the WASD keys blue, and the rest of the keyboard white (note the lack of a key name before `ffffff`, implying the whole keyboard is to be set).

By default, the controller runs at 30 FPS, meaning that attempts to animate the LEDs faster than that will be ignored. If you wish to change it, send the command `fps <n>`. The maximum frame rate is 60.

For devices running in 512-color mode, color dithering can be enabled by sending the command `dither 1`. The command `dither 0` disables dithering.

Indicators

The indicator LEDs (Num Lock, Caps Lock, Scroll Lock) are controlled with the `i` commands.

- `ioff <led>` turns an indicator off permanently. Valid LED names are `num`, `caps`, and `scroll`.
- `ion <led>` turns an indicator on permanently.
- `iauto <led>` turns an indicator off or on automatically (default behavior).

Binding keys

Keys may be rebound through use of the `bind` commands. Binding is a 1-to-1 operation that translates one keypress to a different keypress regardless of circumstance.

- `bind <key1>:<key2>` remaps key1 to key2.
- `unbind <key>` unbinds a key, causing it to lose all function.
- `rebind <key>` resets a key, returning it to its default binding.

Examples:

- `bind g1:esc` makes G1 become an alternate Esc key (the actual Esc key is not changed).
- `bind caps:tab tab:caps` switches the functions of the Tab and Caps Lock keys.
- `unbind lwin rwin` disables both Windows keys, even without using the keyboard's Windows Lock function.
- `rebind all` resets the whole keyboard to its default bindings.

Key macros

Macros are a more advanced form of key binding, controlled with the `macro` command.

- `macro <keys>:<command>` binds a key combination to a command, where the command is a series of key presses. To combine keys, separate them with `+`; for instance, `lctrl+a` binds a macro to (left) Ctrl+A. In the command field, enter `+<key>` to trigger a key down or `-<key>` to trigger a key up. To simulate a key press, use `+<key>,-<key>`.
- `macro <keys>:clear` clears commands associated with a key combination. Only one macro may be assigned per combination; assigning a second one will overwrite the first.
- `macro clear` clears all macros.

Examples:

- `macro g1:+lctrl,+a,-a,-lctrl` triggers a Ctrl+A when G1 is pressed.
- `macro g2+g3:+lalt,+f4,-f4,-lalt` triggers an Alt+F4 when G2 and G3 are pressed simultaneously.

Assigning a macro to a key will cause its binding to be ignored; for instance, `macro a:+b,-b` will cause A to generate a B character regardless of its binding. However, `macro lctrl+a:+b,-b` will cause A to generate a B only when Ctrl is also held down.

Macro playback delay

There are two types of playback delay that can be set with macros; global and local. Setting a *global delay* value introduces a time delay between events during macro execution or playback. *Local delay* allows setting the delay after an individual event, overriding the global delay value for that event. Thus global delay can be used to set the overall playback speed of macros and local delays can be used to tune individual events within a macro.

All delay values are specified in microseconds (us) and are positive values from `0` to `UINT_MAX - 1`. This means delays range from 0 to just over 1 hour (4,294,967,294us, 4,294 seconds, 71 minutes, or 1.19 hours). A value of zero (0) represents no delay between actions.

Global macro delay (default delay)

Global delay allows macro playback speed to be changed. It sets the time between (actually after) each recorded macro event. If global delay is set to 1 microsecond then a 1 ms delay will follow each individual macro event when the macro is triggered.

The *global delay* is set with the ckb-daemon's existing (in testing branch) `delay` command followed by an unsigned integer representing the number of microseconds to wait after each macro action and before the next.

Global delay can also be set to `on` which maintains backwards compatibility with the current development of `ckb-daemon` for long macro playback. That is, setting the global delay to `on` introduces a 30us and a 100us delay based on the macro's length during playback.

NOTE: This setting also introduces a delay after the last macro action. This functionality exists in the current testing branch and was left as-is. It is still to be determined if this is a bug or a feature.

Examples:

- `delay 1000` sets a 1,000us delay between action playback.
- `delay on` sets long macro delay; 30us for actions between 20 and 200, 100us for actions > 200.
- `delay off` sets no delay (same as 0).
- `delay 0` sets no delay (same as off).
- `delay spearmint-potato` is invalid input, sets no delay (same as off).

Local macro delay (keystroke delay)

Local Delay allows each macro action to have a post-action delay associated with it. This allows a macro to vary it's playback speed for each event. If no local delay is specified for a macro action, then the global `delay` (above) is used. All delay values are in microsecods (us) as with the global delay setting.

Examples:

- `macro g5:+d,-d,+e=5000,-e,+l,-l=10000,+a,-a,+y,-y=1000000,+enter,-enter` define a macro for `g5` with a 5,000us delay between the `e` down and `e` up actions. A 1,000us delay between `l` up and `a` down, a delay of one second (1,000,000us) after `y` up and before `enter`, and the global delay for all other actions.
- `macro g5:+d,-d=0` use default delay between `d` down and `d` up and no delay (0us) after `d` up. This removes the noted feature/bug (above) where the last action has a trailing delay associated with it.

DPI and mouse settings

DPI settings are stored in a bank. They are controlled with the `dpi` command.

- `dpi <stage>:<x>,<y>` sets the DPI for a given `stage` to `x` by `y`. Valid stages are `0` through `5`. In hardware, `1` is the first (lowest) stage and `5` is the highest. Stage `0` is used for Sniper mode.
- `dpi <stage>:<xy>` sets both X and Y.

- `dpi <stage>:off` disables a DPI stage.
- `dpisel <stage>` sets the current stage selection.

In order to change the mouse's current DPI, first update one of the stages with the value you want, then select that stage. For instance:

- `dpi 1:1000 dpisel 1` sets the current DPI to 1000x1000.

Additional mouse settings:

- `lift <height>` sets the lift height, from `1` (lowest) to `5` (highest)
- `snap <on|off>` enables or disables Angle Snap.

Notifications

The keyboard can be configured to generate user-readable notifications on keypress events. These are controlled with the `notify` commands. In order to see events, read from `/dev/input/ckb*/notify0`. In a terminal, you can do this like `cat /dev/input/ckb1/notify0`. Programmatically, you can open it for reading like a regular file.

Note that the file can only reliably be read by one application: if you try to open it in two different programs, they may both fail to get data. Data will be buffered as long as no programs are reading, so you will receive all unread notifications as soon as you open the file. If you'd like to read notifications from two separate applications, send the command `notifyon <n>` to the keyboard you wish to receive notifications from, where N is a number between 1 and 9. If `/dev/input/ckb*/notify<n>` does not already exist, it will be created, and you can read notifications from there without disrupting any other program. To close a notification node, send `notifyoff <n>`.

`notify0` is always open and will not be affected by `notifyon`/`notifyoff` commands. By default, all notifications are printed to `notify0`. To print output to a different node, prefix your command with `@<node>`.

Notifications are printed with one notification per line. Commands are as follows:

- `notify <key>:on` or simply `notify <key>` enables notifications for a key. Each key will generate two notifications: `key +<key>` when the key is pressed, and `key -<key>` when it is released.
- `notify <key>:off` turns notifications off for a key.

Examples:

- `notify w a s d` sends notifications whenever W, A, S, or D is pressed.
- `notify g1 g2 g3 g4 g5 g6 g7 g8 g9 g10 g11 g12 g13 g14 g15 g16 g17 g18 mr m1 m2 m3 light lock` prints a notification whenever a non-standard key is pressed.
- `notify all:off` turns all key notifications off.
- `@5 notify esc` prints Esc key notifications to `notify5`.

Note: Key notifications are *not* affected by bindings. For instance, if you run `echo bind a:b notify a > /dev/input/ckb1/cmd` and then press the A key, the notifications will read `key +a key -a`, despite the fact that the character printed on screen will be `b`. Likewise, unbinding a key or assigning a macro to a key does not affect the notifications.

Indicator notifications

You can also choose to receive notifications for the indicator LEDs by using the `inotify` command. For instance, `inotify caps:on` or simply `inotify caps` will print notifications whenever the Caps Lock LED is toggled. The notifications will read `i +caps` when the light is turned on and `i -caps` when it is turned off. It is also possible to toggle all indicators at once using `inotify all` or `inotify all:off`.

Like key notifications, indicator notifications are not affected by bindings, nor by the `ion`, `ioff`, or `iauto` commands. The notifications will reflect the state of the LEDs as seen be the event device.

Getting parameters

Parameters can be retrieved using the `get` command. The data will be sent out as a notification. Generally, the syntax to get the data associated with a command is `get :<command>` (note the colon), and the associated data will be returned in the form of `<command> <data>`. The following data may be gotten:

- `get :mode` returns the current mode in the form of a `switch` command. (Note: Do not use this in a line containing a `mode` command or it will return the mode that you selected, rather than the keyboard's current mode.)
- `get :name` returns the current mode's name in the form of `mode <n> name <name>`. To see the name of another mode, use `mode <n> get :name`. The name is URL-encoded; spaces are written as %20. The name may be truncated, so `name <some long string> get :name` may return something shorter than what was entered.
- `get :profilename` returns the profile's name, in the form of `profilename <name>`. As above, it is URL-encoded and may be truncated.
- `get :hwname` and `get :hwprofilename` return the same thing except taken from the current hardware profile instead of the in-memory profile. The output is identical but will read `hwname` instead of `name` and `hwprofilename` instead of `profilename`.
- `get :id` returns the current mode's ID and modification number in the form of `mode <n> id <guid> <modification>`.
- `get :profileid` returns the current profile's ID and modification number in the form of `profileid <guid> <modification>`.
- `get :hwid` and `get :hwprofileid` return the same thing except from the current hardware profile/mode. As before, the ouput will be the same but with `hwid` and `hwprofileid` instead of `id` and `profileid`.
- `get :rgb` returns an `rgb` command equivalent to the current RGB state.
- `get :hwrgb` does the same thing, but retrieves the colors currently stored in the hardware profile. The output will say `hwrgb` instead of `rgb`.
- `get :dpi` returns a `dpi` command equivalent to the current DPI bank.
- `get :dpisel` returns a `dpisel` command for the currently-selected DPI stage.
- `get :lift` returns a `lift` command for the current lift height.
- `get :snap` returns the current angle snap status.
- `get :hwdpi`, `get :hwdpisel`, `get :hwlift`, and `get :hwsnap` return the same properties, but for the current hardware profile.
- `get :keys` and `get :i` return the current keypress status and indicator status, respectively. They will indicate all currently pressed keys and all currently active indicators, like `key +enter` and `i +num`.

Like `notify`, you must prefix your command with `@<node>` to get data printed to a node other than `notify0`.

Firmware updates

WARNING: Improper use of `fwupdate` may brick your device; use this command *at your own risk*. I accept no responsibility for broken keyboards.

The latest firmware versions and their URLs can be found in the `FIRMWARE` document. To update your keyboard's firmware, first extract the contents of the zip file and then issue the command `fwupdate /path/to/fw/file.bin` to the keyboard you wish to update. The path name must be absolute and must not include spaces. If it succeeded, you should see `fwupdate <path> ok` logged to the keyboard's notification node and then the device will disconnect and reconnect. If you see `fwupdate <path> invalid` it means that the firmware file was not valid for the device; more info may be available in the daemon's `stdout`. If you see `fwupdate <path> fail` it means that the file was valid but the update failed at a hardware level. The keyboard may disconnect/reconnect anyway or it may remain in operation.

When the device reconnects you should see the new firmware version in its `fwversion` node; if you see `0000` instead it means that the keyboard did not update successfully and will need another `fwupdate` command in order to function again. If the update fails repeatedly, try connecting the keyboard to a Windows PC and using the official firmware update in CUE.

Restart

Because sometimes the communication between the daemon and the keyboard is corrupted after resuming from standby or suspend, a restart function is implemented. It first calls the quit() function, then it calls main() again with the original parameter list.

There are two ways to restart the daemon:

- send the string "restart some-description-as-one-word" to the cmd-pipe (normally /dev/input/ckb1/cmd or /dev/input/ckb2/cmd, depending on what device gets which ID.
- send SIGUSR1 to the daemon process (as root).

Later on, there may be a user interface in the client for the first method.

Security

By default, all of the `ckb*` nodes may be accessed by any user. For most single-user systems this should not present any security issues, since only one person will have access to the computer anyway. However, if you'd like to restrict the users that can write to the `cmd` nodes or read from the `notify` nodes, you can specify the `--gid=<group>` option at start up. For instance, on most systems you could run `ckb-daemon --gid=1000` to make them accessible only by the system's primary user. `ckb-daemon` must still be run as root, regardless of which `gid` you specify. The `gid` option may be set only at startup and cannot be changed while the daemon is running.

The daemon additionally supports a `--nonotify` option to disable key notifications, to prevent unauthorized programs from logging key input. Note that this will interfere with some of `ckb`'s abilities. It is also highly unlikely to increase security unless you are using the program in a stripped down terminal environment without Xorg. For most use cases there are many other (more likely) ways that a keylogger program could compromise your system. Nevertheless, the option is provided for the sake of paranoia. If you'd like to disable key rebinding as well, launch the daemon with `--nobind`. `--nobind` implies `--nonotify`, so notifications will also be disabled. As with `--gid`, these options must be set at startup and cannot be changed while the daemon is running.

Chapter 5

Todo List

Global _usbsend (usbdevice ∗kb, const uchar ∗messages, int count, const char ∗file, int line)

A lot of different conditions are combined in this code. Don't think, it is good in every combination...

Check whether this is the same in the macOS variant. It is not dramatic, but if errors occur, it can certainly irritate the devices completely if they receive incomplete data streams. Do we have errors with the messages "Wrote YY bytes (expected 64)" in the system logs? If not, we do not need to look any further.

Global closeusb (usbdevice ∗kb)

What is not yet comprehensible is the call to updateconnected() BEFORE os_closeusb(). Should that be in the other sequence? Or is updateconnected() not displaying the connected usb devices, but the representation which uinput devices are loaded? Questions about questions ...

Global devmain (usbdevice ∗kb)

Hope to find the need for dmutex usage later.

Should this function be declared as pthread_t∗ function, because of the defintion of pthread-create? But void∗ works also...

readcmd() gets a **line**, not **lines**. Have a look on that later.

Is the condition IS_CONNECTED valid? What functions change the condititon for the macro?

Global get_vtable (short vendor, short product)

Is the last point really a good decision and always correct?

Global inputupdate_keys (usbdevice ∗kb)

If we want to get all keys typed while a macro is played, add the code for it here.

Global Kb::Kb (QObject ∗parent, const QString &path)

Is adding notify2 to the notifypaths neccessary?

Global Kb::∼Kb ()

I don't think, that notifypaths is used somewhere. So why do we have it? If we do not need it, searching for an ununsed notify channel can easy be refactored to a private member function.

Global macro_pt_enqueue ()

find a better exit strategy if no more mem available.

Global os_inputmain (void ∗context)

This function is a collection of many tasks. It should be divided into several sub-functions for the sake of greater convenience:

Global os_resetusb (usbdevice ∗kb, const char ∗file, int line)

it seems that no one wants to try the reset again. But I'v seen it somewhere...

Global os_setupusb (usbdevice ∗kb)

in these modules a pullrequest is outstanding

Global os_usbsend (usbdevice ∗kb, const uchar ∗out_msg, int is_recv, const char ∗file, int line)

Since the handling of endpoints has already led to problems elsewhere, this implementation is extremely hardware-dependent and critical!

Eg. the new keyboard K95PLATINUMRGB has a version number significantly less than 2.0 - will it run with this implementation?

Global product_str (short product)

There are macros defined in usb.h to detect all the combinations below. the only difference is the parameter: The macros need the *kb∗*, product_str() needs the *product ID*

Global RebindWidget::applyChanges (const QStringList &keys, bool doUnbind)

There is still a bug in the state machine: If you record a macro in asTyped-mode, switch to another mode and change the vontent of the pteMacroBox manually, then the changes are not saved in the timing buffer. But anyhow, let's do more relevant things...

Global RebindWidget::on_btnClearMacro_clicked ()

I do not know what is the better solution with the delay-buttons in case of clicking clear: Reset the button to the default value or do not touch it? Not clear is ignored.

Global revertusb (usbdevice ∗kb)

Why is this useful? Are there problems seen with deactivating a device with older fw-version??? Why isn't this an error indicating reason and we return success (0)?

The return value of nk95cmd() is ignored (but sending the ioctl may produce an error and _nk95_cmd will indicate this), instead revertusb() returns success in any case.

Global udevthread

These two thread vasriables seem to be unused: usbtread, udevthread

Global udevthread

These two thread vasriables seem to be unused: usbtread, udevthread

Global usb_add_device (struct udev_device ∗dev)

So why the hell not a transformation between the string and the short presentation? Lets check if the string representation is used elsewhere.

Global usb_tryreset (usbdevice ∗kb)

Why does usb_tryreset() hide the information returned from resetusb()? Isn't it needed by the callers?

Global usbmain ()

Why isn't missing of uinput a fatal error?

lae. here the work has to go on...

Global usbmutex

We should have a look why this mutex is never used.

Chapter 6

Deprecated List

Class QuaZipFileInfo

Use QuaZipFileInfo64 instead. Not only it supports large files, but also more convenience methods as well.

Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui 41

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

_model . . . 1583
_readlines_ctx . . . 1257
AutoRun . . . 120
binding . . . 1504
ckb_gradient . . . 1649
ckb_key . . . 1649
ckb_runctx . . . 1649
CkbSettings . . . 124
ColorMap . . . 151
curfile64_info . . . 1898
devcmd . . . 1224
devcmd.__unnamed__ . . . 160
dpiset . . . 1504
MPerfWidget::DpiUi . . . 781
file_in_zip64_read_info_s . . . 1827
KbFirmware::FW . . . 183
hwprofile . . . 1504
KPerfWidget::IndicatorUi . . . 705
JlCompress . . . 243
KbId . . . 1683
Key . . . 602
key . . . 1740
keyAnim . . . 1404
keyanim . . . 1411
keymacro . . . 1504
KeyMap . . . 636
KeyPatch . . . 1717
kiss_fft_cpx . . . 1625
kiss_fft_state . . . 1609
kiss_fftr_state . . . 1632
lighting . . . 1504
linkedlist_data_s . . . 1898
linkedlist_datablock_internal_s . . . 1898
macroaction . . . 1504
AnimScript::Param . . . 54
parameter . . . 1317
GradientDialog::Preset . . . 799
ptlist . . . 1317

QDialog . . . 801
AnimAddDialog . . . 48
AnimSettingDialog . . . 99
ExtraSettingsWidget . . . 175
FwUpgradeDialog . . . 184
GradientDialog . . . 214
KbProfileDialog . . . 556
LayoutDialog . . . 722
ModeSelectDialog . . . 757
QIODevice . . . 802
QuaGzipFile . . . 818
QuaZIODevice . . . 825
QuaZipFile . . . 883
QIODevice_descriptor . . . 803
QListWidget . . . 804
RListWidget . . . 1007
QMainWindow . . . 805
MainWindow . . . 737
QObject . . . 806
AnimScript . . . 54
CkbSettingsWriter . . . 139
KbAnim . . . 311
KbBind . . . 366
KbFirmware . . . 413
KbLight . . . 425
KbManager . . . 477
KbMode . . . 489
KbPerf . . . 503
KbProfile . . . 538
KeyAction . . . 605
QPushButton . . . 807
ColorButton . . . 141
GradientButton . . . 200
qt_meta_stringdata_AnimAddDialog_t . . . 1966
qt_meta_stringdata_AnimScript_t . . . 1969
qt_meta_stringdata_AnimSettingDialog_t . . . 1972
qt_meta_stringdata_CkbSettingsWriter_t . . . 1975
qt_meta_stringdata_ColorButton_t . . . 1977
qt_meta_stringdata_ExtraSettingsWidget_t . . . 1979
qt_meta_stringdata_FwUpgradeDialog_t . . . 1982
qt_meta_stringdata_GradientButton_t . . . 1985
qt_meta_stringdata_GradientDialog_t . . . 1988
qt_meta_stringdata_GradientDialogWidget_t . . . 1991
qt_meta_stringdata_Kb_t . . . 1993
qt_meta_stringdata_KbAnim_t . . . 1996
qt_meta_stringdata_KbAnimWidget_t . . . 1998
qt_meta_stringdata_KbBind_t . . . 2001
qt_meta_stringdata_KbBindWidget_t . . . 2003
qt_meta_stringdata_KbFirmware_t . . . 2005
qt_meta_stringdata_KbLight_t . . . 2008
qt_meta_stringdata_KbLightWidget_t . . . 2011
qt_meta_stringdata_KbManager_t . . . 2014
qt_meta_stringdata_KbMode_t . . . 2016
qt_meta_stringdata_KbPerf_t . . . 2018
qt_meta_stringdata_KbProfile_t . . . 2021
qt_meta_stringdata_KbProfileDialog_t . . . 2023
qt_meta_stringdata_KbWidget_t . . . 2027

qt_meta_stringdata_KeyAction_t . . . 2029
qt_meta_stringdata_KeyWidget_t . . . 2031
qt_meta_stringdata_KPerfWidget_t . . . 2033
qt_meta_stringdata_LayoutDialog_t . . . 2036
qt_meta_stringdata_MacroReader_t . . . 2039
qt_meta_stringdata_MacroReaderThread_t . . . 2039
qt_meta_stringdata_MainWindow_t . . . 2043
qt_meta_stringdata_ModeSelectDialog_t . . . 2045
qt_meta_stringdata_MPerfWidget_t . . . 2047
qt_meta_stringdata_QuaGzipFile_t . . . 2050
qt_meta_stringdata_QuaZIODevice_t . . . 2052
qt_meta_stringdata_QuaZipFile_t . . . 2054
qt_meta_stringdata_RebindWidget_t . . . 2056
qt_meta_stringdata_RListWidget_t . . . 2059
qt_meta_stringdata_SettingsWidget_t . . . 2061
QThread . . . 808
  Kb . . . 261
  MacroReader . . . 725
  MacroReaderThread . . . 729
QuaChecksum32 . . . 812
  QuaAdler32 . . . 809
  QuaCrc32 . . . 815
QuaZip . . . 833
QuaZipDir . . . 865
QuaZipFileInfo . . . 906
QuaZipFileInfo64 . . . 909
QuaZipFilePrivate . . . 917
QuaZipNewInfo . . . 922
QuaZipPrivate . . . 930
QWidget . . . 939
  GradientDialogWidget . . . 227
  KbAnimWidget . . . 348
  KbBindWidget . . . 403
  KbLightWidget . . . 464
  KbWidget . . . 570
  KeyWidget . . . 661
  KPerfWidget . . . 705
  MPerfWidget . . . 781
  RebindWidget . . . 971
  SettingsWidget . . . 1012
rgb . . . 1433
SGroup . . . 1036
tm_unz_s . . . 1878
tm_zip_s . . . 1954
Ui_AnimAddDialog . . . 1039
  Ui::AnimAddDialog . . . 43
Ui_AnimSettingDialog . . . 1043
  Ui::AnimSettingDialog . . . 86
Ui_ExtraSettingsWidget . . . 1056
  Ui::ExtraSettingsWidget . . . 162
Ui_FwUpgradeDialog . . . 1068
  Ui::FwUpgradeDialog . . . 193
Ui_GradientDialog . . . 1074
  Ui::GradientDialog . . . 206
Ui_KbAnimWidget . . . 1082
  Ui::KbAnimWidget . . . 338

Ui_KbBindWidget . . . 1091
Ui::KbBindWidget . . . 396
Ui_KbLightWidget . . . 1097
Ui::KbLightWidget . . . 455
Ui_KbProfileDialog . . . 1105
Ui::KbProfileDialog . . . 565
Ui_KbWidget . . . 1109
Ui::KbWidget . . . 592
Ui_KPerfWidget . . . 1119
Ui::KPerfWidget . . . 684
Ui_LayoutDialog . . . 1138
Ui::LayoutDialog . . . 717
Ui_MainWindow . . . 1143
Ui::MainWindow . . . 733
Ui_ModeSelectDialog . . . 1147
Ui::ModeSelectDialog . . . 753
Ui_MPerfWidget . . . 1151
Ui::MPerfWidget . . . 761
Ui_RebindWidget . . . 1171
Ui::RebindWidget . . . 940
Ui_SettingsWidget . . . 1201
Ui::SettingsWidget . . . 1024
unz64_file_pos_s . . . 1878
unz64_s . . . 1827
unz_file_info64_internal_s . . . 1827
unz_file_info64_s . . . 1878
unz_file_info_s . . . 1878
unz_file_pos_s . . . 1878
unz_global_info64_s . . . 1878
unz_global_info_s . . . 1878
usbdevice . . . 1504
UsbId . . . 1213
usbid . . . 1504
usbinput . . . 1504
usbmode . . . 1504
usbprofile . . . 1504
zip64_internal . . . 1898
zip_fileinfo . . . 1954
zlib_filefunc64_32_def_s . . . 1780
zlib_filefunc64_def_s . . . 1780
zlib_filefunc_def_s . . . 1780

Chapter 9

Data Structure Index

9.1 Data Structures

Here are the data structures with brief descriptions:

Ui::AnimAddDialog . . . 43
AnimAddDialog . . . 48
AnimScript . . . 54
Ui::AnimSettingDialog . . . 86
AnimSettingDialog . . . 99
AutoRun . . . 120
CkbSettings . . . 124
CkbSettingsWriter . . . 139
ColorButton . . . 141
ColorMap . . . 151
devcmd.__unnamed__ . . . 160
Ui::ExtraSettingsWidget . . . 162
ExtraSettingsWidget . . . 175
KbFirmware::FW . . . 183
FwUpgradeDialog . . . 184
Ui::FwUpgradeDialog . . . 193
GradientButton . . . 200
Ui::GradientDialog . . . 206
GradientDialog . . . 214
GradientDialogWidget . . . 227
JlCompress
    Utility class for typical operations . . . 243
Kb . . . 261
KbAnim . . . 311
Ui::KbAnimWidget . . . 338
KbAnimWidget . . . 348
KbBind . . . 366
Ui::KbBindWidget . . . 396
KbBindWidget . . . 403
KbFirmware . . . 413
KbLight . . . 425
Ui::KbLightWidget . . . 455
KbLightWidget . . . 464
KbManager . . . 477
KbMode . . . 489
KbPerf . . . 503
KbProfile . . . 538
KbProfileDialog . . . 556

Ui::KbProfileDialog . . . 565
KbWidget . . . 570
Ui::KbWidget . . . 592
Key . . . 602
KeyAction . . . 605
KeyMap . . . 636
KeyWidget . . . 661
Ui::KPerfWidget . . . 684
KPerfWidget . . . 705
Ui::LayoutDialog . . . 717
LayoutDialog . . . 722
MacroReader
Creates a worker thread object. It does a connect do delayed deletion of thread local variables in the case the worker thread terminates . . . 725
MacroReaderThread
Responsible for reading Macro Key Values. It is created as a separate thread (worker thread) for reading macro commands from an fresh opened notify channel. Standard notify channel for macro definitions is number 2 . . . 729
Ui::MainWindow . . . 733
MainWindow . . . 737
Ui::ModeSelectDialog . . . 753
ModeSelectDialog . . . 757
Ui::MPerfWidget . . . 761
MPerfWidget . . . 781
GradientDialog::Preset . . . 799
QDialog . . . 801
QIODevice . . . 802
QIODevice_descriptor . . . 803
QListWidget . . . 804
QMainWindow . . . 805
QObject . . . 806
QPushButton . . . 807
QThread . . . 808
QuaAdler32
Adler32 checksum . . . 809
QuaChecksum32
Checksum interface . . . 812
QuaCrc32
CRC32 checksum . . . 815
QuaGzipFile
GZIP file . . . 818
QuaZIODevice
A class to compress/decompress QIODevice . . . 825
QuaZip
ZIP archive . . . 833
QuaZipDir
Provides ZIP archive navigation . . . 865
QuaZipFile
A file inside ZIP archive . . . 883
QuaZipFileInfo
Information about a file inside archive . . . 906
QuaZipFileInfo64
Information about a file inside archive (with zip64 support) . . . 909
QuaZipFilePrivate
The implementation class for QuaZip . . . 917
QuaZipNewInfo
Information about a file to be created . . . 922

QuaZipPrivate
All the internal stuff for the QuaZip class . . . 930
QWidget . . . 939
Ui::RebindWidget . . . 940
RebindWidget . . . 971
RListWidget . . . 1007
SettingsWidget . . . 1012
Ui::SettingsWidget . . . 1024
SGroup . . . 1036
Ui_AnimAddDialog . . . 1039
Ui_AnimSettingDialog . . . 1043
Ui_ExtraSettingsWidget . . . 1056
Ui_FwUpgradeDialog . . . 1068
Ui_GradientDialog . . . 1074
Ui_KbAnimWidget . . . 1082
Ui_KbBindWidget . . . 1091
Ui_KbLightWidget . . . 1097
Ui_KbProfileDialog . . . 1105
Ui_KbWidget . . . 1109
Ui_KPerfWidget . . . 1119
Ui_LayoutDialog . . . 1138
Ui_MainWindow . . . 1143
Ui_ModeSelectDialog . . . 1147
Ui_MPerfWidget . . . 1151
Ui_RebindWidget . . . 1171
Ui_SettingsWidget . . . 1201
UsbId . . . 1213

Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

src/ckb-daemon/command.c . . . 1219
src/ckb-daemon/command.h . . . 1224
src/ckb-daemon/device.c . . . 1233
src/ckb-daemon/device.h . . . 1237
src/ckb-daemon/device_keyboard.c . . . 1247
src/ckb-daemon/device_mouse.c . . . 1252
src/ckb-daemon/device_vtable.c . . . 1255
src/ckb-daemon/devnode.c . . . 1257
src/ckb-daemon/devnode.h . . . 1271
src/ckb-daemon/dpi.c . . . 1279
src/ckb-daemon/dpi.h . . . 1285
src/ckb-daemon/extra_mac.c . . . 1291
src/ckb-daemon/firmware.c . . . 1291
src/ckb-daemon/firmware.h . . . 1296
src/ckb-daemon/includes.h . . . 1299
src/ckb-daemon/input.c . . . 1303
src/ckb-daemon/input.h . . . 1317
src/ckb-daemon/input_linux.c . . . 1328
src/ckb-daemon/input_mac.c . . . 1335
src/ckb-daemon/input_mac_mouse.c . . . 1335
src/ckb-daemon/keymap.c . . . 1335
src/ckb-daemon/keymap.h . . . 1740
src/ckb-daemon/keymap_mac.h . . . 1340
src/ckb-daemon/led.c . . . 1341
src/ckb-daemon/led.h . . . 1346
src/ckb-daemon/led_keyboard.c . . . 1358
src/ckb-daemon/led_mouse.c . . . 1392
src/ckb-daemon/main.c . . . 1396
src/ckb-daemon/notify.c . . . 1452
src/ckb-daemon/notify.h . . . 1461
src/ckb-daemon/os.h . . . 1466
src/ckb-daemon/profile.c . . . 1467
src/ckb-daemon/profile.h . . . 1484
src/ckb-daemon/profile_keyboard.c . . . 1499
src/ckb-daemon/profile_mouse.c . . . 1502
src/ckb-daemon/structures.h . . . 1504
src/ckb-daemon/usb.c . . . 1523

src/ckb-daemon/usb.h
Definitions for using USB interface . . . 1541
src/ckb-daemon/usb_linux.c . . . 1583
src/ckb-daemon/usb_mac.c . . . 1609
src/ckb-gradient/main.c . . . 1404
src/ckb-heat/main.c . . . 1411
src/ckb-mviz/_kiss_fft_guts.h . . . 1609
src/ckb-mviz/kiss_fft.c . . . 1614
src/ckb-mviz/kiss_fft.h . . . 1625
src/ckb-mviz/kiss_fftr.c . . . 1632
src/ckb-mviz/kiss_fftr.h . . . 1637
src/ckb-mviz/main.c . . . 1416
src/ckb-pinwheel/main.c . . . 1422
src/ckb-rain/main.c . . . 1427
src/ckb-random/main.c . . . 1433
src/ckb-ripple/main.c . . . 1438
src/ckb-wave/main.c . . . 1444
src/ckb/animadddialog.cpp . . . 1641
src/ckb/animadddialog.h . . . 1642
src/ckb/animscript.cpp . . . 1642
src/ckb/animscript.h . . . 1644
src/ckb/animsettingdialog.cpp . . . 1645
src/ckb/animsettingdialog.h . . . 1647
src/ckb/autorun.cpp . . . 1648
src/ckb/autorun.h . . . 1649
src/ckb/ckb-anim.h . . . 1649
src/ckb/ckbsettings.cpp . . . 1673
src/ckb/ckbsettings.h . . . 1676
src/ckb/ckbsettingswriter.cpp . . . 1676
src/ckb/ckbsettingswriter.h . . . 1678
src/ckb/colorbutton.cpp . . . 1678
src/ckb/colorbutton.h . . . 1679
src/ckb/colormap.cpp . . . 1680
src/ckb/colormap.h . . . 1681
src/ckb/extrasettingswidget.cpp . . . 1682
src/ckb/extrasettingswidget.h . . . 1683
src/ckb/fwupgradedialog.cpp . . . 1683
src/ckb/fwupgradedialog.h . . . 1686
src/ckb/gradientbutton.cpp . . . 1687
src/ckb/gradientbutton.h . . . 1688
src/ckb/gradientdialog.cpp . . . 1689
src/ckb/gradientdialog.h . . . 1689
src/ckb/gradientdialogwidget.cpp . . . 1690
src/ckb/gradientdialogwidget.h . . . 1690
src/ckb/kb.cpp . . . 1691
src/ckb/kb.h . . . 1692
src/ckb/kbanim.cpp . . . 1693
src/ckb/kbanim.h . . . 1695
src/ckb/kbanimwidget.cpp . . . 1696
src/ckb/kbanimwidget.h . . . 1696
src/ckb/kbbind.cpp . . . 1697
src/ckb/kbbind.h . . . 1698
src/ckb/kbbindwidget.cpp . . . 1699
src/ckb/kbbindwidget.h . . . 1699
src/ckb/kbfirmware.cpp . . . 1700
src/ckb/kbfirmware.h . . . 1701
src/ckb/kblight.cpp . . . 1702
src/ckb/kblight.h . . . 1705

src/ckb/kblightwidget.cpp . . . 1706
src/ckb/kblightwidget.h . . . 1706
src/ckb/kbmanager.cpp . . . 1707
src/ckb/kbmanager.h . . . 1708
src/ckb/kbmode.cpp . . . 1709
src/ckb/kbmode.h . . . 1709
src/ckb/kbperf.cpp . . . 1710
src/ckb/kbperf.h . . . 1710
src/ckb/kbprofile.cpp . . . 1711
src/ckb/kbprofile.h . . . 1712
src/ckb/kbprofiledialog.cpp . . . 1712
src/ckb/kbprofiledialog.h . . . 1713
src/ckb/kbwidget.cpp . . . 1714
src/ckb/kbwidget.h . . . 1714
src/ckb/keyaction.cpp . . . 1715
src/ckb/keyaction.h . . . 1716
src/ckb/keymap.cpp . . . 1717
src/ckb/keymap.h . . . 1739
src/ckb/keywidget.cpp . . . 1748
src/ckb/keywidget.h . . . 1751
src/ckb/kperfwidget.cpp . . . 1752
src/ckb/kperfwidget.h . . . 1752
src/ckb/layoutdialog.cpp . . . 1754
src/ckb/layoutdialog.h . . . 1755
src/ckb/macroreader.cpp . . . 1756
src/ckb/macroreader.h . . . 1756
src/ckb/main.cpp . . . 1757
src/ckb/mainwindow.cpp . . . 1763
src/ckb/mainwindow.h . . . 1764
src/ckb/media.h . . . 1766
src/ckb/media_linux.cpp . . . 1768
src/ckb/media_mac.m . . . 1769
src/ckb/modeselectdialog.cpp . . . 1770
src/ckb/modeselectdialog.h . . . 1771
src/ckb/mperfwidget.cpp . . . 1772
src/ckb/mperfwidget.h . . . 1774
src/ckb/rebindwidget.cpp . . . 1965
src/ckb/rebindwidget.h . . . 1965
src/ckb/rlistwidget.cpp . . . 2072
src/ckb/rlistwidget.h . . . 2072
src/ckb/settingswidget.cpp . . . 2073
src/ckb/settingswidget.h . . . 2075
src/ckb/ui_animadddialog.h . . . 2076
src/ckb/ui_animsettingdialog.h . . . 2076
src/ckb/ui_extrasettingswidget.h . . . 2077
src/ckb/ui_fwupgradedialog.h . . . 2078
src/ckb/ui_gradientdialog.h . . . 2079
src/ckb/ui_kbanimwidget.h . . . 2080
src/ckb/ui_kbbindwidget.h . . . 2081
src/ckb/ui_kblightwidget.h . . . 2082
src/ckb/ui_kbprofiledialog.h . . . 2083
src/ckb/ui_kbwidget.h . . . 2084
src/ckb/ui_kperfwidget.h . . . 2085
src/ckb/ui_layoutdialog.h . . . 2086
src/ckb/ui_mainwindow.h . . . 2087
src/ckb/ui_modeselectdialog.h . . . 2088
src/ckb/ui_mperfwidget.h . . . 2089
src/ckb/ui_rebindwidget.h . . . 2090

src/ckb/ui_settingswidget.h . . . 2091
src/ckb/quazip/crypt.h . . . 1776
src/ckb/quazip/ioapi.h . . . 1780
src/ckb/quazip/JlCompress.cpp . . . 1787
src/ckb/quazip/JlCompress.h . . . 1789
src/ckb/quazip/qioapi.cpp . . . 1789
src/ckb/quazip/quaadler32.cpp . . . 1801
src/ckb/quazip/quaadler32.h . . . 1802
src/ckb/quazip/quachecksum32.h . . . 1803
src/ckb/quazip/quacrc32.cpp . . . 1804
src/ckb/quazip/quacrc32.h . . . 1805
src/ckb/quazip/quagzipfile.cpp . . . 1806
src/ckb/quazip/quagzipfile.h . . . 1807
src/ckb/quazip/quaziodevice.cpp . . . 1808
src/ckb/quazip/quaziodevice.h . . . 1809
src/ckb/quazip/quazip.cpp . . . 1810
src/ckb/quazip/quazip.h . . . 1812
src/ckb/quazip/quazip_global.h . . . 1813
src/ckb/quazip/quazipdir.cpp . . . 1815
src/ckb/quazip/quazipdir.h . . . 1817
src/ckb/quazip/quazipfile.cpp . . . 1817
src/ckb/quazip/quazipfile.h . . . 1818
src/ckb/quazip/quazipfileinfo.cpp . . . 1819
src/ckb/quazip/quazipfileinfo.h . . . 1821
src/ckb/quazip/quazipnewinfo.cpp . . . 1822
src/ckb/quazip/quazipnewinfo.h . . . 1826
src/ckb/quazip/unzip.c . . . 1827
src/ckb/quazip/unzip.h . . . 1878
src/ckb/quazip/zip.c . . . 1898
src/ckb/quazip/zip.h . . . 1954
src/ckb/release/moc_animadddialog.cpp . . . 1966
src/ckb/release/moc_animscript.cpp . . . 1969
src/ckb/release/moc_animsettingdialog.cpp . . . 1972
src/ckb/release/moc_ckbsettingswriter.cpp . . . 1975
src/ckb/release/moc_colorbutton.cpp . . . 1977
src/ckb/release/moc_extrasettingswidget.cpp . . . 1979
src/ckb/release/moc_fwupgradedialog.cpp . . . 1982
src/ckb/release/moc_gradientbutton.cpp . . . 1985
src/ckb/release/moc_gradientdialog.cpp . . . 1988
src/ckb/release/moc_gradientdialogwidget.cpp . . . 1991
src/ckb/release/moc_kb.cpp . . . 1993
src/ckb/release/moc_kbanim.cpp . . . 1996
src/ckb/release/moc_kbanimwidget.cpp . . . 1998
src/ckb/release/moc_kbbind.cpp . . . 2001
src/ckb/release/moc_kbbindwidget.cpp . . . 2003
src/ckb/release/moc_kbfirmware.cpp . . . 2005
src/ckb/release/moc_kblight.cpp . . . 2008
src/ckb/release/moc_kblightwidget.cpp . . . 2011
src/ckb/release/moc_kbmanager.cpp . . . 2014
src/ckb/release/moc_kbmode.cpp . . . 2016
src/ckb/release/moc_kbperf.cpp . . . 2018
src/ckb/release/moc_kbprofile.cpp . . . 2021
src/ckb/release/moc_kbprofiledialog.cpp . . . 2023
src/ckb/release/moc_kbwidget.cpp . . . 2027
src/ckb/release/moc_keyaction.cpp . . . 2029
src/ckb/release/moc_keywidget.cpp . . . 2031
src/ckb/release/moc_kperfwidget.cpp . . . 2033
src/ckb/release/moc_layoutdialog.cpp . . . 2036

src/ckb/release/moc_macroreader.cpp . . . 2039
src/ckb/release/moc_mainwindow.cpp . . . 2043
src/ckb/release/moc_modeselectdialog.cpp . . . 2045
src/ckb/release/moc_mperfwidget.cpp . . . 2047
src/ckb/release/moc_quagzipfile.cpp . . . 2050
src/ckb/release/moc_quaziodevice.cpp . . . 2052
src/ckb/release/moc_quazipfile.cpp . . . 2054
src/ckb/release/moc_rebindwidget.cpp . . . 2056
src/ckb/release/moc_rlistwidget.cpp . . . 2059
src/ckb/release/moc_settingswidget.cpp . . . 2061
src/ckb/release/qrc_binary.cpp . . . 2063
src/ckb/release/qrc_image.cpp . . . 2066
src/ckb/release/qrc_text.cpp . . . 2069

Chapter 11

Namespace Documentation

11.1 Ui Namespace Reference

Data Structures

- class AnimAddDialog
- class AnimSettingDialog
- class ExtraSettingsWidget
- class FwUpgradeDialog
- class GradientDialog
- class KbAnimWidget
- class KbBindWidget
- class KbLightWidget
- class KbProfileDialog
- class KbWidget
- class KPerfWidget
- class LayoutDialog
- class MainWindow
- class ModeSelectDialog
- class MPerfWidget
- class RebindWidget
- class SettingsWidget

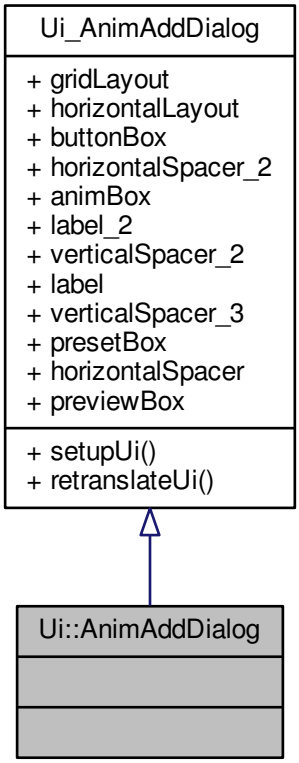
Chapter 12

Data Structure Documentation

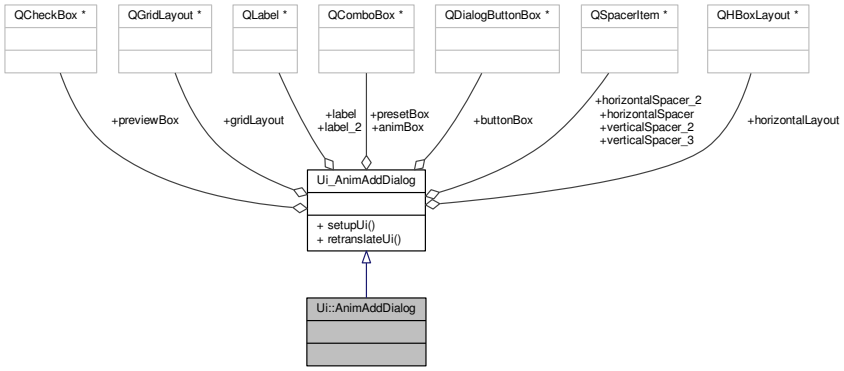
12.1 Ui::AnimAddDialog Class Reference

`#include <src/ckb/ui_animadddialog.h>`

Inheritance diagram for Ui::AnimAddDialog:



Collaboration diagram for Ui::AnimAddDialog:



Public Member Functions

- void setupUi (QDialog *AnimAddDialog)
- void retranslateUi (QDialog *AnimAddDialog)

Data Fields

- QGridLayout * gridLayout
- QHBoxLayout * horizontalLayout
- QDialogButtonBox * buttonBox
- QSpacerItem * horizontalSpacer_2
- QComboBox * animBox
- QLabel * label_2
- QSpacerItem * verticalSpacer_2
- QLabel * label
- QSpacerItem * verticalSpacer_3
- QComboBox * presetBox
- QSpacerItem * horizontalSpacer
- QCheckBox * previewBox

12.1.1 Detailed Description

Definition at line 146 of file ui_animadddialog.h.

12.1.2 Member Function Documentation

12.1.2.1 void Ui_AnimAddDialog::retranslateUi (QDialog * *AnimAddDialog*) [inline], [inherited]

Definition at line 135 of file ui_animadddialog.h.

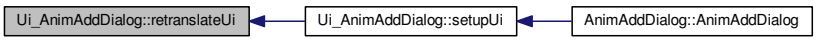
References Ui_AnimAddDialog::label, Ui_AnimAddDialog::label_2, and Ui_AnimAddDialog::previewBox.

Referenced by Ui_AnimAddDialog::setupUi().

```
136     {
137         AnimAddDialog->setWindowTitle(QApplication::translate("AnimAddDialog", "Add Animation", Q_NULLPTR))
    ;
138         label_2->setText(QApplication::translate("AnimAddDialog", "Preset:", Q_NULLPTR));
```

```
139         label->setText(QApplication::translate("AnimAddDialog", "Animation:", Q_NULLPTR));
140         previewBox->setText(QApplication::translate("AnimAddDialog", "Show preview", Q_NULLPTR));
141     } // retranslateUi
```

Here is the caller graph for this function:



12.1.2.2 void Ui_AnimAddDialog::setupUi (QDialog * *AnimAddDialog*) [inline],[inherited]

Definition at line 44 of file ui_animadddialog.h.

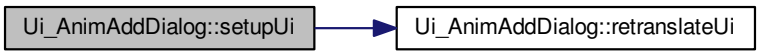
References Ui_AnimAddDialog::animBox, Ui_AnimAddDialog::buttonBox, Ui_AnimAddDialog::gridLayout, Ui_AnimAddDialog::horizontalLayout, Ui_AnimAddDialog::horizontalSpacer, Ui_AnimAddDialog::horizontalSpacer_2, Ui_AnimAddDialog::label, Ui_AnimAddDialog::label_2, Ui_AnimAddDialog::presetBox, Ui_AnimAddDialog::previewBox, Ui_AnimAddDialog::retranslateUi(), Ui_AnimAddDialog::verticalSpacer_2, and Ui_AnimAddDialog::verticalSpacer_3.

Referenced by AnimAddDialog::AnimAddDialog().

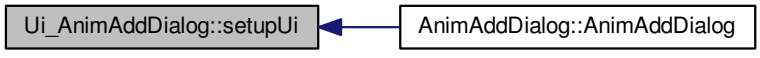
```
45     {
46         if (AnimAddDialog->objectName().isEmpty())
47             AnimAddDialog->setObjectName(QStringLiteral("AnimAddDialog"));
48         AnimAddDialog->resize(350, 205);
49         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
50         sizePolicy.setHorizontalStretch(0);
51         sizePolicy.setVerticalStretch(0);
52         sizePolicy.setHeightForWidth(AnimAddDialog->sizePolicy().hasHeightForWidth());
53         AnimAddDialog->setSizePolicy(sizePolicy);
54         gridLayout = new QGridLayout(AnimAddDialog);
55         gridLayout->setObjectName(QStringLiteral("gridLayout"));
56         horizontalLayout = new QHBoxLayout();
57         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
58         buttonBox = new QDialogButtonBox(AnimAddDialog);
59         buttonBox->setObjectName(QStringLiteral("buttonBox"));
60         buttonBox->setOrientation(Qt::Horizontal);
61         buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
62         buttonBox->setCenterButtons(false);
63
64         horizontalLayout->addWidget(buttonBox);
65
66
67         gridLayout->addLayout(horizontalLayout, 5, 0, 1, 4);
68
69         horizontalSpacer_2 = new QSpacerItem(50, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
70
71         gridLayout->addItem(horizontalSpacer_2, 1, 3, 1, 1);
72
73         animBox = new QComboBox(AnimAddDialog);
74         animBox->setObjectName(QStringLiteral("animBox"));
75         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Fixed);
76         sizePolicy1.setHorizontalStretch(0);
77         sizePolicy1.setVerticalStretch(0);
78         sizePolicy1.setHeightForWidth(animBox->sizePolicy().hasHeightForWidth());
79         animBox->setSizePolicy(sizePolicy1);
80
81         gridLayout->addWidget(animBox, 1, 2, 1, 1);
82
83         label_2 = new QLabel(AnimAddDialog);
84         label_2->setObjectName(QStringLiteral("label_2"));
85         QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Preferred);
86         sizePolicy2.setHorizontalStretch(0);
87         sizePolicy2.setVerticalStretch(0);
88         sizePolicy2.setHeightForWidth(label_2->sizePolicy().hasHeightForWidth());
89         label_2->setSizePolicy(sizePolicy2);
90         label_2->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
91
92         gridLayout->addWidget(label_2, 2, 1, 1, 1);
93
```

```
94          verticalSpacer_2 = new QSpacerItem(20, 0, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
95
96          gridLayout->addItem(verticalSpacer_2, 4, 2, 1, 1);
97
98          label = new QLabel(AnimAddDialog);
99          label->setObjectName(QStringLiteral("label"));
100          sizePolicy2.setHeightForWidth(label->sizePolicy().hasHeightForWidth());
101          label->setSizePolicy(sizePolicy2);
102          label->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
103
104          gridLayout->addWidget(label, 1, 1, 1, 1);
105
106          verticalSpacer_3 = new QSpacerItem(20, 0, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
107
108          gridLayout->addItem(verticalSpacer_3, 0, 2, 1, 1);
109
110          presetBox = new QComboBox(AnimAddDialog);
111          presetBox->setObjectName(QStringLiteral("presetBox"));
112          sizePolicy1.setHeightForWidth(presetBox->sizePolicy().hasHeightForWidth());
113          presetBox->setSizePolicy(sizePolicy1);
114
115          gridLayout->addWidget(presetBox, 2, 2, 1, 1);
116
117          horizontalSpacer = new QSpacerItem(50, 20, QSizePolicy::Fixed, QSizePolicy::Minimum
    );
118
119          gridLayout->addItem(horizontalSpacer, 1, 0, 1, 1);
120
121          previewBox = new QCheckBox(AnimAddDialog);
122          previewBox->setObjectName(QStringLiteral("previewBox"));
123          previewBox->setChecked(true);
124
125          gridLayout->addWidget(previewBox, 3, 2, 1, 1);
126
127
128          retranslateUi(AnimAddDialog);
129          QObject::connect(buttonBox, SIGNAL(accepted()), AnimAddDialog, SLOT(accept()));
130          QObject::connect(buttonBox, SIGNAL(rejected()), AnimAddDialog, SLOT(reject()));
131
132          QMetaObject::connectSlotsByName(AnimAddDialog);
133      } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.1.3 Field Documentation

12.1.3.1 QComboBox* Ui_AnimAddDialog::animBox [inherited]

Definition at line 35 of file ui_animadddialog.h.

Referenced by AnimAddDialog::AnimAddDialog(), AnimAddDialog::chosenScript(), and Ui_AnimAddDialog::setupUi().

12.1.3.2 QDialogButtonBox* Ui_AnimAddDialog::buttonBox [inherited]

Definition at line 33 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

12.1.3.3 QGridLayout* Ui_AnimAddDialog::gridLayout [inherited]

Definition at line 31 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

12.1.3.4 QHBoxLayout* Ui_AnimAddDialog::horizontalLayout [inherited]

Definition at line 32 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

12.1.3.5 QSpacerItem* Ui_AnimAddDialog::horizontalSpacer [inherited]

Definition at line 41 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

12.1.3.6 QSpacerItem* Ui_AnimAddDialog::horizontalSpacer_2 [inherited]

Definition at line 34 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

12.1.3.7 QLabel* Ui_AnimAddDialog::label [inherited]

Definition at line 38 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::retranslateUi(), and Ui_AnimAddDialog::setupUi().

12.1.3.8 QLabel* Ui_AnimAddDialog::label_2 [inherited]

Definition at line 36 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::retranslateUi(), and Ui_AnimAddDialog::setupUi().

12.1.3.9 QComboBox* Ui_AnimAddDialog::presetBox [inherited]

Definition at line 40 of file ui_animadddialog.h.

Referenced by AnimAddDialog::chosenPreset(), AnimAddDialog::on_animBox_activated(), AnimAddDialog::on_previewBox_clicked(), and Ui_AnimAddDialog::setupUi().

12.1.3.10 QCheckBox* Ui_AnimAddDialog::previewBox [inherited]

Definition at line 42 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::retranslateUi(), and Ui_AnimAddDialog::setupUi().

12.1.3.11 QSpacerItem* Ui_AnimAddDialog::verticalSpacer_2 [inherited]

Definition at line 37 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

12.1.3.12 QSpacerItem* Ui_AnimAddDialog::verticalSpacer_3 [inherited]

Definition at line 39 of file ui_animadddialog.h.

Referenced by Ui_AnimAddDialog::setupUi().

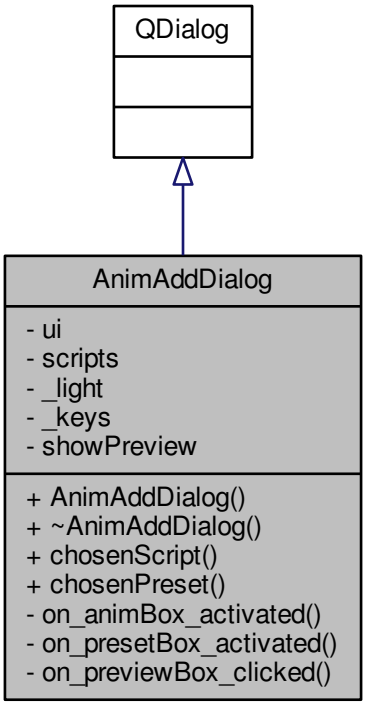
The documentation for this class was generated from the following file:

- src/ckb/ui_animadddialog.h

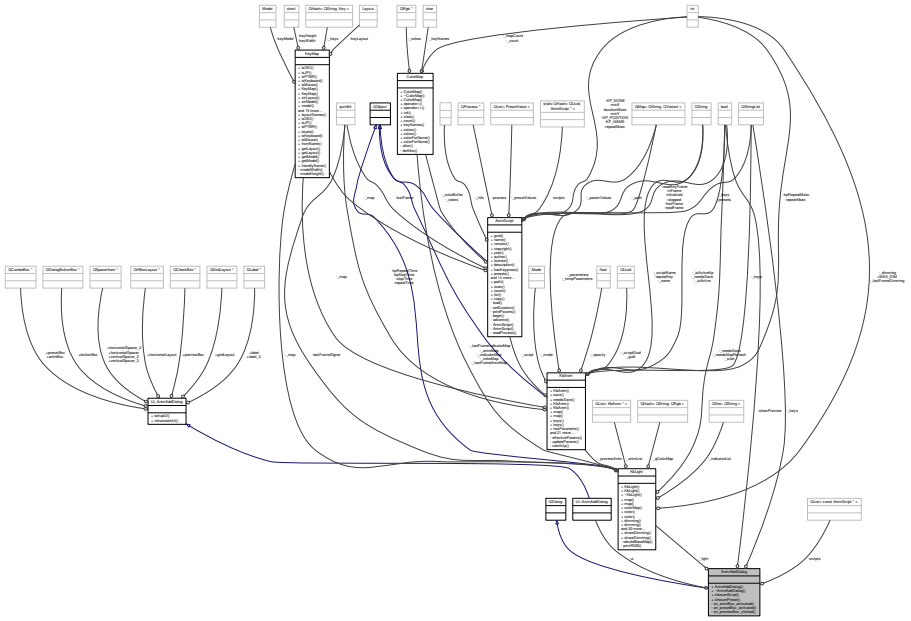
12.2 AnimAddDialog Class Reference

`#include <src/ckb/animadddialog.h>`

Inheritance diagram for AnimAddDialog:



Collaboration diagram for AnimAddDialog:



Public Member Functions

- AnimAddDialog (QWidget *parent, KbLight *light, const QStringList &keys)
- ~AnimAddDialog ()
- const AnimScript * chosenScript ()
- int chosenPreset ()

Private Slots

- void on_animBox_activated (int index)
- void on_presetBox_activated (int index)
- void on_previewBox_clicked (bool checked)

Private Attributes

- Ui::AnimAddDialog * ui
- QList< const AnimScript * > scripts
- KbLight * _light
- QStringList _keys
- bool showPreview

12.2.1 Detailed Description

Definition at line 12 of file animadddialog.h.

12.2.2 Constructor & Destructor Documentation

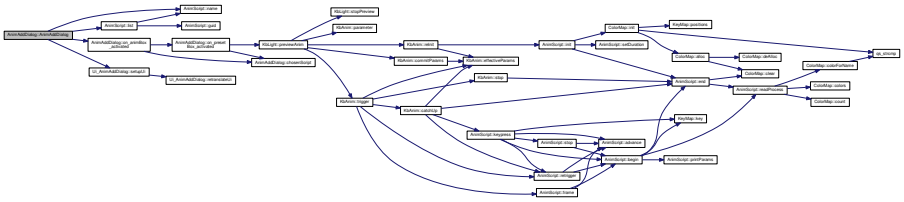
12.2.2.1 AnimAddDialog::AnimAddDialog (QWidget * *parent*, KbLight * *light*, const QStringList & *keys*) [explicit]

Definition at line 4 of file animadddialog.cpp.

References Ui_AnimAddDialog::animBox, AnimScript::list(), AnimScript::name(), on_animBox_activated(), scripts, Ui_AnimAddDialog::setupUi(), and ui.

```
4                                                                                :
5     QDialog(parent),
6     ui(new Ui::AnimAddDialog), _light(light), _keys(keys),
      showPreview(true)
7 {
8     ui->setupUi(this);
9     scripts = AnimScript::list();
10     foreach(const AnimScript* script, scripts)
11         ui->animBox->addItem(script->name());
12     on_animBox_activated(0);
13 }
```

Here is the call graph for this function:



12.2.2.2 AnimAddDialog::~AnimAddDialog ( )

Definition at line 15 of file animadddialog.cpp.

References _light, KbLight::stopPreview(), and ui.

```
15                                {
16     delete ui;
17     _light->stopPreview();
18 }
```

Here is the call graph for this function:

12.2.3 Member Function Documentation

12.2.3.1 int AnimAddDialog::chosenPreset ( )

Definition at line 24 of file animadddialog.cpp.

References Ui_AnimAddDialog::presetBox, and ui.

Referenced by KbLightWidget::on_animButton_clicked().

```
24                                {
25     return ui->presetBox->currentIndex();
26 }
```

Here is the caller graph for this function:

12.2.3.2 const AnimScript * AnimAddDialog::chosenScript ( )

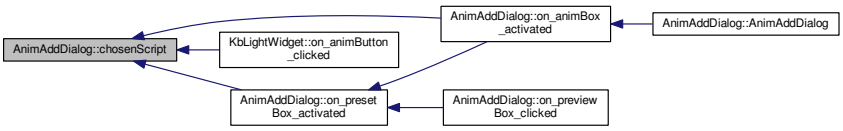
Definition at line 20 of file animadddialog.cpp.

References Ui_AnimAddDialog::animBox, scripts, and ui.

Referenced by on_animBox_activated(), KbLightWidget::on_animButton_clicked(), and on_presetBox_activated().

```
20                                          {
21     return scripts[ui->animBox->currentIndex()];
22 }
```

Here is the caller graph for this function:



12.2.3.3 void AnimAddDialog::on_animBox_activated (int *index*) [private], [slot]

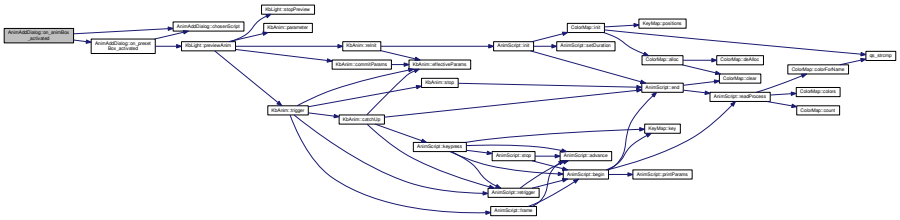
Definition at line 28 of file animadddialog.cpp.

References chosenScript(), on_presetBox_activated(), Ui_AnimAddDialog::presetBox, and ui.

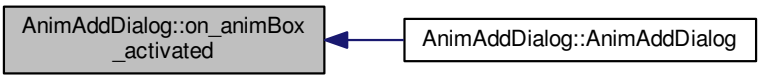
Referenced by AnimAddDialog().

```
28                                              {
29     // Update preset list
30     ui->presetBox->clear();
31     foreach(const QString& preset, chosenScript()->presets())
32         ui->presetBox->addItem(preset);
33     ui->presetBox->setCurrentIndex(0);
34     on_presetBox_activated(0);
35 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.2.3.4 void AnimAddDialog::on_presetBox_activated (int *index*) [private], [slot]

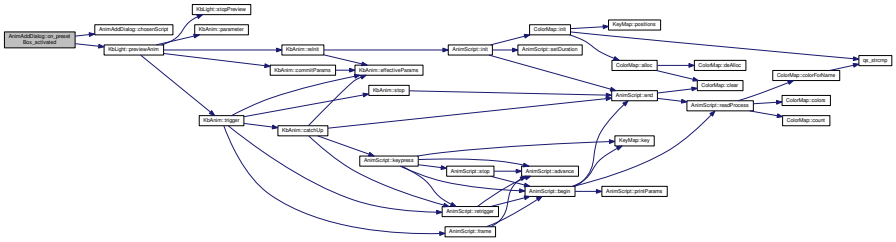
Definition at line 37 of file animadddialog.cpp.

References _keys, _light, chosenScript(), KbLight::previewAnim(), and showPreview.

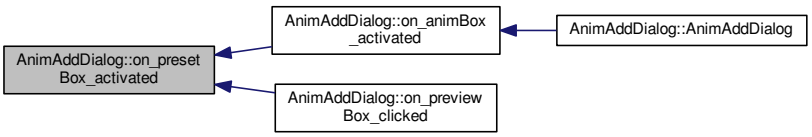
Referenced by on_animBox_activated(), and on_previewBox_clicked().

```
37                                                  {
38     // Update preview
39     if(showPreview)
40         _light->previewAnim(chosenScript(), _keys,
       chosenScript()->preset(index));
41 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



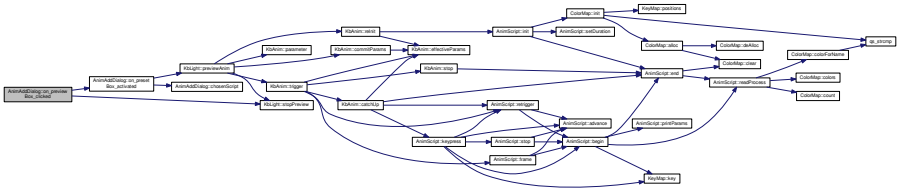
12.2.3.5 void AnimAddDialog::on_previewBox_clicked (bool *checked*) [private], [slot]

Definition at line 43 of file animadddialog.cpp.

References _light, on_presetBox_activated(), Ui_AnimAddDialog::presetBox, showPreview, KbLight::stopPreview(), and ui.

```
43                                                  {
44     showPreview = checked;
45     if(showPreview)
46         on_presetBox_activated(ui->presetBox->currentIndex());
47     else
48         _light->stopPreview();
49 }
```

Here is the call graph for this function:



12.2.4 Field Documentation

12.2.4.1 QStringList AnimAddDialog::_keys [private]

Definition at line 33 of file animadddialog.h.

Referenced by on_presetBox_activated().

12.2.4.2 KbLight* AnimAddDialog::_light [private]

Definition at line 32 of file animadddialog.h.

Referenced by on_presetBox_activated(), on_previewBox_clicked(), and ~AnimAddDialog().

12.2.4.3 QList<const AnimScript*> AnimAddDialog::scripts [private]

Definition at line 30 of file animadddialog.h.

Referenced by AnimAddDialog(), and chosenScript().

12.2.4.4 bool AnimAddDialog::showPreview [private]

Definition at line 34 of file animadddialog.h.

Referenced by on_presetBox_activated(), and on_previewBox_clicked().

12.2.4.5 Ui::AnimAddDialog* AnimAddDialog::ui [private]

Definition at line 29 of file animadddialog.h.

Referenced by AnimAddDialog(), chosenPreset(), chosenScript(), on_animBox_activated(), on_previewBox_clicked(), and ~AnimAddDialog().

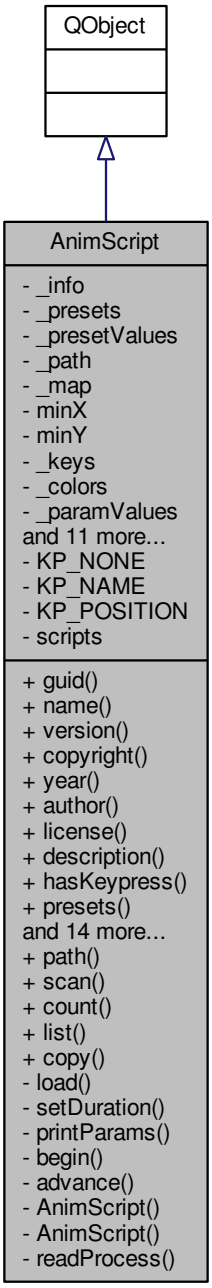
The documentation for this class was generated from the following files:

- src/ckb/animadddialog.h
- src/ckb/animadddialog.cpp

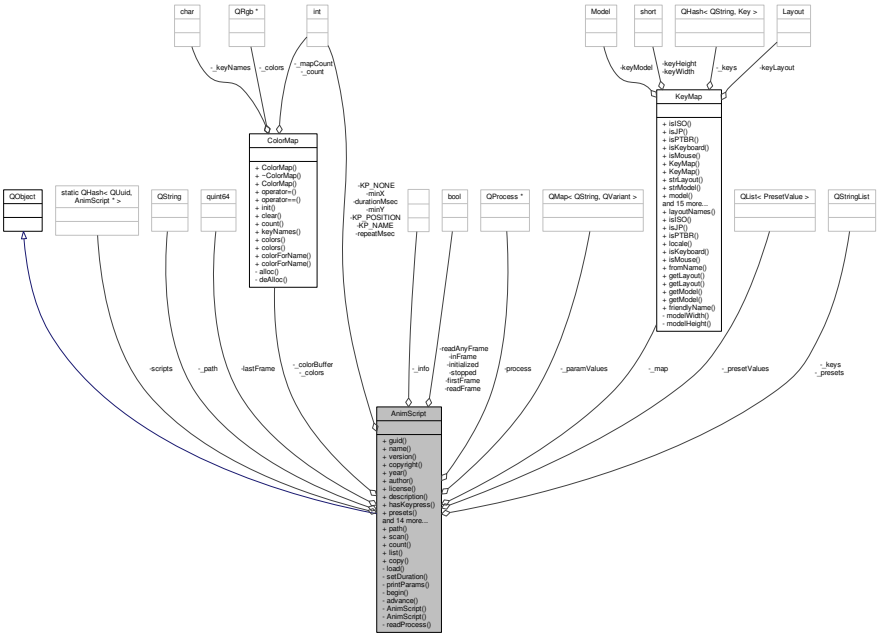
12.3 AnimScript Class Reference

`#include <src/ckb/animscript.h>`

Inheritance diagram for AnimScript:



Collaboration diagram for AnimScript:



Data Structures

- struct Param

Public Types

- typedef QMap< QString, QVariant > PresetValue

Public Member Functions

- const QUuid & guid () const
- const QString & name () const
- const QString & version () const
- QString copyright () const
- const QString & year () const
- const QString & author () const
- const QString & license () const
- const QString & description () const
- bool hasKeypress () const
- const QStringList & presets () const
- const PresetValue & preset (int index) const
- QListIterator< Param > paramIterator () const
- Param param (const QString &name) const
- bool hasParam (const QString &name) const
- void init (const KeyMap &map, const QStringList &keys, const QMap< QString, QVariant > &paramValues)
- void parameters (const QMap< QString, QVariant > &paramValues)
- void retrigger (quint64 timestamp, bool allowPreempt=false)
- void stop (quint64 timestamp)
- void keypress (const QString &key, bool pressed, quint64 timestamp)

- void frame (quint64 timestamp)
- void end ()
- bool hasFrame () const
- const ColorMap & colors () const
- ~AnimScript ()

Static Public Member Functions

- static QString path ()
- static void scan ()
- static int count ()
- static QList< const AnimScript * > list ()
- static AnimScript * copy (QObject *parent, const QUuid &id)

Private Slots

- void readProcess ()

Private Member Functions

- bool load ()
- void setDuration ()
- void printParams ()
- void begin (quint64 timestamp)
- void advance (quint64 timestamp)
- AnimScript (QObject *parent, const QString &path)
- AnimScript (QObject *parent, const AnimScript &base)

Private Attributes

- struct {
  QUuid **guid**
  QString **name**
  QString **version**
  QString **year**
  QString **author**
  QString **license**
  QString **description**
  QList< Param > **params**
  int **kpMode**:3
  bool **absoluteTime**:1
  bool **repeat**:1
  bool **preempt**:1
  bool **liveParams**:1
  } _info

- QStringList _presets
- QList< PresetValue > _presetValues
- QString _path
- KeyMap _map
- int minX
- int minY
- QStringList _keys

- ColorMap _colors
- PresetValue _paramValues
- quint64 lastFrame
- int durationMsec
- int repeatMsec
- bool initialized:1
- bool firstFrame:1
- bool readFrame:1
- bool readAnyFrame:1
- bool stopped:1
- bool inFrame:1
- QProcess * process
- ColorMap _colorBuffer

Static Private Attributes

- static const int KP_NONE = 0
- static const int KP_NAME = 1
- static const int KP_POSITION = 2
- static QHash< QUuid, AnimScript * > scripts

12.3.1 Detailed Description

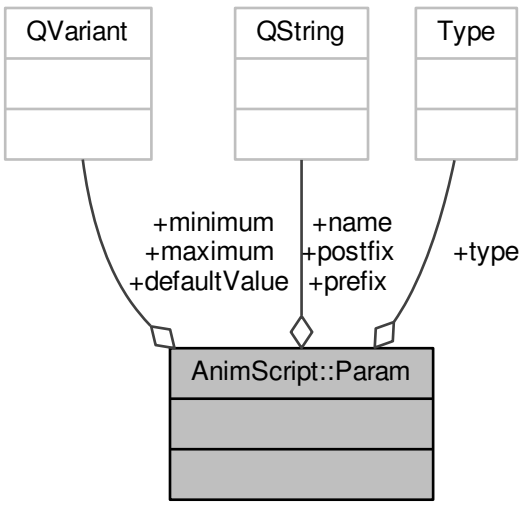
Definition at line 16 of file animscript.h.

12.3.2 Data Structure Documentation

12.3.2.1 struct AnimScript::Param

Definition at line 21 of file animscript.h.

Collaboration diagram for AnimScript::Param:



Data Fields

	Type	

Data Fields

QVariant	defaultValue	
QVariant	maximum	
QVariant	minimum	
QString	name	
QString	postfix	
QString	prefix	
Type	type	

12.3.3 Member Typedef Documentation

12.3.3.1 typedef QMap<QString, QVariant> AnimScript::PresetValue

Definition at line 58 of file animscript.h.

12.3.4 Constructor & Destructor Documentation

12.3.4.1 AnimScript::~AnimScript ( )

Definition at line 21 of file animscript.cpp.

References process.

```
21                          {
22     if(process){
23         process->kill();
24         process->waitForFinished(1000);
25         delete process;
26     }
27 }
```

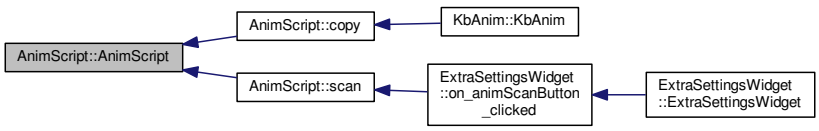
12.3.4.2 AnimScript::AnimScript (QObject * *parent*, const QString & *path*) [private]

Definition at line 11 of file animscript.cpp.

Referenced by copy(), and scan().

```
11                                                          :
12     QObject(parent), _path(path), initialized(false),
       process(0)
13 {
14 }
```

Here is the caller graph for this function:



12.3.4.3 AnimScript::AnimScript (QObject * *parent*, const AnimScript & *base*) [private]

Definition at line 16 of file animscript.cpp.

```
16                                                          :
17     QObject(parent), _info(base._info), _path(base._path),
       initialized(false), process(0)
18 {
19 }
```

12.3.5 Member Function Documentation

12.3.5.1 void AnimScript::advance (quint64 *timestamp*) [private]

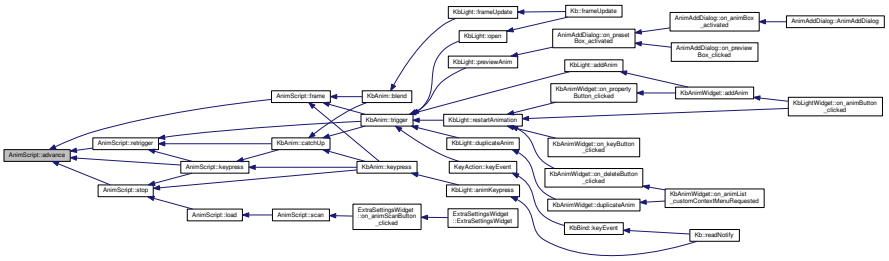
Definition at line 445 of file animscript.cpp.

References _info, durationMsec, lastFrame, and process.

Referenced by frame(), keypress(), retrigger(), and stop().

```
445                                          {
446     if(timestamp <= lastFrame || !process)
447         // Don't do anything if the time hasn't actually advanced.
448         return;
449     double delta = (timestamp - lastFrame) / (double)durationMsec;
450     if(!_info.absoluteTime){
451         // Skip any complete durations
452         while(delta > 1.){
453             process->write("time 1\n");
454             delta--;
455         }
456     }
457     process->write(QString("time %1\n").arg(delta).toLatin1());
458     lastFrame = timestamp;
459 }
```

Here is the caller graph for this function:



12.3.5.2 const QString& AnimScript::author ( ) const [inline]

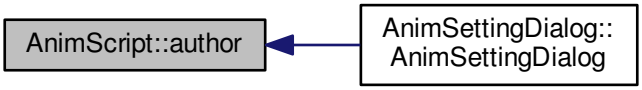
Definition at line 64 of file animscript.h.

References _info.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
64 { return _info.author; }
```

Here is the caller graph for this function:



12.3.5.3 void AnimScript::begin (quint64 *timestamp*) [private]

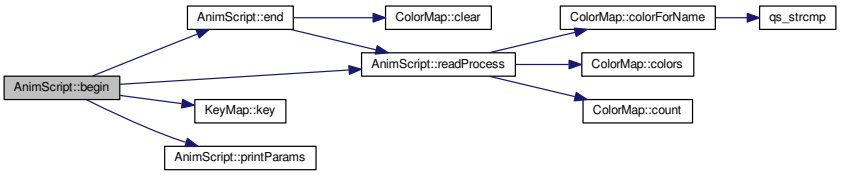
Definition at line 287 of file animscript.cpp.

References _keys, _map, _path, end(), firstFrame, inFrame, initialized, KeyMap::key(), lastFrame, minX, minY, printParams(), process, readAnyFrame, readFrame, readProcess(), stopped, Key::x, and Key::y.

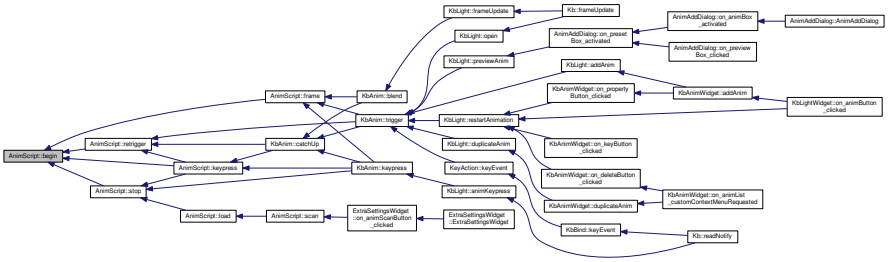
Referenced by frame(), keypress(), retrigger(), and stop().

```
287                                     {
288     if(!initialized)
289         return;
290     end();
291     stopped = firstFrame = readFrame = readAnyFrame =
      inFrame = false;
292     // Determine the upper left corner of the given keys
293     QStringList keysCopy = _keys;
294     minX = INT_MAX;
295     minY = INT_MAX;
296     foreach(const QString& key, keysCopy){
297         const Key& pos = _map.key(key);
298         if(!pos){
299             keysCopy.removeAll(key);
300             continue;
301         }
302         if(pos.x < minX)
303             minX = pos.x;
304         if(pos.y < minY)
305             minY = pos.y;
306     }
307     if(keysCopy.isEmpty()){
308         // If the key list is empty, don't actually start the animation but pretend it's running anyway
309         firstFrame = readFrame = readAnyFrame = true;
310         return;
311     }
312     process = new QProcess(this);
313     connect(process, SIGNAL(readyRead()), this, SLOT(readProcess()));
314     process->start(_path, QStringList("--ckb-run"));
315     qDebug() << "Starting " << _path;
316     // Write the keymap to the process
317     process->write("begin keymap\n");
318     process->write(QString("keycount %1\n").arg(keysCopy.count()).toLatin1());
319     foreach(const QString& key, keysCopy){
320         const Key& pos = _map.key(key);
321         process->write(QString("key %1 %2,%3\n").arg(key).arg(pos.x -
      minX).arg(pos.y - minY).toLatin1());
322     }
323     process->write("end keymap\n");
324     // Write parameters
325     printParams();
326     // Begin animating
327     process->write("begin run\n");
328     lastFrame = timestamp;
329 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.4 const ColorMap& AnimScript::colors ( ) const [inline]

Definition at line 99 of file animscript.h.

References _colors.

Referenced by KbAnim::blend().

```
99 { return _colors; }
```

Here is the caller graph for this function:

12.3.5.5 AnimScript * AnimScript::copy (QObject * *parent*, const QUuid & *id*) [static]

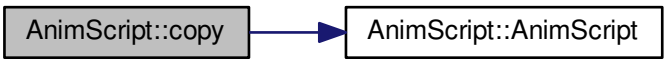
Definition at line 67 of file animscript.cpp.

References AnimScript(), and scripts.

Referenced by KbAnim::KbAnim().

```
67                                                      {
68     return scripts.contains(id) ? new AnimScript(parent, *
    scripts.value(id)) : 0;
69 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.5.6 QString AnimScript::copyright ( ) const [inline]

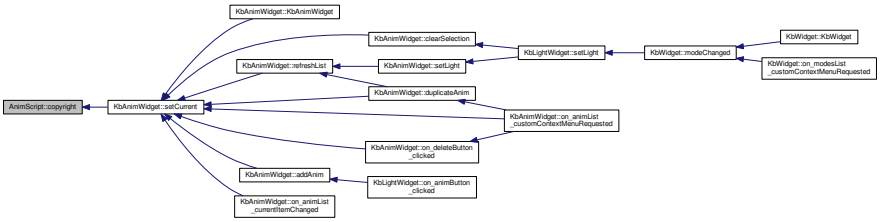
Definition at line 62 of file animscript.h.

References _info.

Referenced by KbAnimWidget::setCurrent().

```
62 { return "Copyright © " + _info.year + " " + _info.author; }
```

Here is the caller graph for this function:



12.3.5.7 static int AnimScript::count ( ) [inline], [static]

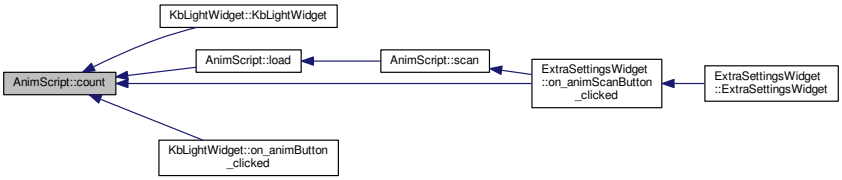
Definition at line 54 of file animscript.h.

References scripts.

Referenced by KbLightWidget::KbLightWidget(), load(), KbLightWidget::on_animButton_clicked(), and ExtraSettingsWidget::on_animScanButton_clicked().

```
54 { return scripts.count(); }
```

Here is the caller graph for this function:



12.3.5.8 const QString& AnimScript::description ( ) const [inline]

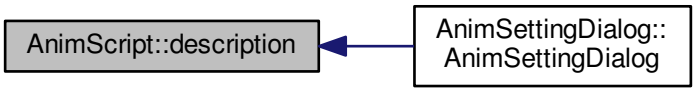
Definition at line 66 of file animscript.h.

References _info.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
66 { return _info.description; }
```

Here is the caller graph for this function:



12.3.5.9 void AnimScript::end ( )

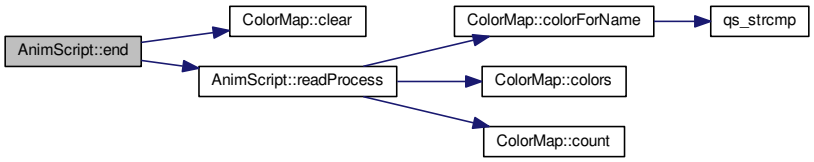
Definition at line 387 of file animscript.cpp.

References _colors, ColorMap::clear(), process, and readProcess().

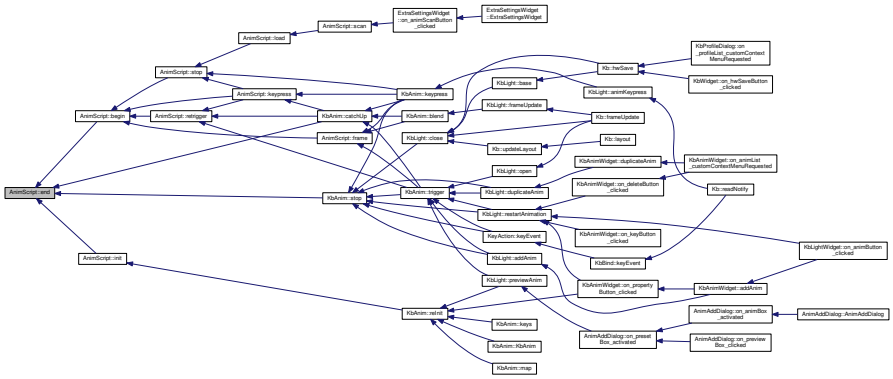
Referenced by begin(), KbAnim::catchUp(), init(), and KbAnim::stop().

```
387                     {
388     _colors.clear();
389     if(process){
390         process->kill();
391         connect(process, SIGNAL(finished(int)), process, SLOT(deleteLater()));
392         disconnect(process, SIGNAL(readyRead()), this, SLOT(readProcess()));
393         process = 0;
394     }
395 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.10 void AnimScript::frame (quint64 *timestamp*)

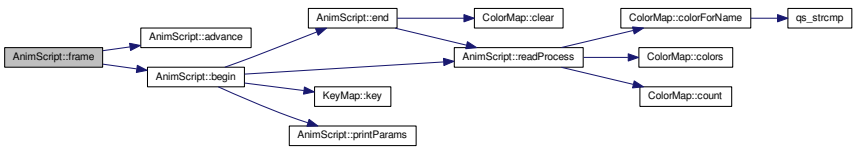
Definition at line 430 of file animscript.cpp.

References advance(), begin(), firstFrame, initialized, process, readFrame, and stopped.

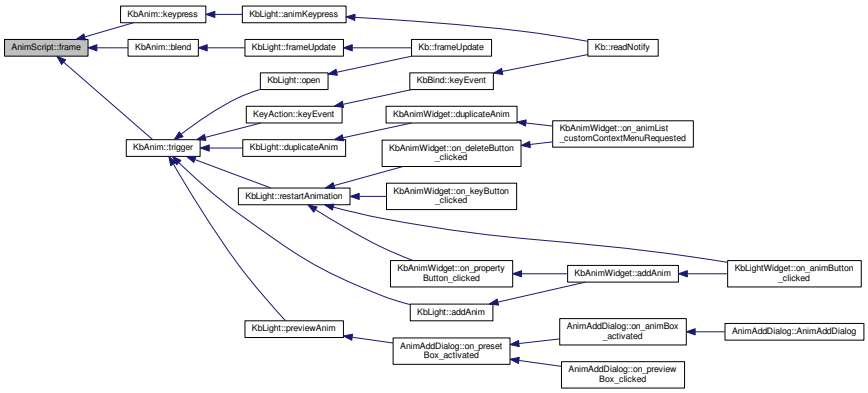
Referenced by KbAnim::blend(), KbAnim::keypress(), and KbAnim::trigger().

```
430                                     {
431     if(!initialized || stopped)
432         return;
433     // Start the animation if it's not running yet
434     if(!process)
435         begin(timestamp);
436
437     advance(timestamp);
438     if((readFrame || !firstFrame) && process)
439         // Don't ask for a new frame if the animation hasn't delivered the last one yet
440         process->write("frame\n");
441     firstFrame = true;
442     readFrame = false;
443 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.11 const QUuid& AnimScript::guid ( ) const [inline]

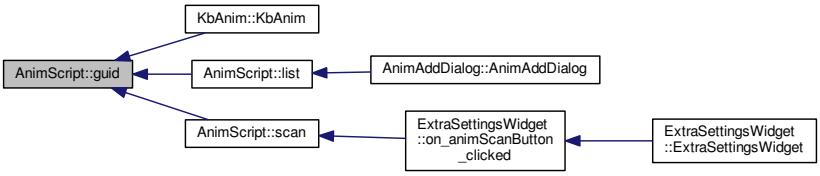
Definition at line 59 of file animscript.h.

References _info.

Referenced by KbAnim::KbAnim(), list(), and scan().

```
59 { return _info.guid; }
```

Here is the caller graph for this function:



12.3.5.12 bool AnimScript::hasFrame ( ) const [inline]

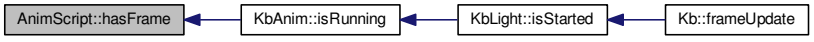
Definition at line 96 of file animscript.h.

References initialized, and readAnyFrame.

Referenced by KbAnim::isRunning().

```
96 { return initialized && readAnyFrame; }
```

Here is the caller graph for this function:



12.3.5.13 bool AnimScript::hasKeypress ( ) const [inline]

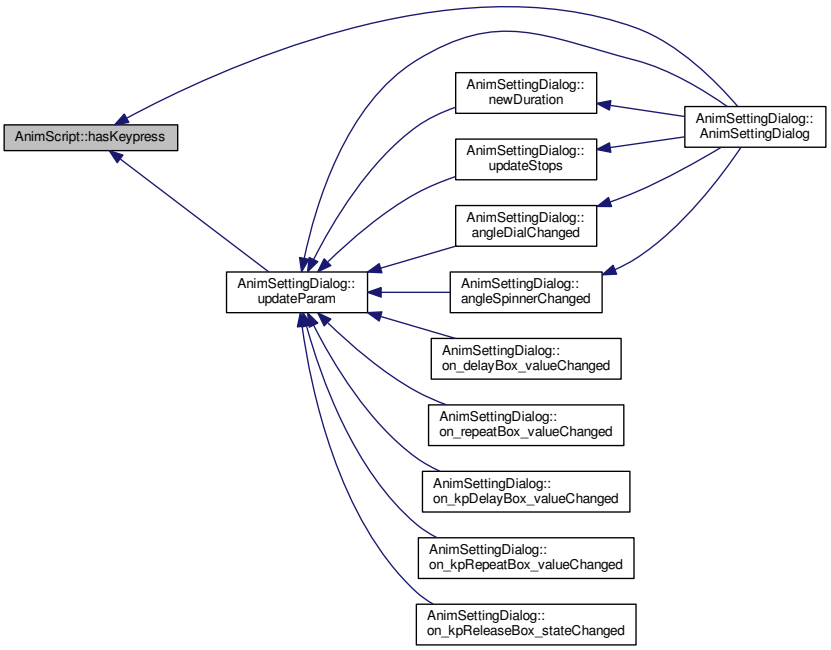
Definition at line 67 of file animscript.h.

References _info, and KP_NONE.

Referenced by AnimSettingDialog::AnimSettingDialog(), and AnimSettingDialog::updateParam().

```
67 { return _info.kpMode != KP_NONE; }
```

Here is the caller graph for this function:



12.3.5.14 bool AnimScript::hasParam (const QString & *name*) const [inline]

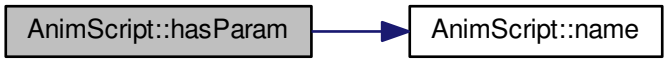
Definition at line 74 of file animscript.h.

References _info, and name().

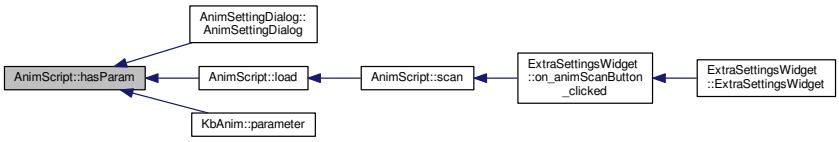
Referenced by AnimSettingDialog::AnimSettingDialog(), load(), and KbAnim::parameter().

```
74 { QListIterator<Param> i(_info.params); while(i.hasNext()){ if(i.next().name ==
     name) return true; } return false; }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.15 void AnimScript::init (const KeyMap & *map,* const QStringList & *keys,* const QMap< QString, QVariant > & *paramValues*)

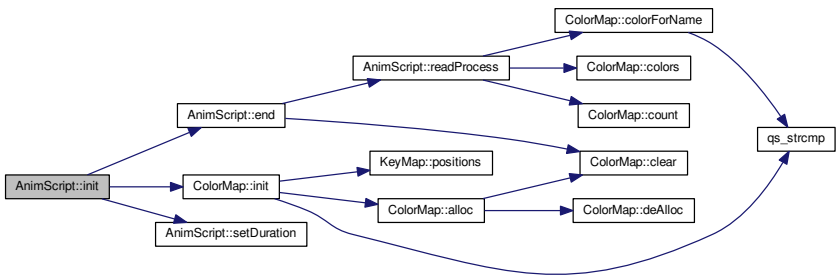
Definition at line 239 of file animscript.cpp.

References _colorBuffer, _colors, _keys, _map, _paramValues, _path, end(), firstFrame, ColorMap::init(), initialized, setDuration(), and stopped.

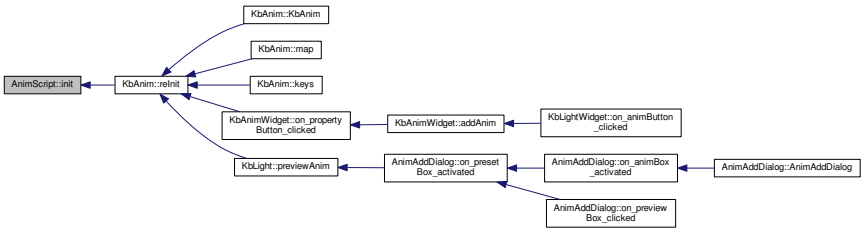
Referenced by KbAnim::reInit().

```
239
    {
240     if(_path == "")
241         return;
242     end();
243     _map = map;
244     _colors.init(map);
245     _colorBuffer.init(map);
246     _keys = keys;
247     _paramValues = paramValues;
248     setDuration();
249     stopped = firstFrame = false;
250     initialized = true;
251 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.16 void AnimScript::keypress (const QString & *key,* bool *pressed,* quint64 *timestamp*)

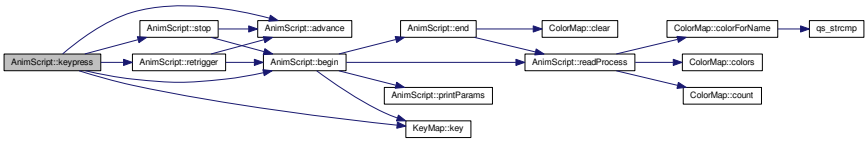
Definition at line 354 of file animscript.cpp.

References _info, _map, _paramValues, advance(), begin(), initialized, KeyMap::key(), KP_NAME, KP_NONE, KP-_POSITION, minX, minY, process, retrigger(), stop(), Key::x, and Key::y.

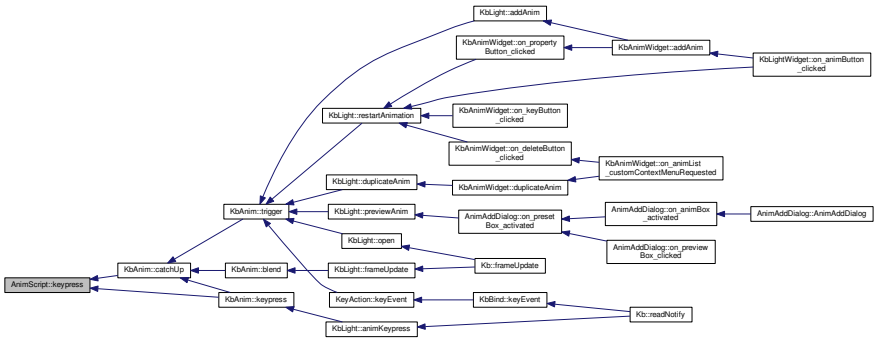
Referenced by KbAnim::catchUp(), and KbAnim::keypress().

```
354                                                                    {
355     if(!initialized)
356         return;
357     if(!process)
358         begin(timestamp);
359     int kpMode = _info.kpMode;
360     if(_paramValues.value("kpmode", 0).toInt() != 0)
361         // Disable KP mode according to user preferences
362         kpMode = KP_NONE;
363     switch(kpMode){
364     case KP_NONE:
365         // If KPs aren't allowed, call retrigger/stop instead
366         if(pressed)
367             retrigger(timestamp);
368         else if(_paramValues.value("kprelease", false).toBool())
369             stop(timestamp);
370         break;
371     case KP_NAME:
372         // Print keypress by name
373         advance(timestamp);
374         process->write(("key " + key + (pressed ? " down\n" : " up\n")).toLatin1());
375         break;
376     case KP_POSITION:
377         // Print keypress by position
378         const Key& kp = _map.key(key);
379         if(!kp)
380             return;
381         advance(timestamp);
382         process->write(("key " + QString("%1,%2").arg(kp.x - minX).arg(kp.
      y - minY) + (pressed ? " down\n" : " up\n")).toLatin1());
383         break;
384     }
385 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.17 const QString& AnimScript::license ( ) const [inline]

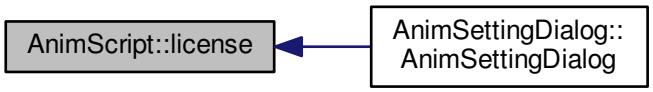
Definition at line 65 of file animscript.h.

References _info.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
65 { return _info.license; }
```

Here is the caller graph for this function:



12.3.5.18 QList< const AnimScript * > AnimScript::list ( ) [static]

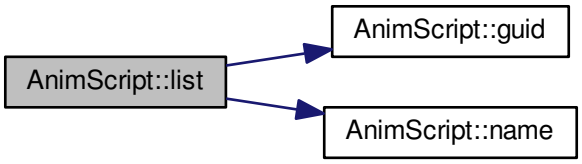
Definition at line 51 of file animscript.cpp.

References _info, guid(), name(), and scripts.

Referenced by AnimAddDialog::AnimAddDialog().

```
51                                        {
52     // Gather the animations into an alphabetically-sorted list
53     QMap<QString, const AnimScript*> result;
54     foreach(AnimScript* script, scripts.values()){
55         QString name = script->name();
56         if(result.contains(name)){
57             // If duplicate names exist, make them unique by including their GUIDs
58             AnimScript* last = (AnimScript*)result[name];
59             last->_info.name += " " + last->guid().toString().toUpper();
60             script->_info.name += " " + script->guid().toString().toUpper();
61         }
62         result[script->name()] = script;
63     }
64     return result.values();
65 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.5.19 bool AnimScript::load ( ) [private]

Definition at line 77 of file animscript.cpp.

References _info, _path, _presets, _presetValues, AnimScript::Param::AGRADIENT, AnimScript::Param::ANGLE, AnimScript::Param::ARGB, AnimScript::Param::BOOL, count(), AnimScript::Param::defaultValue, AnimScript::Param::DOUBLE, AnimScript::Param::GRADIENT, hasParam(), AnimScript::Param::INVALID, KP_NAME, KP_NONE, KP_POSITION, kprelease, AnimScript::Param::LABEL, AnimScript::Param::LONG, name(), ONE_DAY, param(), preset(), AnimScript::Param::RGB, stop(), AnimScript::Param::STRING, and urlParam().

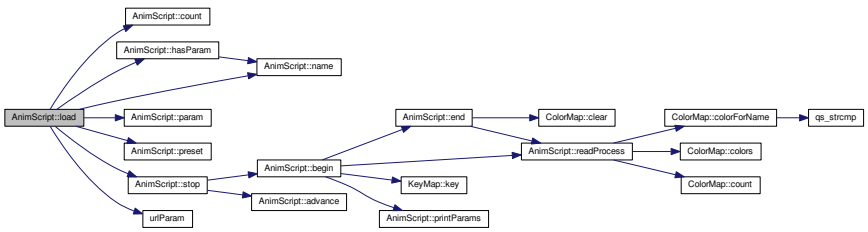
Referenced by scan().

```
77                      {
78     // Run the process to get script info
79     QProcess infoProcess;
80     infoProcess.start(_path, QStringList("--ckb-info"));
81     qDebug() << "Scanning " << _path;
82     infoProcess.waitForFinished(1000);
83     if(infoProcess.state() == QProcess::Running){
84         // Kill the process if it takes more than 1s
85         infoProcess.kill();
86         return false;
87     }
88     // Set defaults for performance info
89     _info.kpMode = KP_NONE;
90     _info.absoluteTime = _info.preempt = _info.liveParams = false;
91     _info.repeat = true;
92     // Read output
93     QString line;
94     while((line = infoProcess.readLine()) != ""){
95         line = line.trimmed();
96         QStringList components = line.split(" ");
97         int count = components.count();
98         if(count < 2)
99             continue;
100         QString param = components[0].trimmed();
101         if(param == "guid")
102             _info.guid = QUuid(urlParam(components[1]));
```

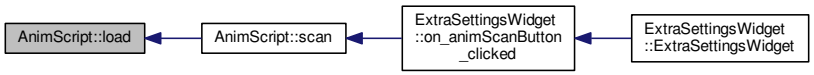
```
103         else if(param == "name")
104             _info.name = urlParam(components[1]);
105         else if(param == "version")
106             _info.version = urlParam(components[1]);
107         else if(param == "year")
108             _info.year = urlParam(components[1]);
109         else if(param == "author")
110             _info.author = urlParam(components[1]);
111         else if(param == "license")
112             _info.license = urlParam(components[1]);
113         else if(param == "description")
114             _info.description = urlParam(components[1]);
115         else if(param == "kpmode")
116             _info.kpMode = (components[1] == "position") ? KP_POSITION : (components[1] ==
    "name") ? KP_NAME : KP_NONE;
117         else if(param == "time")
118             _info.absoluteTime = (components[1] == "absolute");
119         else if(param == "repeat")
120             _info.repeat = (components[1] == "on");
121         else if(param == "preempt")
122             _info.preempt = (components[1] == "on");
123         else if(param == "parammode")
124             _info.liveParams = (components[1] == "live");
125         else if(param == "param"){
126             // Read parameter
127             if(count < 3)
128                 continue;
129             while(components.count() < 8)
130                 components.append("");
131             Param::Type type = Param::INVALID;
132             QString sType = components[1].toLower();
133             if(sType == "long")
134                 type = Param::LONG;
135             else if(sType == "double")
136                 type = Param::DOUBLE;
137             else if(sType == "bool")
138                 type = Param::BOOL;
139             else if(sType == "rgb")
140                 type = Param::RGB;
141             else if(sType == "argb")
142                 type = Param::ARGB;
143             else if(sType == "gradient")
144                 type = Param::GRADIENT;
145             else if(sType == "agradient")
146                 type = Param::AGRADIENT;
147             else if(sType == "angle")
148                 type = Param::ANGLE;
149             else if(sType == "string")
150                 type = Param::STRING;
151             else if(sType == "label")
152                 type = Param::LABEL;
153             else
154                 continue;
155             // "param <type> <name> <prefix> <postfix> <default>"
156             QString name = components[2].toLower();
157             // Make sure it's not present already
158             if(hasParam(name))
159                 continue;
160             QString prefix = urlParam(components[3]), postfix = urlParam(components[4]);
161             QVariant def = urlParam(components[5]), minimum = urlParam(components[6]),
    maximum = urlParam(components[7]);
162             // Don't allow predefined params
163             if(name == "trigger" || name == "kptrigger" || name == "kpmode" || name == "duration" || name
    == "delay" || name == "kpdelay" || name == "repeat" || name == "kprepeat" || name == "stop" || name == "
    kpstop" || name == "kpmodestop" || name == "kprelease")
164                 continue;
165             Param param = { type, name, prefix, postfix, def, minimum, maximum };
166             _info.params.append(param);
167         } else if(param == "preset"){
168             // Add preset
169             QString name = urlParam(components.at(1));
170             QMap<QString, QVariant> preset;
171             for(int i = 2; i < count; i++){
172                 // Scan name/value setting pairs
173                 QString setting = components.at(i);
174                 QStringList sComponents = setting.split("=");
175                 if(sComponents.count() != 2)
176                     continue;
177                 QString param = sComponents.first().trimmed();
178                 QString value = urlParam(sComponents.last());
179                 preset[param] = value;
180             }
181             // If the preset contains a duration, set the repeat time to the same value
182             if(preset.contains("duration")){
183                 QVariant duration = preset.value("duration");
184                 preset["repeat"] = duration;
185                 preset["kprepeat"] = duration;
```

```
186             }
187             _presets.append(name);
188             _presetValues.append(preset);
189         }
190     }
191     // Make sure the required parameters are filled out
192     if(_info.guid.isNull() || _info.name == "" || _info.version == "" ||
    _info.year == "" || _info.author == "" || _info.license == "")
193         return false;
194     // Add timing parameters
195     double defaultDuration = -1.;
196     if(!_info.absoluteTime){
197         defaultDuration = 1.;
198         Param duration = { Param::DOUBLE, "duration", "", "", defaultDuration, 0.1,
    ONE_DAY };
199         _info.params.append(duration);
200     }
201     Param trigger = { Param::BOOL, "trigger", "", "", true, 0, 0 };
202     _info.params.append(trigger);
203     Param kptrigger = { Param::BOOL, "kptrigger", "", "", false, 0, 0 };
204     _info.params.append(kptrigger);
205     if(_info.absoluteTime || !_info.repeat)
206         _info.preempt = false;
207     Param kpmode = { Param::LONG, "kpmode", "", "", 1, 0, 0 };
208     if(_info.kpMode)
209         kpmode.defaultValue = 0;
210     _info.params.append(kpmode);
211     Param delay = { Param::DOUBLE, "delay", "", "", 0., 0., ONE_DAY };
212     Param kpdelay = { Param::DOUBLE, "kpdelay", "", "", 0., 0.,
    ONE_DAY };
213     _info.params.append(delay);
214     _info.params.append(kpdelay);
215     Param kpmodestop = { Param::BOOL, "kpmodestop", "", "", false, 0, 0 };
216     Param kprelease = { Param::BOOL, "kprelease", "", "", false, 0, 0 };
217     _info.params.append(kpmodestop);
218     _info.params.append(kprelease);
219     if(_info.repeat){
220         Param repeat = { Param::DOUBLE, "repeat", "", "", defaultDuration, 0.1,
    ONE_DAY };
221         Param kprepeat = { Param::DOUBLE, "kprepeat", "", "", defaultDuration, 0.1,
    ONE_DAY };
222         // When repeats are enabled, stop and kpstop are LONG values (number of repeats)
223         Param stop = { Param::LONG, "stop", "", "", -1, 0, 1000 };
224         Param kpstop = { Param::LONG, "kpstop", "", "", 0, 0, 1000 };
225         _info.params.append(repeat);
226         _info.params.append(kprepeat);
227         _info.params.append(stop);
228         _info.params.append(kpstop);
229     } else {
230         // When repeats are disabled, stop and kpstop are DOUBLE values (seconds)
231         Param stop = { Param::DOUBLE, "stop", "", "", -1., 0.1,
    ONE_DAY };
232         Param kpstop = { Param::DOUBLE, "kpstop", "", "", -1., 0.1,
    ONE_DAY };
233         _info.params.append(stop);
234         _info.params.append(kpstop);
235     }
236     return true;
237 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.20 const QString& AnimScript::name ( ) const [inline]

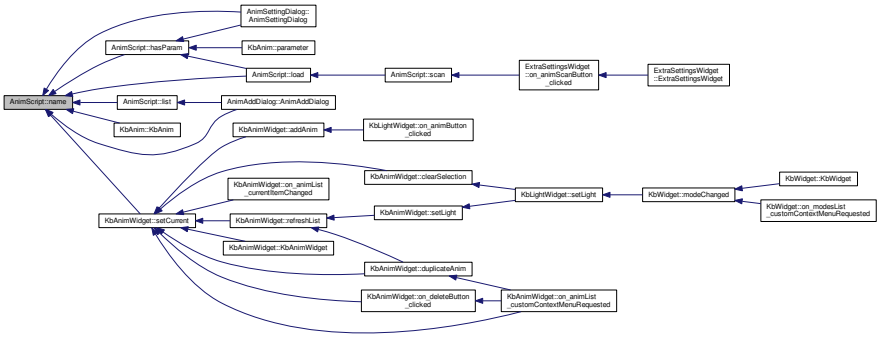
Definition at line 60 of file animscript.h.

References _info.

Referenced by AnimAddDialog::AnimAddDialog(), AnimSettingDialog::AnimSettingDialog(), hasParam(), KbAnim::-KbAnim(), list(), load(), and KbAnimWidget::setCurrent().

```
60 { return _info.name; }
```

Here is the caller graph for this function:



12.3.5.21 Param AnimScript::param (const QString & *name*) const [inline]

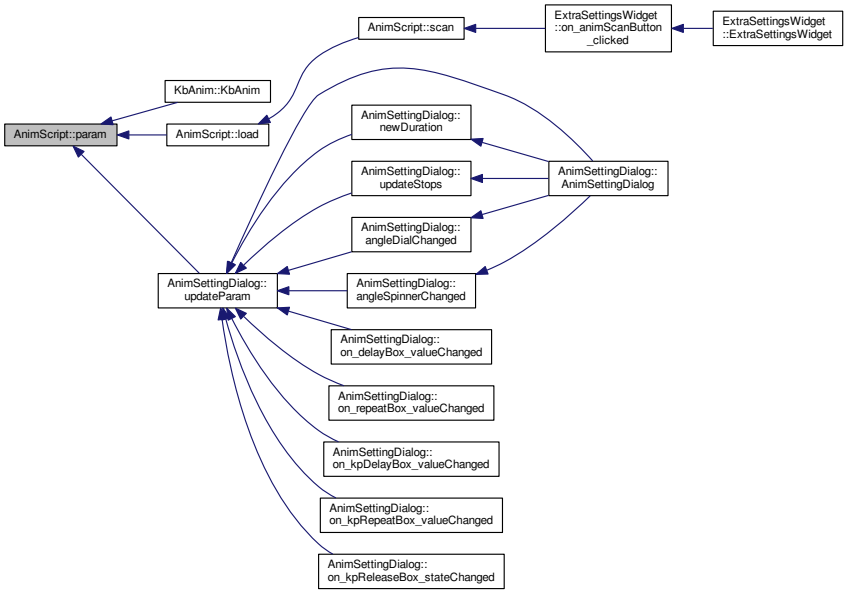
Definition at line 73 of file animscript.h.

References _info, AnimScript::Param::INVALID, and AnimScript::Param::name.

Referenced by KbAnim::KbAnim(), load(), and AnimSettingDialog::updateParam().

```
73 { QListIterator<Param> i(_info.params); while(i.hasNext()){ Param p = i.next(); if(p.name ==
     name) return p; } return ((Param[]){ { Param::INVALID, "", "", "", 0, 0, 0 } })[0]; }
```

Here is the caller graph for this function:



12.3.5.22 void AnimScript::parameters (const QMap< QString, QVariant > & *paramValues*)

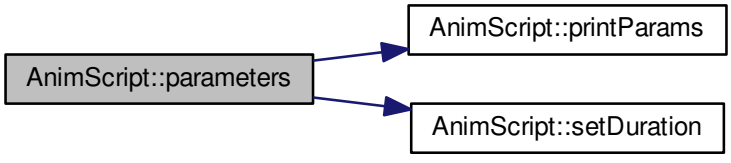
Definition at line 265 of file animscript.cpp.

References _info, _paramValues, initialized, printParams(), process, and setDuration().

Referenced by KbAnim::updateParams().

```
265                                                                    {
266     if(!initialized || !process || !_info.liveParams)
267         return;
268     _paramValues = paramValues;
269     setDuration();
270     printParams();
271 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.3.5.23 QListIterator<Param> AnimScript::paramIterator ( ) const [inline]

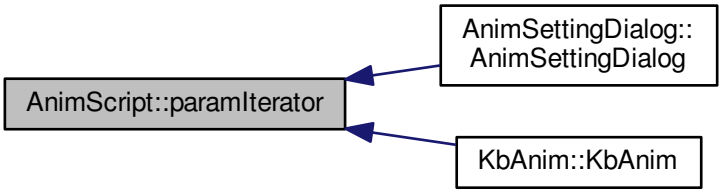
Definition at line 72 of file animscript.h.

References _info.

Referenced by AnimSettingDialog::AnimSettingDialog(), and KbAnim::KbAnim().

```
72 { return _info.params; }
```

Here is the caller graph for this function:



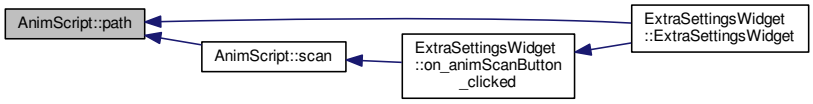
12.3.5.24 QString AnimScript::path ( ) [static]

Definition at line 29 of file animscript.cpp.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and scan().

```
29                       {
30 #ifdef __APPLE__
31     return QDir(QApplication::applicationDirPath() + "/../Resources").absoluteFilePath("ckb-animations");
32 #else
33     return QDir("/usr/lib").absoluteFilePath("ckb-animations");
34 #endif
35 }
```

Here is the caller graph for this function:



12.3.5.25 const PresetValue& AnimScript::preset (int *index*) const [inline]

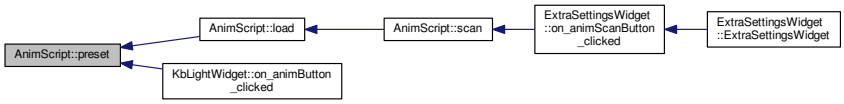
Definition at line 69 of file animscript.h.

References _presetValues.

Referenced by load(), and KbLightWidget::on_animButton_clicked().

```
69 { return _presetValues[index]; }
```

Here is the caller graph for this function:



12.3.5.26 const QStringList& AnimScript::presets ( ) const [inline]

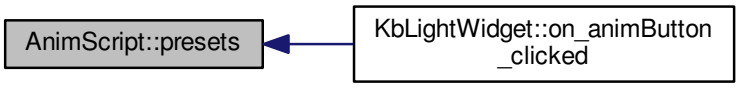
Definition at line 68 of file animscript.h.

References _presets.

Referenced by KbLightWidget::on_animButton_clicked().

```
68 { return _presets; }
```

Here is the caller graph for this function:



12.3.5.27 void AnimScript::printParams ( ) [private]

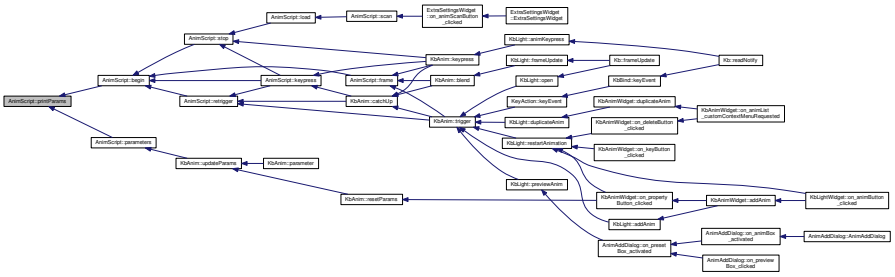
Definition at line 273 of file animscript.cpp.

References _paramValues, and process.

Referenced by begin(), and parameters().

```
273                             {
274     process->write("begin params\n");
275     QMapIterator<QString, QVariant> i(_paramValues);
276     while(i.hasNext()){
277         i.next();
278         process->write("param ");
279         process->write(i.key().toLatin1());
280         process->write(" ");
281         process->write(QUrl::toPercentEncoding(i.value().toString()));
282         process->write("\n");
283     }
284     process->write("end params\n");
285 }
```

Here is the caller graph for this function:



12.3.5.28 void AnimScript::readProcess ( ) [private], [slot]

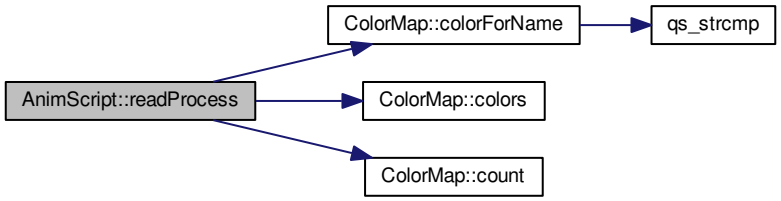
Definition at line 397 of file animscript.cpp.

References _colorBuffer, _colors, ColorMap::colorForName(), ColorMap::colors(), ColorMap::count(), inFrame, process, readAnyFrame, readFrame, and stopped.

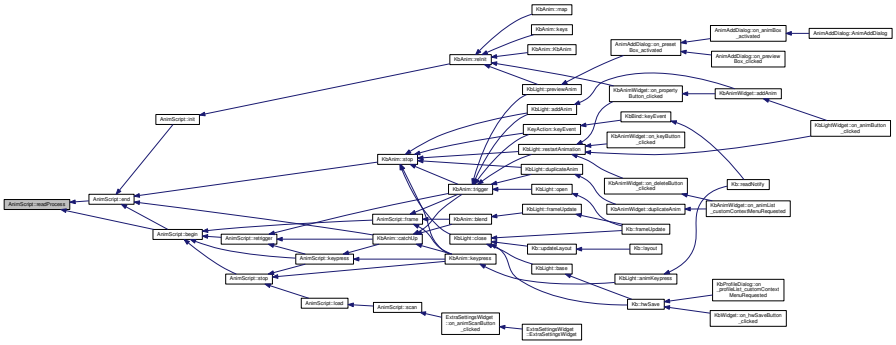
Referenced by begin(), and end().

```
397                             {
398     while(process->canReadLine()){
399         QByteArray line = process->readLine().trimmed();
400         if(!inFrame){
401             // Ignore anything not between "begin frame" and "end frame", except for "end run", which
      indicates that the program is done.
402             if(line == "begin frame")
403                 inFrame = true;
404             else if(line == "end run"){
405                 stopped = true;
406                 return;
407             }
408             continue;
409         }
410         if(line.startsWith("argb ")){
411             // Add a color to the buffer
412             char keyName[31];
413             QRgb keyColor = 0;
414             if(sscanf(line, "argb %30s %x", keyName, &keyColor) != 2)
415                 continue;
416             QRgb* inMap = _colorBuffer.colorForName(keyName);
417             if(!inMap)
418                 continue;
419             *inMap = keyColor;
420         }
421         if(line == "end frame"){
422             // Frame is finished. Copy color buffer back to the atomic map
423             memcpy(_colors.colors(), _colorBuffer.
      colors(), sizeof(QRgb) * _colors.count());
424             inFrame = false;
425             readFrame = readAnyFrame = true;
426         }
427     }
428 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.29 void AnimScript::retrigger (quint64 *timestamp,* bool *allowPreempt* = `false`)

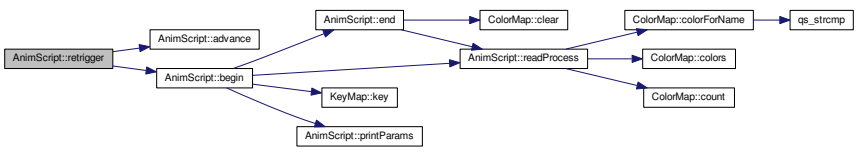
Definition at line 331 of file animscript.cpp.

References _info, advance(), begin(), initialized, process, and repeatMsec.

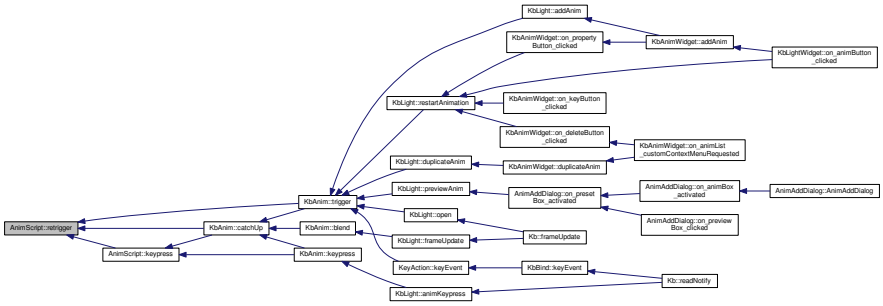
Referenced by KbAnim::catchUp(), keypress(), and KbAnim::trigger().

```
331                                                          {
332     if(!initialized)
333         return;
334     if(allowPreempt && _info.preempt && repeatMsec > 0)
335         // If preemption is wanted, trigger the animation 1 duration in the past first
336         retrigger(timestamp - repeatMsec);
337     if(!process)
338         begin(timestamp);
339     advance(timestamp);
340     if(process)
341         process->write("start\n");
342 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.30 void AnimScript::scan ( ) [static]

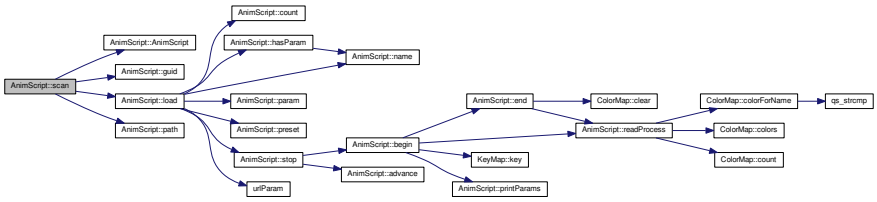
Definition at line 37 of file animscript.cpp.

References _info, AnimScript(), guid(), load(), path(), and scripts.

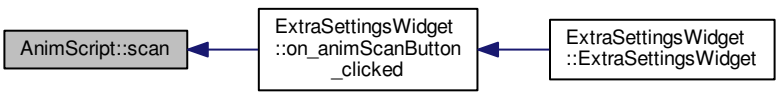
Referenced by ExtraSettingsWidget::on_animScanButton_clicked().

```
37                     {
38     QDir dir(path());
39     foreach(AnimScript* script, scripts)
40         delete script;
41     scripts.clear();
42     foreach(QString file, dir.entryList(QDir::Files | QDir::Executable)){
43         AnimScript* script = new AnimScript(qApp, dir.absoluteFilePath(file));
44         if(script->load() && !scripts.contains(script->_info.
       guid))
45             scripts[script->_info.guid] = script;
46         else
47             delete script;
48     }
49 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.31 void AnimScript::setDuration ( ) [private]

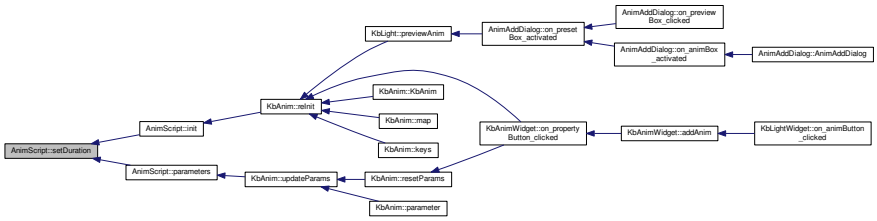
Definition at line 253 of file animscript.cpp.

References _info, _paramValues, durationMsec, and repeatMsec.

Referenced by init(), and parameters().

```
253                             {
254     if(_info.absoluteTime){
255         durationMsec = 1000;
256         repeatMsec = 0;
257     } else {
258         durationMsec = round(_paramValues.value("duration").toDouble() * 1000.);
259         if(durationMsec <= 0)
260             durationMsec = -1;
261         repeatMsec = round(_paramValues.value("repeat").toDouble() * 1000.);
262     }
263 }
```

Here is the caller graph for this function:



12.3.5.32 void AnimScript::stop (quint64 *timestamp*)

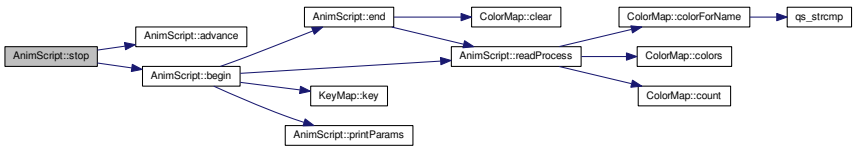
Definition at line 344 of file animscript.cpp.

References advance(), begin(), initialized, and process.

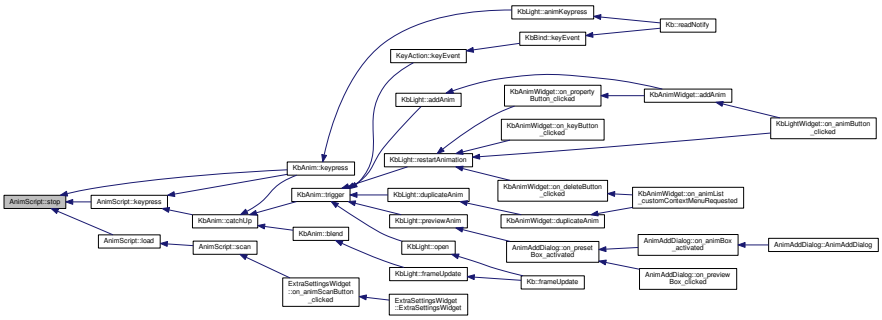
Referenced by KbAnim::keypress(), keypress(), and load().

```
344                                     {
345     if(!initialized)
346         return;
347     if(!process)
348         begin(timestamp);
349     advance(timestamp);
350     if(process)
351         process->write("stop\n");
352 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.3.5.33 const QString& AnimScript::version ( ) const [inline]

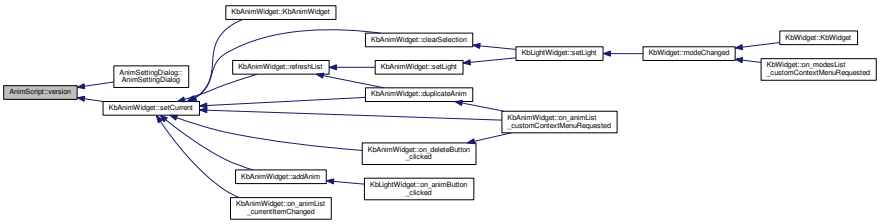
Definition at line 61 of file animscript.h.

References _info.

Referenced by AnimSettingDialog::AnimSettingDialog(), and KbAnimWidget::setCurrent().

```
61 { return _info.version; }
```

Here is the caller graph for this function:



12.3.5.34 const QString& AnimScript::year ( ) const [inline]

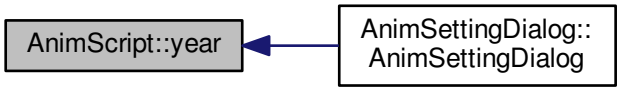
Definition at line 63 of file animscript.h.

References _info.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
63 { return _info.year; }
```

Here is the caller graph for this function:



12.3.6 Field Documentation

12.3.6.1 ColorMap AnimScript::_colorBuffer [private]

Definition at line 143 of file animscript.h.

Referenced by init(), and readProcess().

12.3.6.2 ColorMap AnimScript::_colors [private]

Definition at line 135 of file animscript.h.

Referenced by colors(), end(), init(), and readProcess().

12.3.6.3 struct { ... } AnimScript::_info [private]

Referenced by advance(), author(), copyright(), description(), guid(), hasKeypress(), hasParam(), keypress(), license(), list(), load(), name(), param(), parameters(), paramIterator(), retrigger(), scan(), setDuration(), version(), and year().

12.3.6.4 QStringList AnimScript::_keys [private]

Definition at line 133 of file animscript.h.

Referenced by begin(), and init().

12.3.6.5 KeyMap AnimScript::_map [private]

Definition at line 130 of file animscript.h.

Referenced by begin(), init(), and keypress().

12.3.6.6 PresetValue AnimScript::_paramValues [private]

Definition at line 136 of file animscript.h.

Referenced by init(), keypress(), parameters(), printParams(), and setDuration().

12.3.6.7 QString AnimScript::_path [private]

Definition at line 128 of file animscript.h.

Referenced by begin(), init(), and load().

12.3.6.8 QStringList AnimScript::_presets [private]

Definition at line 125 of file animscript.h.

Referenced by load(), and presets().

12.3.6.9 QList<PresetValue> AnimScript::_presetValues [private]

Definition at line 126 of file animscript.h.

Referenced by load(), and preset().

12.3.6.10 int AnimScript::durationMsec [private]

Definition at line 140 of file animscript.h.

Referenced by advance(), and setDuration().

12.3.6.11 bool AnimScript::firstFrame [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), frame(), and init().

12.3.6.12 bool AnimScript::inFrame [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), and readProcess().

12.3.6.13 bool AnimScript::initialized [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), frame(), hasFrame(), init(), keypress(), parameters(), retrigger(), and stop().

12.3.6.14 const int AnimScript::KP_NAME = 1 [static], [private]

Definition at line 124 of file animscript.h.

Referenced by keypress(), and load().

12.3.6.15 const int AnimScript::KP_NONE = 0 [static], [private]

Definition at line 124 of file animscript.h.

Referenced by hasKeypress(), keypress(), and load().

12.3.6.16 const int AnimScript::KP_POSITION = 2 [static], [private]

Definition at line 124 of file animscript.h.

Referenced by keypress(), and load().

12.3.6.17 quint64 AnimScript::lastFrame [private]

Definition at line 139 of file animscript.h.

Referenced by advance(), and begin().

12.3.6.18 int AnimScript::minX [private]

Definition at line 131 of file animscript.h.

Referenced by begin(), and keypress().

12.3.6.19 int AnimScript::minY [private]

Definition at line 131 of file animscript.h.

Referenced by begin(), and keypress().

12.3.6.20 QProcess* AnimScript::process [private]

Definition at line 142 of file animscript.h.

Referenced by advance(), begin(), end(), frame(), keypress(), parameters(), printParams(), readProcess(), retrigger(), stop(), and ~AnimScript().

12.3.6.21 bool AnimScript::readAnyFrame [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), hasFrame(), and readProcess().

12.3.6.22 bool AnimScript::readFrame [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), frame(), and readProcess().

12.3.6.23 int AnimScript::repeatMsec [private]

Definition at line 140 of file animscript.h.

Referenced by retrigger(), and setDuration().

12.3.6.24 QHash< QUuid, AnimScript * > AnimScript::scripts [static], [private]

Definition at line 152 of file animscript.h.

Referenced by copy(), count(), list(), and scan().

12.3.6.25 bool AnimScript::stopped [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), frame(), init(), and readProcess().

The documentation for this class was generated from the following files:

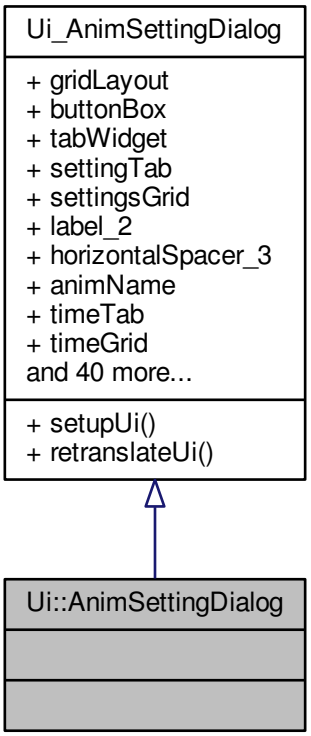
- src/ckb/animscript.h

- src/ckb/animscript.cpp

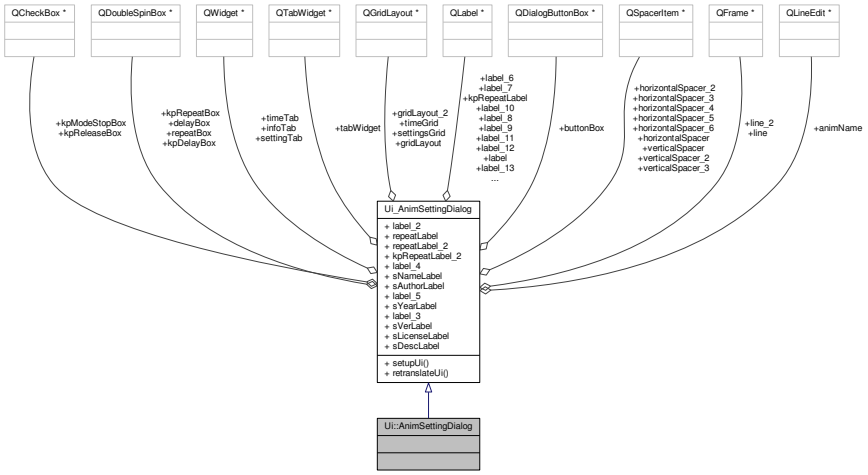
12.4 Ui::AnimSettingDialog Class Reference

`#include <src/ckb/ui_animsettingdialog.h>`

Inheritance diagram for Ui::AnimSettingDialog:



Collaboration diagram for Ui::AnimSettingDialog:



Public Member Functions

- void setupUi (QDialog *AnimSettingDialog)
- void retranslateUi (QDialog *AnimSettingDialog)

Data Fields

- QGridLayout * gridLayout
- QDialogButtonBox * buttonBox
- QTabWidget * tabWidget
- QWidget * settingTab
- QGridLayout * settingsGrid
- QLabel * label_2
- QSpacerItem * horizontalSpacer_3
- QLineEdit * animName
- QWidget * timeTab
- QGridLayout * timeGrid
- QSpacerItem * horizontalSpacer_5
- QLabel * kpRepeatLabel
- QLabel * repeatLabel
- QDoubleSpinBox * repeatBox
- QLabel * repeatLabel_2
- QFrame * line
- QLabel * kpRepeatLabel_2
- QLabel * label_4
- QLabel * label_8
- QDoubleSpinBox * kpDelayBox
- QLabel * label_13
- QLabel * label_12
- QDoubleSpinBox * delayBox
- QCheckBox * kpModeStopBox
- QLabel * label_10
- QSpacerItem * verticalSpacer_3
- QSpacerItem * horizontalSpacer_4

- QCheckBox * kpReleaseBox
- QLabel * label_6
- QDoubleSpinBox * kpRepeatBox
- QSpacerItem * verticalSpacer_2
- QFrame * line_2
- QSpacerItem * horizontalSpacer_6
- QWidget * infoTab
- QGridLayout * gridLayout_2
- QLabel * sNameLabel
- QLabel * label_7
- QLabel * label_11
- QLabel * sAuthorLabel
- QLabel * label_9
- QLabel * label_5
- QLabel * sYearLabel
- QLabel * label_3
- QSpacerItem * horizontalSpacer
- QSpacerItem * horizontalSpacer_2
- QLabel * sVerLabel
- QLabel * sLicenseLabel
- QLabel * label
- QSpacerItem * verticalSpacer
- QLabel * sDescLabel

12.4.1 Detailed Description

Definition at line 392 of file ui_animsettingdialog.h.

12.4.2 Member Function Documentation

12.4.2.1 void Ui_AnimSettingDialog::retranslateUi (QDialog * *AnimSettingDialog*) [inline], [inherited]

Definition at line 356 of file ui_animsettingdialog.h.

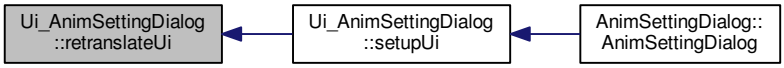
References Ui_AnimSettingDialog::infoTab, Ui_AnimSettingDialog::kpModeStopBox, Ui_AnimSettingDialog::kpReleaseBox, Ui_AnimSettingDialog::kpRepeatLabel, Ui_AnimSettingDialog::kpRepeatLabel_2, Ui_AnimSettingDialog::label, Ui_AnimSettingDialog::label_10, Ui_AnimSettingDialog::label_11, Ui_AnimSettingDialog::label_12, Ui_AnimSettingDialog::label_13, Ui_AnimSettingDialog::label_2, Ui_AnimSettingDialog::label_3, Ui_AnimSettingDialog::label_4, Ui_AnimSettingDialog::label_5, Ui_AnimSettingDialog::label_6, Ui_AnimSettingDialog::label_7, Ui_AnimSettingDialog::label_8, Ui_AnimSettingDialog::label_9, Ui_AnimSettingDialog::repeatLabel, Ui_AnimSettingDialog::repeatLabel_2, Ui_AnimSettingDialog::sAuthorLabel, Ui_AnimSettingDialog::sDescLabel, Ui_AnimSettingDialog::settingTab, Ui_AnimSettingDialog::sLicenseLabel, Ui_AnimSettingDialog::sNameLabel, Ui_AnimSettingDialog::sVerLabel, Ui_AnimSettingDialog::sYearLabel, Ui_AnimSettingDialog::tabWidget, and Ui_AnimSettingDialog::timeTab.

Referenced by Ui_AnimSettingDialog::setupUi().

```
357     {
358         AnimSettingDialog->setWindowTitle(QApplication::translate("AnimSettingDialog", "Dialog", Q_NULLPTR)
    );
359         label_2->setText(QApplication::translate("AnimSettingDialog", "Name:", Q_NULLPTR));
360         tabWidget->setTabText(tabWidget->indexOf(settingTab),
    QApplication::translate("AnimSettingDialog", "Settings", Q_NULLPTR));
361         kpRepeatLabel->setText(QApplication::translate("AnimSettingDialog", "Repeat every:",
    Q_NULLPTR));
362         repeatLabel->setText(QApplication::translate("AnimSettingDialog", "Repeat every:",
    Q_NULLPTR));
363         repeatLabel_2->setText(QApplication::translate("AnimSettingDialog", "seconds",
    Q_NULLPTR));
364         kpRepeatLabel_2->setText(QApplication::translate("AnimSettingDialog", "seconds",
```

```
    Q_NULLPTR));
365         label_4->setText(QApplication::translate("AnimSettingDialog", "<b>On mode start</b>",
    Q_NULLPTR));
366         label_8->setText(QApplication::translate("AnimSettingDialog", "seconds", Q_NULLPTR));
367         label_13->setText(QApplication::translate("AnimSettingDialog", "seconds", Q_NULLPTR));
368         label_12->setText(QApplication::translate("AnimSettingDialog", "Start delay:", Q_NULLPTR));
369         kpModeStopBox->setText(QApplication::translate("AnimSettingDialog", "Stop on key press
    ", Q_NULLPTR));
370         label_10->setText(QApplication::translate("AnimSettingDialog", "Start delay:", Q_NULLPTR));
371         kpReleaseBox->setText(QApplication::translate("AnimSettingDialog", "Stop on key release
    ", Q_NULLPTR));
372         label_6->setText(QApplication::translate("AnimSettingDialog", "<b>On key press</b>",
    Q_NULLPTR));
373         tabWidget->setTabText(tabWidget->indexOf(timeTab), QApplication::translate
    ("AnimSettingDialog", "Timing", Q_NULLPTR));
374         sNameLabel->setText(QApplication::translate("AnimSettingDialog", "Name", Q_NULLPTR));
375         label_7->setText(QApplication::translate("AnimSettingDialog", "Year:", Q_NULLPTR));
376         label_11->setText(QApplication::translate("AnimSettingDialog", "Description:", Q_NULLPTR));
377         sAuthorLabel->setText(QApplication::translate("AnimSettingDialog", "Author", Q_NULLPTR)
    );
378         label_9->setText(QApplication::translate("AnimSettingDialog", "License:", Q_NULLPTR));
379         label_5->setText(QApplication::translate("AnimSettingDialog", "Version:", Q_NULLPTR));
380         sYearLabel->setText(QApplication::translate("AnimSettingDialog", "Year", Q_NULLPTR));
381         label_3->setText(QApplication::translate("AnimSettingDialog", "Author:", Q_NULLPTR));
382         sVerLabel->setText(QApplication::translate("AnimSettingDialog", "Version", Q_NULLPTR));
383         sLicenseLabel->setText(QApplication::translate("AnimSettingDialog", "License",
    Q_NULLPTR));
384         label->setText(QApplication::translate("AnimSettingDialog", "Script name:", Q_NULLPTR));
385         sDescLabel->setText(QApplication::translate("AnimSettingDialog", "Description", Q_NULLPTR
    ));
386         tabWidget->setTabText(tabWidget->indexOf(infoTab), QApplication::translate
    ("AnimSettingDialog", "About", Q_NULLPTR));
387     } // retranslateUi
```

Here is the caller graph for this function:



12.4.2.2 void Ui_AnimSettingDialog::setupUi (QDialog * *AnimSettingDialog*) [inline], [inherited]

Definition at line 85 of file ui_animsettingdialog.h.

References Ui_AnimSettingDialog::animName, Ui_AnimSettingDialog::buttonBox, Ui_AnimSettingDialog::delayBox, Ui_AnimSettingDialog::gridLayout, Ui_AnimSettingDialog::gridLayout_2, Ui_AnimSettingDialog::horizontalSpacer, Ui_AnimSettingDialog::horizontalSpacer_2, Ui_AnimSettingDialog::horizontalSpacer_3, Ui_AnimSettingDialog::horizontalSpacer_4, Ui_AnimSettingDialog::horizontalSpacer_5, Ui_AnimSettingDialog::horizontalSpacer_6, Ui_AnimSettingDialog::infoTab, Ui_AnimSettingDialog::kpDelayBox, Ui_AnimSettingDialog::kpModeStopBox, Ui_AnimSettingDialog::kpReleaseBox, Ui_AnimSettingDialog::kpRepeatBox, Ui_AnimSettingDialog::kpRepeatLabel, Ui_AnimSettingDialog::kpRepeatLabel_2, Ui_AnimSettingDialog::label, Ui_AnimSettingDialog::label_10, Ui_AnimSettingDialog::label_11, Ui_AnimSettingDialog::label_12, Ui_AnimSettingDialog::label_13, Ui_AnimSettingDialog::label_2, Ui_AnimSettingDialog::label_3, Ui_AnimSettingDialog::label_4, Ui_AnimSettingDialog::label_5, Ui_AnimSettingDialog::label_6, Ui_AnimSettingDialog::label_7, Ui_AnimSettingDialog::label_8, Ui_AnimSettingDialog::label_9, Ui_AnimSettingDialog::line, Ui_AnimSettingDialog::line_2, Ui_AnimSettingDialog::repeatBox, Ui_AnimSettingDialog::repeatLabel, Ui_AnimSettingDialog::repeatLabel_2, Ui_AnimSettingDialog::retranslateUi(), Ui_AnimSettingDialog::sAuthorLabel, Ui_AnimSettingDialog::sDescLabel, Ui_AnimSettingDialog::settingsGrid, Ui_AnimSettingDialog::settingTab, Ui_AnimSettingDialog::sLicenseLabel, Ui_AnimSettingDialog::sNameLabel, Ui_AnimSettingDialog::sVerLabel, Ui_AnimSettingDialog::sYearLabel, Ui_AnimSettingDialog::tabWidget, Ui_AnimSettingDialog::timeGrid, Ui_AnimSettingDialog::timeTab, Ui_AnimSettingDialog::verticalSpacer, Ui_AnimSettingDialog::verticalSpacer_2, and Ui_AnimSettingDialog::verticalSpacer_3.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
86      {
87          if (AnimSettingDialog->objectName().isEmpty())
88              AnimSettingDialog->setObjectName(QStringLiteral("AnimSettingDialog"));
89          AnimSettingDialog->resize(480, 457);
90          AnimSettingDialog->setMinimumSize(QSize(480, 0));
91          AnimSettingDialog->setFocusPolicy(Qt::ClickFocus);
92          gridLayout = new QGridLayout(AnimSettingDialog);
93          gridLayout->setObjectName(QStringLiteral("gridLayout"));
94          buttonBox = new QDialogButtonBox(AnimSettingDialog);
95          buttonBox->setObjectName(QStringLiteral("buttonBox"));
96          buttonBox->setOrientation(Qt::Horizontal);
97          buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
98
99          gridLayout->addWidget(buttonBox, 2, 0, 1, 1);
100
101          tabWidget = new QTabWidget(AnimSettingDialog);
102          tabWidget->setObjectName(QStringLiteral("tabWidget"));
103          settingTab = new QWidget();
104          settingTab->setObjectName(QStringLiteral("settingTab"));
105          settingsGrid = new QGridLayout(settingTab);
106          settingsGrid->setSpacing(12);
107          settingsGrid->setObjectName(QStringLiteral("settingsGrid"));
108          settingsGrid->setContentsMargins(12, 12, 12, 12);
109          label_2 = new QLabel(settingTab);
110          label_2->setObjectName(QStringLiteral("label_2"));
111
112          settingsGrid->addWidget(label_2, 0, 1, 1, 1);
113
114          horizontalSpacer_3 = new QSpacerItem(5, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
115
116          settingsGrid->addItem(horizontalSpacer_3, 0, 2, 1, 1);
117
118          animName = new QLineEdit(settingTab);
119          animName->setObjectName(QStringLiteral("animName"));
120
121          settingsGrid->addWidget(animName, 0, 3, 1, 3);
122
123          tabWidget->addTab(settingTab, QString());
124          timeTab = new QWidget();
125          timeTab->setObjectName(QStringLiteral("timeTab"));
126          timeGrid = new QGridLayout(timeTab);
127          timeGrid->setSpacing(12);
128          timeGrid->setObjectName(QStringLiteral("timeGrid"));
129          timeGrid->setContentsMargins(12, 12, 12, 12);
130          horizontalSpacer_5 = new QSpacerItem(15, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
131
132          timeGrid->addItem(horizontalSpacer_5, 3, 2, 1, 1);
133
134          kpRepeatLabel = new QLabel(timeTab);
135          kpRepeatLabel->setObjectName(QStringLiteral("kpRepeatLabel"));
136
137          timeGrid->addWidget(kpRepeatLabel, 11, 1, 1, 1);
138
139          repeatLabel = new QLabel(timeTab);
140          repeatLabel->setObjectName(QStringLiteral("repeatLabel"));
141
142          timeGrid->addWidget(repeatLabel, 3, 1, 1, 1);
143
144          repeatBox = new QDoubleSpinBox(timeTab);
145          repeatBox->setObjectName(QStringLiteral("repeatBox"));
146          repeatBox->setDecimals(1);
147          repeatBox->setMinimum(0.1);
148          repeatBox->setMaximum(86400);
149
150          timeGrid->addWidget(repeatBox, 3, 3, 1, 1);
151
152          repeatLabel_2 = new QLabel(timeTab);
153          repeatLabel_2->setObjectName(QStringLiteral("repeatLabel_2"));
154
155          timeGrid->addWidget(repeatLabel_2, 3, 4, 1, 1);
156
157          line = new QFrame(timeTab);
158          line->setObjectName(QStringLiteral("line"));
159          line->setFrameShape(QFrame::HLine);
160          line->setFrameShadow(QFrame::Sunken);
161
162          timeGrid->addWidget(line, 1, 0, 1, 6);
163
164          kpRepeatLabel_2 = new QLabel(timeTab);
165          kpRepeatLabel_2->setObjectName(QStringLiteral("kpRepeatLabel_2"));
166
167          timeGrid->addWidget(kpRepeatLabel_2, 11, 4, 1, 1);
168
169          label_4 = new QLabel(timeTab);
170          label_4->setObjectName(QStringLiteral("label_4"));
```

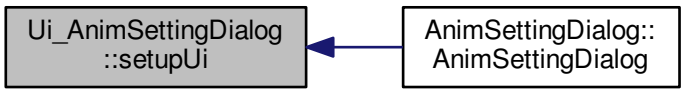
```
171
172         timeGrid->addWidget(label_4, 0, 0, 1, 6);
173
174         label_8 = new QLabel(timeTab);
175         label_8->setObjectName(QStringLiteral("label_8"));
176
177         timeGrid->addWidget(label_8, 2, 4, 1, 1);
178
179         kpDelayBox = new QDoubleSpinBox(timeTab);
180         kpDelayBox->setObjectName(QStringLiteral("kpDelayBox"));
181         kpDelayBox->setDecimals(1);
182         kpDelayBox->setMaximum(86400);
183
184         timeGrid->addWidget(kpDelayBox, 10, 3, 1, 1);
185
186         label_13 = new QLabel(timeTab);
187         label_13->setObjectName(QStringLiteral("label_13"));
188
189         timeGrid->addWidget(label_13, 10, 4, 1, 1);
190
191         label_12 = new QLabel(timeTab);
192         label_12->setObjectName(QStringLiteral("label_12"));
193
194         timeGrid->addWidget(label_12, 10, 1, 1, 1);
195
196         delayBox = new QDoubleSpinBox(timeTab);
197         delayBox->setObjectName(QStringLiteral("delayBox"));
198         delayBox->setDecimals(1);
199         delayBox->setMaximum(86400);
200
201         timeGrid->addWidget(delayBox, 2, 3, 1, 1);
202
203         kpModeStopBox = new QCheckBox(timeTab);
204         kpModeStopBox->setObjectName(QStringLiteral("kpModeStopBox"));
205
206         timeGrid->addWidget(kpModeStopBox, 5, 1, 1, 3);
207
208         label_10 = new QLabel(timeTab);
209         label_10->setObjectName(QStringLiteral("label_10"));
210
211         timeGrid->addWidget(label_10, 2, 1, 1, 1);
212
213         verticalSpacer_3 = new QSpacerItem(0, 10, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
214
215         timeGrid->addItem(verticalSpacer_3, 7, 1, 1, 1);
216
217         horizontalSpacer_4 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
218
219         timeGrid->addItem(horizontalSpacer_4, 2, 5, 1, 1);
220
221         kpReleaseBox = new QCheckBox(timeTab);
222         kpReleaseBox->setObjectName(QStringLiteral("kpReleaseBox"));
223
224         timeGrid->addWidget(kpReleaseBox, 14, 1, 1, 3);
225
226         label_6 = new QLabel(timeTab);
227         label_6->setObjectName(QStringLiteral("label_6"));
228
229         timeGrid->addWidget(label_6, 8, 0, 1, 6);
230
231         kpRepeatBox = new QDoubleSpinBox(timeTab);
232         kpRepeatBox->setObjectName(QStringLiteral("kpRepeatBox"));
233         kpRepeatBox->setDecimals(1);
234         kpRepeatBox->setMinimum(0.1);
235         kpRepeatBox->setMaximum(86400);
236
237         timeGrid->addWidget(kpRepeatBox, 11, 3, 1, 1);
238
239         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
240
241         timeGrid->addItem(verticalSpacer_2, 15, 1, 1, 1);
242
243         line_2 = new QFrame(timeTab);
244         line_2->setObjectName(QStringLiteral("line_2"));
245         line_2->setFrameShape(QFrame::HLine);
246         line_2->setFrameShadow(QFrame::Sunken);
247
248         timeGrid->addWidget(line_2, 9, 0, 1, 6);
249
250         horizontalSpacer_6 = new QSpacerItem(0, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
251
252         timeGrid->addItem(horizontalSpacer_6, 12, 2, 1, 1);
253
```

```
254         tabWidget->addTab(timeTab, QString());
255         infoTab = new QWidget();
256         infoTab->setObjectName(QStringLiteral("infoTab"));
257         gridLayout_2 = new QGridLayout(infoTab);
258         gridLayout_2->setSpacing(12);
259         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
260         gridLayout_2->setContentsMargins(12, 12, 12, 12);
261         sNameLabel = new QLabel(infoTab);
262         sNameLabel->setObjectName(QStringLiteral("sNameLabel"));
263 
264         gridLayout_2->addWidget(sNameLabel, 0, 2, 1, 1);
265 
266         label_7 = new QLabel(infoTab);
267         label_7->setObjectName(QStringLiteral("label_7"));
268 
269         gridLayout_2->addWidget(label_7, 3, 0, 1, 1);
270 
271         label_11 = new QLabel(infoTab);
272         label_11->setObjectName(QStringLiteral("label_11"));
273 
274         gridLayout_2->addWidget(label_11, 5, 0, 1, 1);
275 
276         sAuthorLabel = new QLabel(infoTab);
277         sAuthorLabel->setObjectName(QStringLiteral("sAuthorLabel"));
278 
279         gridLayout_2->addWidget(sAuthorLabel, 2, 2, 1, 1);
280 
281         label_9 = new QLabel(infoTab);
282         label_9->setObjectName(QStringLiteral("label_9"));
283 
284         gridLayout_2->addWidget(label_9, 4, 0, 1, 1);
285 
286         label_5 = new QLabel(infoTab);
287         label_5->setObjectName(QStringLiteral("label_5"));
288 
289         gridLayout_2->addWidget(label_5, 1, 0, 1, 1);
290 
291         sYearLabel = new QLabel(infoTab);
292         sYearLabel->setObjectName(QStringLiteral("sYearLabel"));
293 
294         gridLayout_2->addWidget(sYearLabel, 3, 2, 1, 1);
295 
296         label_3 = new QLabel(infoTab);
297         label_3->setObjectName(QStringLiteral("label_3"));
298 
299         gridLayout_2->addWidget(label_3, 2, 0, 1, 1);
300 
301         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
302 
303         gridLayout_2->addItem(horizontalSpacer, 0, 3, 1, 1);
304 
305         horizontalSpacer_2 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
306 
307         gridLayout_2->addItem(horizontalSpacer_2, 0, 1, 1, 1);
308 
309         sVerLabel = new QLabel(infoTab);
310         sVerLabel->setObjectName(QStringLiteral("sVerLabel"));
311 
312         gridLayout_2->addWidget(sVerLabel, 1, 2, 1, 1);
313 
314         sLicenseLabel = new QLabel(infoTab);
315         sLicenseLabel->setObjectName(QStringLiteral("sLicenseLabel"));
316 
317         gridLayout_2->addWidget(sLicenseLabel, 4, 2, 1, 1);
318 
319         label = new QLabel(infoTab);
320         label->setObjectName(QStringLiteral("label"));
321 
322         gridLayout_2->addWidget(label, 0, 0, 1, 1);
323 
324         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
325 
326         gridLayout_2->addItem(verticalSpacer, 6, 0, 1, 1);
327 
328         sDescLabel = new QLabel(infoTab);
329         sDescLabel->setObjectName(QStringLiteral("sDescLabel"));
330         sDescLabel->setAlignment(Qt::AlignLeading|Qt::AlignLeft|Qt::AlignTop);
331 
332         gridLayout_2->addWidget(sDescLabel, 5, 2, 2, 2);
333 
334         tabWidget->addTab(infoTab, QString());
335 
336         gridLayout->addWidget(tabWidget, 0, 0, 1, 1);
337 
```

```
338         QWidget::setTabOrder(tabWidget, animName);
339         QWidget::setTabOrder(animName, delayBox);
340         QWidget::setTabOrder(delayBox, repeatBox);
341         QWidget::setTabOrder(repeatBox, kpModeStopBox);
342         QWidget::setTabOrder(kpModeStopBox, kpDelayBox);
343         QWidget::setTabOrder(kpDelayBox, kpRepeatBox);
344         QWidget::setTabOrder(kpRepeatBox, kpReleaseBox);
345 
346         retranslateUi(AnimSettingDialog);
347         QObject::connect(buttonBox, SIGNAL(accepted()), AnimSettingDialog, SLOT(accept()));
348         QObject::connect(buttonBox, SIGNAL(rejected()), AnimSettingDialog, SLOT(reject()));
349 
350         tabWidget->setCurrentIndex(0);
351 
352 
353         QMetaObject::connectSlotsByName(AnimSettingDialog);
354     } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.4.3 Field Documentation

12.4.3.1 QLineEdit* Ui_AnimSettingDialog::animName [inherited]

Definition at line 41 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), AnimSettingDialog::name(), and Ui_AnimSettingDialog-::setupUi().

12.4.3.2 QDialogButtonBox* Ui_AnimSettingDialog::buttonBox [inherited]

Definition at line 35 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.3 QDoubleSpinBox* Ui_AnimSettingDialog::delayBox [inherited]

Definition at line 56 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and Ui_AnimSettingDialog::setupUi().

12.4.3.4 QGridLayout* Ui_AnimSettingDialog::gridLayout [inherited]

Definition at line 34 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.5 QGridLayout* Ui_AnimSettingDialog::gridLayout_2 [inherited]

Definition at line 68 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.6 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer [inherited]

Definition at line 77 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.7 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_2 [inherited]

Definition at line 78 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.8 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_3 [inherited]

Definition at line 40 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.9 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_4 [inherited]

Definition at line 60 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.10 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_5 [inherited]

Definition at line 44 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.11 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_6 [inherited]

Definition at line 66 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.12 QWidget* Ui_AnimSettingDialog::infoTab [inherited]

Definition at line 67 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.13 QDoubleSpinBox* Ui_AnimSettingDialog::kpDelayBox [inherited]

Definition at line 53 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and Ui_AnimSettingDialog::setupUi().

12.4.3.14 QCheckBox* Ui_AnimSettingDialog::kpModeStopBox [inherited]

Definition at line 57 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.15 QCheckBox* Ui_AnimSettingDialog::kpReleaseBox [inherited]

Definition at line 61 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.16 QDoubleSpinBox* Ui_AnimSettingDialog::kpRepeatBox [inherited]

Definition at line 63 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and Ui_AnimSettingDialog::setupUi().

12.4.3.17 QLabel* Ui_AnimSettingDialog::kpRepeatLabel [inherited]

Definition at line 45 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.18 QLabel* Ui_AnimSettingDialog::kpRepeatLabel_2 [inherited]

Definition at line 50 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.19 QLabel* Ui_AnimSettingDialog::label [inherited]

Definition at line 81 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.20 QLabel* Ui_AnimSettingDialog::label_10 [inherited]

Definition at line 58 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.21 QLabel* Ui_AnimSettingDialog::label_11 [inherited]

Definition at line 71 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.22 QLabel* Ui_AnimSettingDialog::label_12 [inherited]

Definition at line 55 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.23 QLabel* Ui_AnimSettingDialog::label_13 [inherited]

Definition at line 54 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.24 QLabel* Ui_AnimSettingDialog::label_2 [inherited]

Definition at line 39 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.25 QLabel* Ui_AnimSettingDialog::label_3 [inherited]

Definition at line 76 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.26 QLabel* Ui_AnimSettingDialog::label_4 [inherited]

Definition at line 51 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.27 QLabel* Ui_AnimSettingDialog::label_5 [inherited]

Definition at line 74 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.28 QLabel* Ui_AnimSettingDialog::label_6 [inherited]

Definition at line 62 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.29 QLabel* Ui_AnimSettingDialog::label_7 [inherited]

Definition at line 70 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.30 QLabel* Ui_AnimSettingDialog::label_8 [inherited]

Definition at line 52 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.31 QLabel* Ui_AnimSettingDialog::label_9 [inherited]

Definition at line 73 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.32 QFrame* Ui_AnimSettingDialog::line [inherited]

Definition at line 49 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.33 QFrame* Ui_AnimSettingDialog::line_2 [inherited]

Definition at line 65 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.34 QDoubleSpinBox* Ui_AnimSettingDialog::repeatBox [inherited]

Definition at line 47 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and Ui_AnimSettingDialog::setupUi().

12.4.3.35 QLabel* Ui_AnimSettingDialog::repeatLabel [inherited]

Definition at line 46 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.36 QLabel* Ui_AnimSettingDialog::repeatLabel_2 [inherited]

Definition at line 48 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.37 QLabel* Ui_AnimSettingDialog::sAuthorLabel [inherited]

Definition at line 72 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.38 QLabel* Ui_AnimSettingDialog::sDescLabel [inherited]

Definition at line 83 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.39 QGridLayout* Ui_AnimSettingDialog::settingsGrid [inherited]

Definition at line 38 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and Ui_AnimSettingDialog::setupUi().

12.4.3.40 QWidget* Ui_AnimSettingDialog::settingTab [inherited]

Definition at line 37 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.41 QLabel* Ui_AnimSettingDialog::sLicenseLabel [inherited]

Definition at line 80 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.42 QLabel* Ui_AnimSettingDialog::sNameLabel [inherited]

Definition at line 69 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.43 QLabel* Ui_AnimSettingDialog::sVerLabel [inherited]

Definition at line 79 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.44 QLabel* Ui_AnimSettingDialog::sYearLabel [inherited]

Definition at line 75 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.45 QTabWidget* Ui_AnimSettingDialog::tabWidget [inherited]

Definition at line 36 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.46 QGridLayout* Ui_AnimSettingDialog::timeGrid [inherited]

Definition at line 43 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and Ui_AnimSettingDialog::setupUi().

12.4.3.47 QWidget* Ui_AnimSettingDialog::timeTab [inherited]

Definition at line 42 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::retranslateUi(), and Ui_AnimSettingDialog::setupUi().

12.4.3.48 QSpacerItem* Ui_AnimSettingDialog::verticalSpacer [inherited]

Definition at line 82 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.49 QSpacerItem* Ui_AnimSettingDialog::verticalSpacer_2 [inherited]

Definition at line 64 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

12.4.3.50 QSpacerItem* Ui_AnimSettingDialog::verticalSpacer_3 [inherited]

Definition at line 59 of file ui_animsettingdialog.h.

Referenced by Ui_AnimSettingDialog::setupUi().

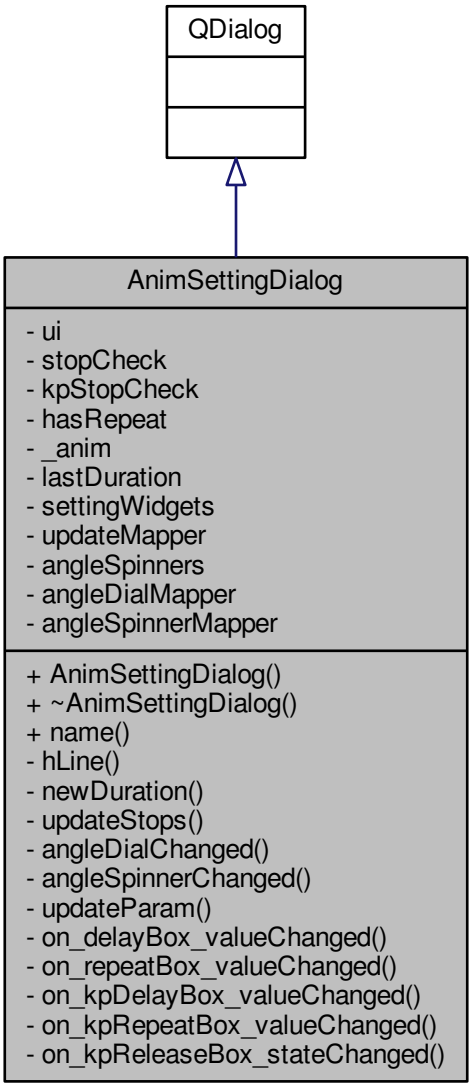
The documentation for this class was generated from the following file:

- src/ckb/ui_animsettingdialog.h

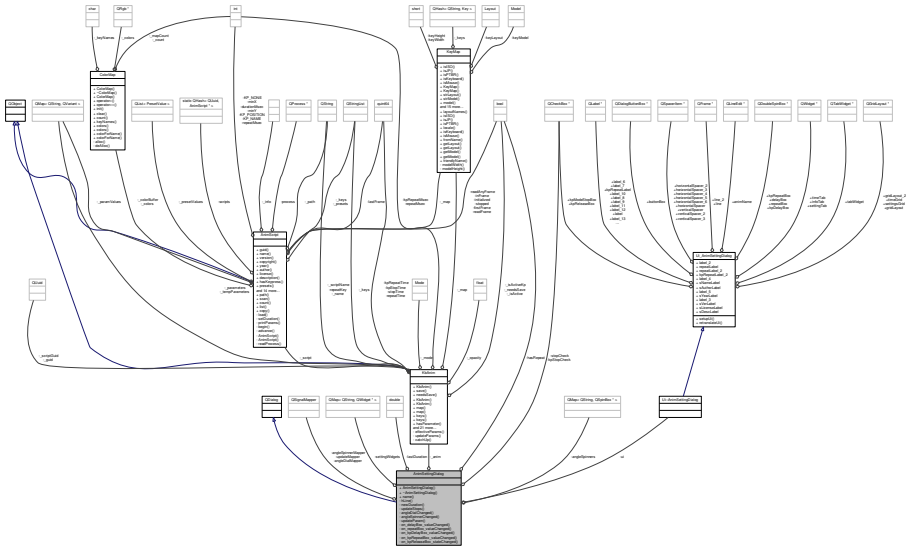
12.5 AnimSettingDialog Class Reference

`#include <src/ckb/animsettingdialog.h>`

Inheritance diagram for AnimSettingDialog:



Collaboration diagram for AnimSettingDialog:



Public Member Functions

- AnimSettingDialog (QWidget ∗parent, KbAnim ∗anim)
- ∼AnimSettingDialog ()
- QString name () const

Private Slots

- void newDuration (double duration)
- void updateStops ()
- void angleDialChanged (QString name)
- void angleSpinnerChanged (QString name)
- void updateParam (QString name)
- void on_delayBox_valueChanged (double arg1)
- void on_repeatBox_valueChanged (double arg1)
- void on_kpDelayBox_valueChanged (double arg1)
- void on_kpRepeatBox_valueChanged (double arg1)
- void on_kpReleaseBox_stateChanged (int arg1)

Private Member Functions

- QFrame ∗ hLine ()

Private Attributes

- Ui::AnimSettingDialog ∗ ui
- QCheckBox ∗ stopCheck
- QCheckBox ∗ kpStopCheck
- bool hasRepeat
- KbAnim ∗ _anim
- double lastDuration
- QMap< QString, QWidget ∗ > settingWidgets

- QSignalMapper updateMapper
- QMap< QString, QSpinBox * > angleSpinners
- QSignalMapper angleDialMapper
- QSignalMapper angleSpinnerMapper

12.5.1 Detailed Description

Definition at line 15 of file animsettingdialog.h.

12.5.2 Constructor & Destructor Documentation

12.5.2.1 AnimSettingDialog::AnimSettingDialog (QWidget * *parent*, KbAnim * *anim*) [explicit]

Definition at line 19 of file animsettingdialog.cpp.

References AnimScript::Param::AGRADIENT, AnimScript::Param::ANGLE, angleDialChanged(), angleDialMapper, angleFlip(), angleSpinnerChanged(), angleSpinnerMapper, angleSpinners, Ui_AnimSettingDialog::animName, AnimScript::Param::ARGB, AnimScript::author(), AnimScript::Param::BOOL, Ui_AnimSettingDialog::delayBox, AnimScript::description(), AnimScript::Param::DOUBLE, AnimScript::Param::GRADIENT, AnimScript::hasKeypress(), AnimScript::hasParam(), KbAnim::hasParameter(), hasRepeat, hLine(), Ui_AnimSettingDialog::kpDelayBox, Ui_AnimSettingDialog::kpModeStopBox, Ui_AnimSettingDialog::kpReleaseBox, Ui_AnimSettingDialog::kpRepeatBox, Ui_AnimSettingDialog::kpRepeatLabel, Ui_AnimSettingDialog::kpRepeatLabel_2, kpStopCheck, AnimScript::Param::LABEL, lastDuration, AnimScript::license(), AnimScript::Param::LONG, AnimScript::Param::maximum, AnimScript::Param::minimum, AnimScript::Param::name, AnimScript::name(), KbAnim::name(), newDuration(), KbAnim::parameter(), AnimScript::paramIterator(), AnimScript::Param::postfix, AnimScript::Param::prefix, Ui_AnimSettingDialog::repeatBox, Ui_AnimSettingDialog::repeatLabel, Ui_AnimSettingDialog::repeatLabel_2, AnimScript::Param::RGB, Ui_AnimSettingDialog::sAuthorLabel, KbAnim::script(), KbAnim::scriptName(), Ui_AnimSettingDialog::sDescLabel, Ui_AnimSettingDialog::settingsGrid, settingWidgets, Ui_AnimSettingDialog::setupUi(), Ui_AnimSettingDialog::sLicenseLabel, Ui_AnimSettingDialog::sNameLabel, stopCheck, AnimScript::Param::STRING, Ui_AnimSettingDialog::sVerLabel, Ui_AnimSettingDialog::sYearLabel, Ui_AnimSettingDialog::timeGrid, AnimScript::Param::type, ui, updateMapper, updateParam(), updateStops(), AnimScript::version(), and AnimScript::year().

```
19                                                                  :
20     QDialog(parent),
21     ui(new Ui::AnimSettingDialog), stopCheck(0),
       kpStopCheck(0),
22     _anim(anim), lastDuration(1.0)
23 {
24     ui->setupUi(this);
25     setWindowTitle(anim->scriptName() + " Animation");
26     ui->animName->setText(anim->name());
27     const AnimScript* script = anim->script();
28
29     connect(&updateMapper, SIGNAL(mapped(QString)), this, SLOT(
       updateParam(QString)));
30     connect(&angleDialMapper, SIGNAL(mapped(QString)), this, SLOT(
       angleDialChanged(QString)));
31     connect(&angleSpinnerMapper, SIGNAL(mapped(QString)), this, SLOT(
       angleSpinnerChanged(QString)));
32
33     // Build settings UI
34     int row = 1;
35     ui->settingsGrid->addItem(new QSpacerItem(0, 10, QSizePolicy::Fixed, QSizePolicy::Fixed),
       row++, 6);
36     ui->settingsGrid->addWidget(new QLabel("<b>Animation</b>", this), row++, 0, 1, 7);
37     ui->settingsGrid->addWidget(hLine(), row++, 0, 1, 7);
38     bool rSpacePlaced = false;
39     QListIterator<AnimScript::Param> i(script->paramIterator());
40     while(i.hasNext()){
41         const AnimScript::Param& param = i.next();
42         // Skip timing/playback params
43         if(param.name == "duration"
44                 || param.name == "trigger" || param.name == "kptrigger"
45                 || param.name == "kpmode"
46                 || param.name == "delay" || param.name == "kpdelay"
47                 || param.name == "repeat" || param.name == "kprepeat"
48                 || param.name == "stop" || param.name == "kpstop"
```

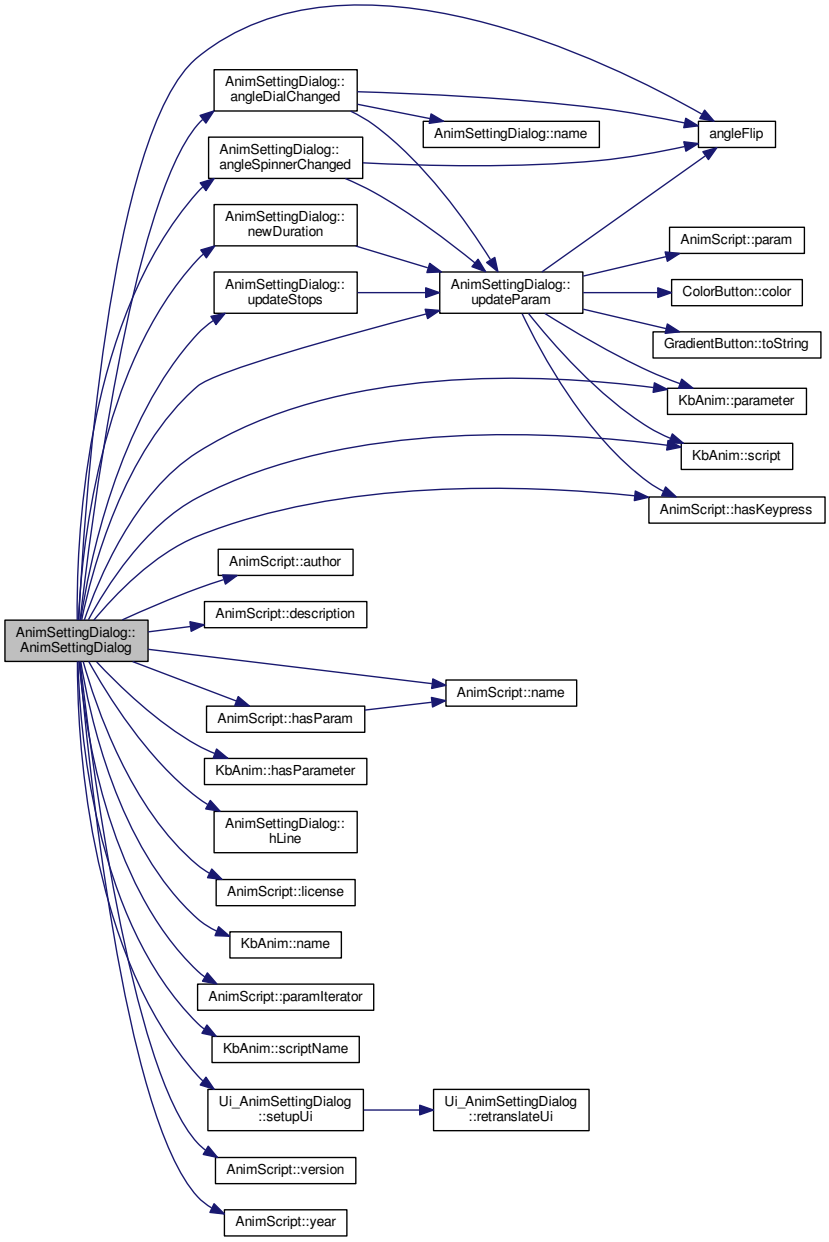
```
49                  || param.name == "kpmodestop" || param.name == "kprelease")
50              continue;
51          QVariant value = anim->parameter(param.name);
52          // Display prefix label on the left (except for booleans and labels)
53          if(param.type != AnimScript::Param::BOOL && param.
        type != AnimScript::Param::LABEL)
54              ui->settingsGrid->addWidget(new QLabel(param.prefix, this), row, 1);
55          // Configure and display main widget
56          QWidget* widget = 0;
57          int colSpan = 1;
58          QString postfix = param.postfix;
59          switch(param.type){
60          case AnimScript::Param::BOOL:
61              widget = new QCheckBox(param.prefix, this);
62              ((QCheckBox*)widget)->setChecked(value.toBool());
63              colSpan = 4;
64              connect(widget, SIGNAL(stateChanged(int)), &updateMapper, SLOT(map()));
65              break;
66          case AnimScript::Param::LONG:
67              widget = new QSpinBox(this);
68              ((QSpinBox*)widget)->setMinimum(param.minimum.toInt());
69              ((QSpinBox*)widget)->setMaximum(param.maximum.toInt());
70              ((QSpinBox*)widget)->setValue(value.toInt());
71              if(postfix.length() <= 3){
72                  ((QSpinBox*)widget)->setSuffix(postfix);
73                  postfix = "";
74              }
75              connect(widget, SIGNAL(valueChanged(int)), &updateMapper, SLOT(map()));
76              break;
77          case AnimScript::Param::DOUBLE:
78              widget = new QDoubleSpinBox(this);
79              ((QDoubleSpinBox*)widget)->setDecimals(1);
80              ((QDoubleSpinBox*)widget)->setMinimum(param.minimum.toDouble());
81              ((QDoubleSpinBox*)widget)->setMaximum(param.maximum.toDouble());
82              ((QDoubleSpinBox*)widget)->setValue(value.toDouble());
83              if(postfix.length() <= 3){
84                  ((QDoubleSpinBox*)widget)->setSuffix(postfix);
85                  postfix = "";
86              }
87              connect(widget, SIGNAL(valueChanged(double)), &updateMapper, SLOT(map()));
88              break;
89          case AnimScript::Param::RGB:
90              widget = new ColorButton(this);
91              ((ColorButton*)widget)->color(QColor("#" + value.toString()));
92              colSpan = 3;
93              connect(widget, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
94              break;
95          case AnimScript::Param::ARGB:{
96              widget = new ColorButton(this, true);
97              QString val = value.toString();
98              QColor color;
99              if(val.length() == 8){
100                 color = "#" + val.right(6);
101                 color.setAlpha(val.left(2).toInt(0, 16));
102             } else
103                 color = "#" + val;
104             ((ColorButton*)widget)->color(color);
105             colSpan = 3;
106             connect(widget, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
107             break;
108         }
109         case AnimScript::Param::GRADIENT:
110             widget = new GradientButton(this);
111             ((GradientButton*)widget)->fromString(value.toString());
112             colSpan = 3;
113             connect(widget, SIGNAL(gradientChanged()), &updateMapper, SLOT(map()));
114             break;
115         case AnimScript::Param::AGRADIENT:
116             widget = new GradientButton(this, true);
117             ((GradientButton*)widget)->fromString(value.toString());
118             colSpan = 3;
119             connect(widget, SIGNAL(gradientChanged()), &updateMapper, SLOT(map()));
120             break;
121         case AnimScript::Param::ANGLE:
122             widget = new QDial(this);
123             ((QDial*)widget)->setFixedSize(60, 60);
124             ((QDial*)widget)->setMinimum(0);
125             ((QDial*)widget)->setMaximum(360);
126             // this is NOT a typo...     ^
127             // Even though 360 shouldn't be valid, the wheel/arrow keys will lose 1 degree at the bottom
       unless 360 is set as max
128             ((QDial*)widget)->setSingleStep(5);
129             ((QDial*)widget)->setPageStep(15);
130             ((QDial*)widget)->setNotchTarget(6);
131             ((QDial*)widget)->setNotchesVisible(true);
132             ((QDial*)widget)->setWrapping(true);
133             ((QDial*)widget)->setInvertedAppearance(true);
```

```
134             ((QDial*)widget)->setValue(angleFlip(value.toInt()));
135             angleDialMapper.setMapping(widget, param.name);
136             connect(widget, SIGNAL(valueChanged(int)), &angleDialMapper, SLOT(map()));
137             break;
138         case AnimScript::Param::STRING:
139             widget = new QLineEdit(this);
140             ((QLineEdit*)widget)->setText(value.toString());
141             colSpan = 3;
142             connect(widget, SIGNAL(textEdited(const QString&)), &updateMapper, SLOT(map()));
143             break;
144         case AnimScript::Param::LABEL:
145             widget = new QLabel(this);
146             ((QLabel*)widget)->setText(param.prefix);
147             colSpan = 4;
148             break;
149         default:
150             break;
151         }
152         if(widget)
153             updateMapper.setMapping(widget, param.name);
154         if(param.type == AnimScript::Param::BOOL || param.
    type == AnimScript::Param::LABEL){
155             // Boolean values are placed on the left with no prefix or postfix
156             settingWidgets[param.name] = widget;
157             ui->settingsGrid->addWidget(widget, row, 3, 1, colSpan);
158         } else {
159             // Display the widget
160             if(widget){
161                 settingWidgets[param.name] = widget;
162                 ui->settingsGrid->addWidget(widget, row, 3, 1, colSpan);
163             }
164             // Angles additionally have a spin box
165             if(param.type == AnimScript::Param::ANGLE){
166                 QSpinBox* spinner = new QSpinBox(this);
167                 spinner->setMinimum(0);
168                 spinner->setMaximum(359);
169                 spinner->setWrapping(true);
170                 spinner->setSuffix("°");
171                 spinner->setValue(value.toInt());
172                 angleSpinners[param.name] = spinner;
173                 angleSpinnerMapper.setMapping(spinner, param.
    name);
174                 connect(spinner, SIGNAL(valueChanged(int)), &angleSpinnerMapper, SLOT(map
    ()));
175                 ui->settingsGrid->addWidget(spinner, row, 4);
176                 colSpan = 2;
177             }
178             // Display postfix label on the right
179             ui->settingsGrid->addWidget(new QLabel(postfix, this), row, 3 + colSpan, 1, 4 -
    colSpan);
180             if(colSpan < 3 && !rSpacePlaced){
181                 // Add a spacer to compress short elements to the left
182                 ui->settingsGrid->addItem(new QSpacerItem(0, 0, QSizePolicy::Minimum), row, 4
     + colSpan);
183                 rSpacePlaced = true;
184             }
185         }
186         row++;
187     }
188     // Add playback info at bottom
189     ui->settingsGrid->addItem(new QSpacerItem(0, 10, QSizePolicy::Fixed, QSizePolicy::Fixed),
     row++, 6);
190     ui->settingsGrid->addWidget(new QLabel("<b>Playback</b>", this), row++, 0, 1, 7);
191     ui->settingsGrid->addWidget(hLine(), row++, 0, 1, 7);
192     if(script->hasParam("duration")){
193         // Show duration spinner (if allowed)
194         lastDuration = anim->parameter("duration").toDouble();
195         ui->settingsGrid->addWidget(new QLabel("Duration:", this), row, 1);
196         QDoubleSpinBox* spinner = new QDoubleSpinBox(this);
197         spinner->setDecimals(1);
198         spinner->setMinimum(0.1);
199         spinner->setValue(lastDuration);
200         connect(spinner, SIGNAL(valueChanged(double)), this, SLOT(newDuration(double)));
201         settingWidgets["duration"] = spinner;
202         ui->settingsGrid->addWidget(spinner, row, 3, 1, 1);
203         ui->settingsGrid->addWidget(new QLabel("seconds", this), row, 4, 1, 2);
204         row++;
205     }
206     // Show boxes for start with mode/with keypress
207     QCheckBox* check = new QCheckBox("Start with mode", this);
208     check->setChecked(anim->parameter("trigger").toBool());
209     ui->settingsGrid->addWidget(check, row, 3, 1, 4);
210     settingWidgets["trigger"] = check;
211     connect(check, SIGNAL(stateChanged(int)), &updateMapper, SLOT(map()));
212     updateMapper.setMapping(check, "trigger");
213     row++;
214     check = new QCheckBox("Start with key press", this);
```

```
215     check->setChecked(anim->parameter("kptrigger").toBool());
216     ui->settingsGrid->addWidget(check, row, 3, 1, 2);
217     settingWidgets["kptrigger"] = check;
218     connect(check, SIGNAL(stateChanged(int)), &updateMapper, SLOT(map()));
219     updateMapper.setMapping(check, "kptrigger");
220     // Add an option allowing the user to select keypress mode
221     QComboBox* combo = new QComboBox(this);
222     int selected = anim->parameter("kpmode").toInt();
223     if(script->hasKeypress()){
224         // If the script supports keypresses, show the option to handle them that way (default)
225         combo->addItem("on pressed key");
226         combo->addItem("on whole keyboard");
227         combo->addItem("on keyboard (once)");
228     } else {
229         selected--;
230         // Otherwise, just show the choice of whether to start it every time or just once
231         combo->addItem("every time");
232         combo->addItem("only once");
233     }
234     if(selected < 0 || selected > combo->count())
235         selected = 0;
236     combo->setCurrentIndex(selected);
237     ui->settingsGrid->addWidget(combo, row, 5, 1, 2);
238     settingWidgets["kpmode"] = combo;
239     connect(combo, SIGNAL(activated(int)), &updateMapper, SLOT(map()));
240     updateMapper.setMapping(combo, "kpmode");
241     row++;
242
243     // Add horizontal spacer to compress content to left
244     ui->settingsGrid->addItem(new QSpacerItem(0, 0, QSizePolicy::Expanding), 0, 6);
245     // Add vertical spacer to compress content to top
246     ui->settingsGrid->addItem(new QSpacerItem(0, 0, QSizePolicy::Minimum,
    QSizePolicy::Expanding), row, 0);
247
248     // Add timing fields
249     settingWidgets["delay"] = ui->delayBox;
250     ui->delayBox->setValue(anim->parameter("delay").toDouble());
251     settingWidgets["kpdelay"] = ui->kpDelayBox;
252     ui->kpDelayBox->setValue(anim->parameter("kpdelay").toDouble());
253     settingWidgets["kpmodestop"] = ui->kpModeStopBox;
254     ui->kpModeStopBox->setChecked(anim->parameter("kpmodestop").toBool());
255     connect(ui->kpModeStopBox, SIGNAL(clicked(bool)), &
    updateMapper, SLOT(map()));
256     updateMapper.setMapping(ui->kpModeStopBox, "kpmodestop");
257     settingWidgets["kprelease"] = ui->kpReleaseBox;
258     ui->kpReleaseBox->setChecked(anim->parameter("kprelease").toBool());
259     if(anim->hasParameter("repeat")){
260         hasRepeat = true;
261         settingWidgets["repeat"] = ui->repeatBox;
262         ui->repeatBox->setValue(anim->parameter("repeat").toDouble());
263         settingWidgets["kprepeat"] = ui->kpRepeatBox;
264         ui->kpRepeatBox->setValue(anim->parameter("kprepeat").toDouble());
265         // If repeat is enabled, add repeat counts as integer values
266         // Mode repeat
267         ui->timeGrid->addWidget(new QLabel("Repeat:", this), 4, 1);
268         QSpinBox* spinner = new QSpinBox(this);
269         spinner->setMinimum(0);
270         spinner->setMaximum(1000000);
271         spinner->setValue(anim->parameter("stop").toInt());
272         settingWidgets["stop"] = spinner;
273         connect(spinner, SIGNAL(valueChanged(int)), &updateMapper, SLOT(map()));
274         updateMapper.setMapping(spinner, "stop");
275         ui->timeGrid->addWidget(spinner, 4, 3);
276         ui->timeGrid->addWidget(new QLabel("times", this), 4, 4);
277         // KP repeat
278         ui->timeGrid->addWidget(new QLabel("Repeat:", this), 12, 1);
279         spinner = new QSpinBox(this);
280         spinner->setMinimum(0);
281         spinner->setMaximum(1000000);
282         spinner->setValue(anim->parameter("kpstop").toInt());
283         settingWidgets["kpstop"] = spinner;
284         connect(spinner, SIGNAL(valueChanged(int)), &updateMapper, SLOT(map()));
285         updateMapper.setMapping(spinner, "kpstop");
286         ui->timeGrid->addWidget(spinner, 12, 3);
287         ui->timeGrid->addWidget(new QLabel("times", this), 12, 4);
288         // Infinite repeat toggles
289         stopCheck = new QCheckBox("Forever", this);
290         stopCheck->setChecked(anim->parameter("stop").toInt() < 0);
291         ui->timeGrid->addWidget(stopCheck, 4, 5);
292         kpStopCheck = new QCheckBox("Forever", this);
293         connect(stopCheck, SIGNAL(clicked()), this, SLOT(updateStops()));
294         kpStopCheck->setChecked(anim->parameter("kpstop").toInt() < 0);
295         ui->timeGrid->addWidget(kpStopCheck, 12, 5);
296         connect(kpStopCheck, SIGNAL(clicked()), this, SLOT(
    updateStops()));
297     } else {
298         hasRepeat = false;
```

```
299         // If repeat is not enabled, hide repeat-related fields
300         ui->repeatBox->setHidden(true);
301         ui->repeatLabel->setHidden(true);
302         ui->repeatLabel_2->setHidden(true);
303         ui->kpRepeatBox->setHidden(true);
304         ui->kpRepeatLabel->setHidden(true);
305         ui->kpRepeatLabel_2->setHidden(true);
306         // Add stop times as double values
307         // Stop time
308         QDoubleSpinBox* spinner = new QDoubleSpinBox(this);
309         spinner->setDecimals(1);
310         spinner->setMinimum(0.1);
311         spinner->setMaximum(24. * 60. * 60.);
312         double stop = anim->parameter("stop").toDouble();
313         if(stop <= 0.)
314             spinner->setValue(lastDuration);
315         else
316             spinner->setValue(stop);
317         settingWidgets["stop"] = spinner;
318         connect(spinner, SIGNAL(valueChanged(double)), &updateMapper, SLOT(map()));
319         updateMapper.setMapping(spinner, "stop");
320         ui->timeGrid->addWidget(spinner, 4, 3);
321         ui->timeGrid->addWidget(new QLabel("seconds", this), 4, 4);
322         // KP stop time
323         spinner = new QDoubleSpinBox(this);
324         spinner->setDecimals(1);
325         spinner->setMinimum(0.1);
326         spinner->setMaximum(24. * 60. * 60.);
327         double kpstop = anim->parameter("kpstop").toDouble();
328         if(kpstop <= 0.)
329             spinner->setValue(lastDuration);
330         else
331             spinner->setValue(kpstop);
332         settingWidgets["kpstop"] = spinner;
333         connect(spinner, SIGNAL(valueChanged(double)), &updateMapper, SLOT(map()));
334         updateMapper.setMapping(spinner, "kpstop");
335         ui->timeGrid->addWidget(spinner, 12, 3);
336         ui->timeGrid->addWidget(new QLabel("seconds", this), 12, 4);
337         // Infinite run toggles
338         stopCheck = new QCheckBox("Stop after:", this);
339         stopCheck->setChecked(stop > 0.);
340         ui->timeGrid->addWidget(stopCheck, 4, 1);
341         connect(stopCheck, SIGNAL(clicked()), this, SLOT(updateStops()));
342         kpStopCheck = new QCheckBox("Stop after:", this);
343         kpStopCheck->setChecked(kpstop > 0.);
344         ui->timeGrid->addWidget(kpStopCheck, 12, 1);
345         connect(kpStopCheck, SIGNAL(clicked()), this, SLOT(
    updateStops()));
346     }
347     updateStops();
348
349     // Set script info UI
350     ui->sNameLabel->setText(script->name());
351     ui->sVerLabel->setText(script->version());
352     ui->sAuthorLabel->setText(script->author());
353     ui->sYearLabel->setText(script->year());
354     ui->sLicenseLabel->setText(script->license());
355     ui->sDescLabel->setText(script->description());
356
357     // Lock dialog size
358     setFixedSize(minimumSize());
359 }
```

Here is the call graph for this function:



12.5.2.2 AnimSettingDialog::~AnimSettingDialog ( )

Definition at line 476 of file animsettingdialog.cpp.

References ui.

```
476                                        {
477     delete ui;
478 }
```

12.5.3 Member Function Documentation

12.5.3.1 void AnimSettingDialog::angleDialChanged (QString *name*) [private], [slot]

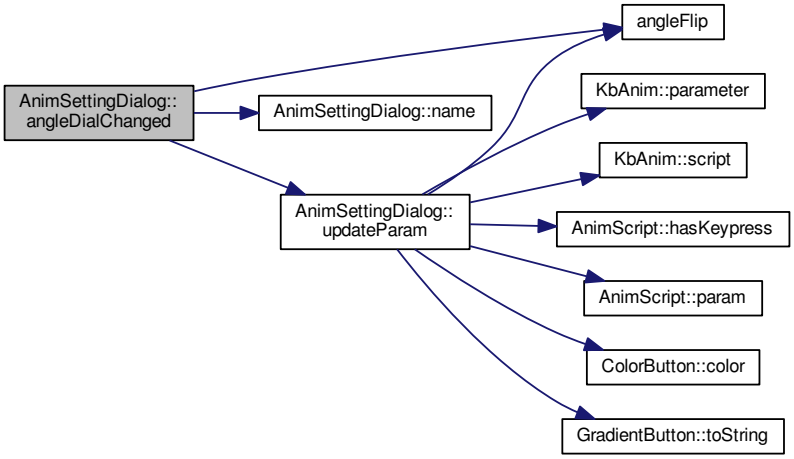
Definition at line 389 of file animsettingdialog.cpp.

References angleFlip(), angleSpinners, name(), settingWidgets, and updateParam().

Referenced by AnimSettingDialog().

```
389                                                 {
390     // Dial changed; update spinner value
391     angleSpinners[name]->setValue(angleFlip(((QDial*)
      settingWidgets[name])->value()));
392     updateParam(name);
393 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.5.3.2 void AnimSettingDialog::angleSpinnerChanged (QString *name*) [private], [slot]

Definition at line 395 of file animsettingdialog.cpp.

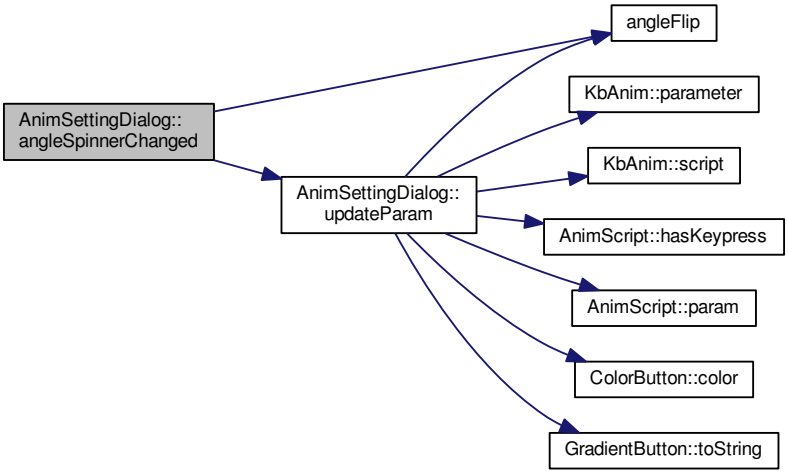
References angleFlip(), angleSpinners, settingWidgets, and updateParam().

Referenced by AnimSettingDialog().

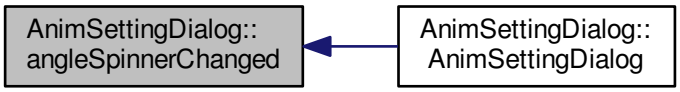
```
395                                                 {
```

```
396     // Spinner changed; update dial value
397     ((QDial*)settingWidgets[name])->setValue(angleFlip(
      angleSpinners[name]->value()));
398     updateParam(name);
399 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.5.3.3 QFrame * AnimSettingDialog::hLine ( ) [private]

Definition at line 480 of file animsettingdialog.cpp.

References frame.

Referenced by AnimSettingDialog().

```
480                                 {
481     QFrame* frame = new QFrame(this);
482     frame->setFrameShape(QFrame::HLine);
483     frame->setFrameShadow(QFrame::Sunken);
484     return frame;
485 }
```

Here is the caller graph for this function:

12.5.3.4 QString AnimSettingDialog::name ( ) const

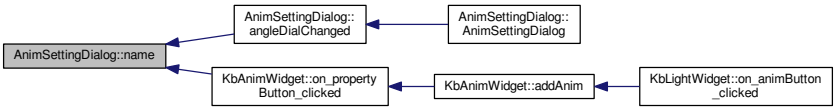
Definition at line 472 of file animsettingdialog.cpp.

References Ui_AnimSettingDialog::animName, and ui.

Referenced by angleDialChanged(), and KbAnimWidget::on_propertyButton_clicked().

```
472                                         {
473     return ui->animName->text();
474 }
```

Here is the caller graph for this function:



12.5.3.5 void AnimSettingDialog::newDuration (double *duration*) [private], [slot]

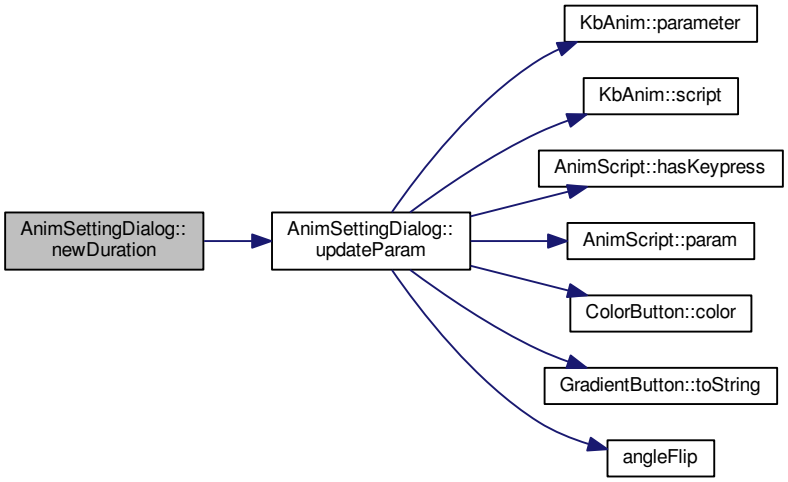
Definition at line 361 of file animsettingdialog.cpp.

References lastDuration, settingWidgets, and updateParam().

Referenced by AnimSettingDialog().

```
361                                                 {
362     // Duration changed. Automatically fix Repeat and KP Repeat to the same values, if they matched before
363     QDoubleSpinBox* rep = (QDoubleSpinBox*)settingWidgets.value("repeat");
364     QDoubleSpinBox* kpRep = (QDoubleSpinBox*)settingWidgets.value("kprepeat");
365     if(rep && rep->value() == lastDuration)
366         rep->setValue(duration);
367     if(kpRep && kpRep->value() == lastDuration)
368         kpRep->setValue(duration);
369     lastDuration = duration;
370     updateParam("duration");
371     updateParam("repeat");
372     updateParam("kprepeat");
373 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

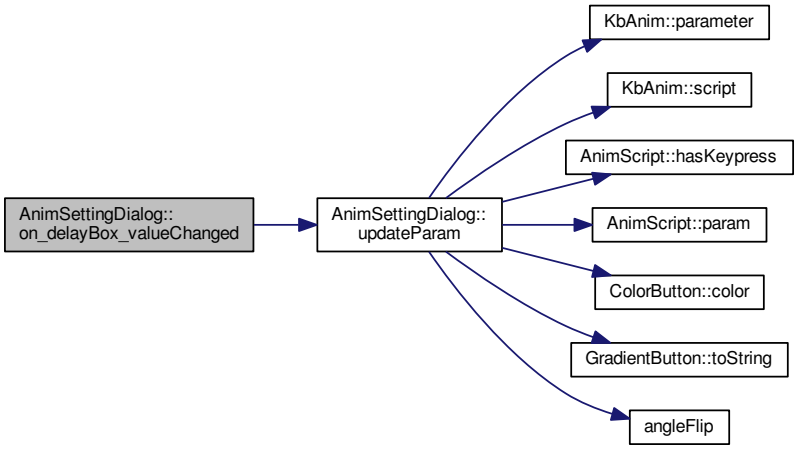
12.5.3.6 void AnimSettingDialog::on_delayBox_valueChanged (double *arg1*) [private], [slot]

Definition at line 487 of file animsettingdialog.cpp.

References updateParam().

```
487                                                    {
488     updateParam("delay");
489 }
```

Here is the call graph for this function:



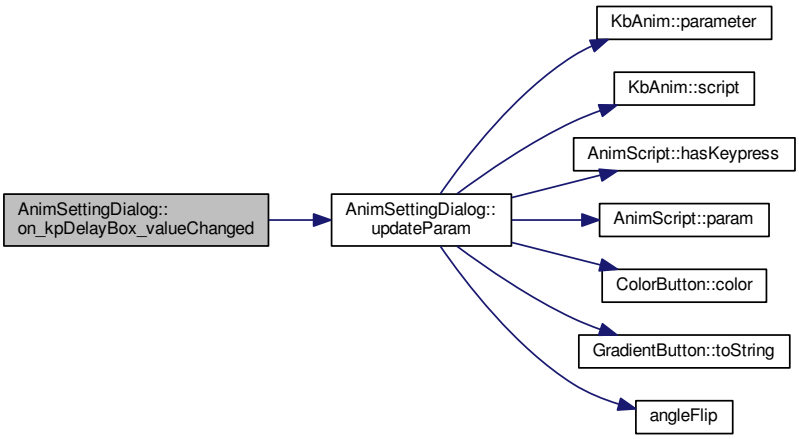
12.5.3.7 void AnimSettingDialog::on_kpDelayBox_valueChanged (double *arg1*) [private], [slot]

Definition at line 495 of file animsettingdialog.cpp.

References updateParam().

```
495                                                    {
496     updateParam("kpdelay");
497 }
```

Here is the call graph for this function:



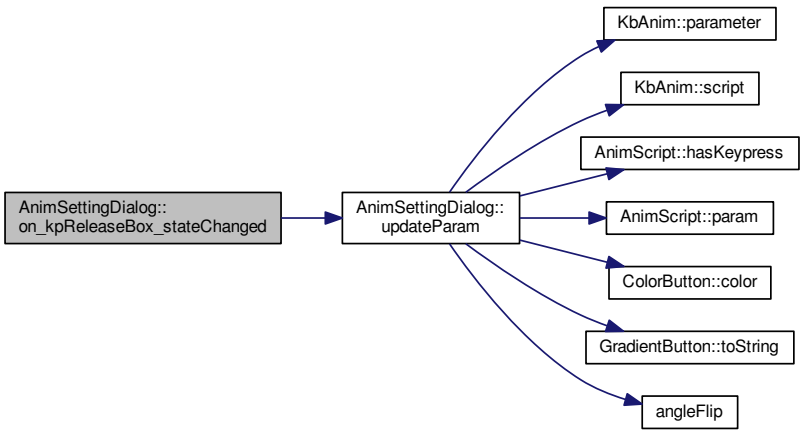
12.5.3.8 void AnimSettingDialog::on_kpReleaseBox_stateChanged (int *arg1*) [private], [slot]

Definition at line 503 of file animsettingdialog.cpp.

References updateParam().

```
503                                                     {
504     updateParam("kprelease");
505 }
```

Here is the call graph for this function:



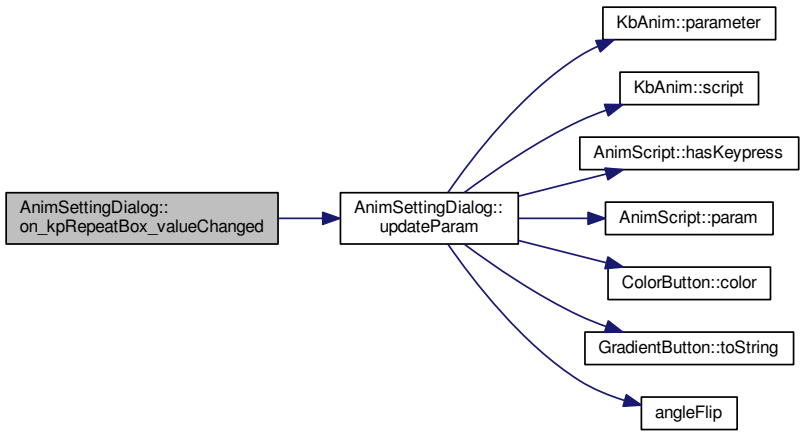
12.5.3.9 void AnimSettingDialog::on_kpRepeatBox_valueChanged (double *arg1*) [private], [slot]

Definition at line 499 of file animsettingdialog.cpp.

References updateParam().

```
499                                                     {
500     updateParam("kprepeat");
501 }
```

Here is the call graph for this function:



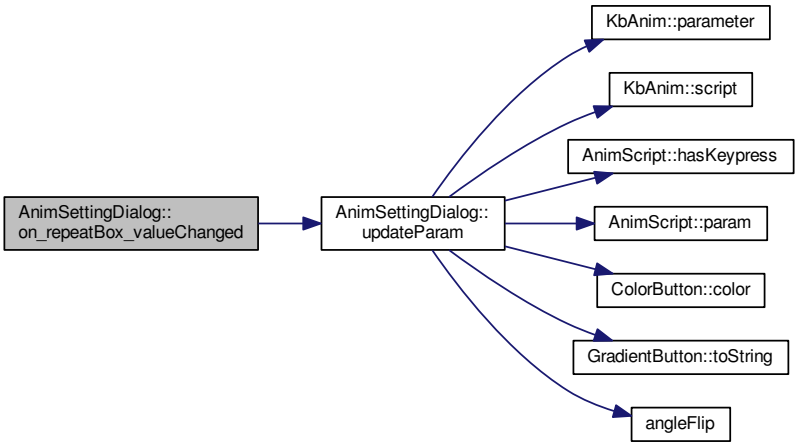
12.5.3.10 void AnimSettingDialog::on_repeatBox_valueChanged (double *arg1*) [private], [slot]

Definition at line 491 of file animsettingdialog.cpp.

References updateParam().

```
491                                                          {
492     updateParam("repeat");
493 }
```

Here is the call graph for this function:



12.5.3.11 void AnimSettingDialog::updateParam (QString *name*) [private], [slot]

Definition at line 401 of file animsettingdialog.cpp.

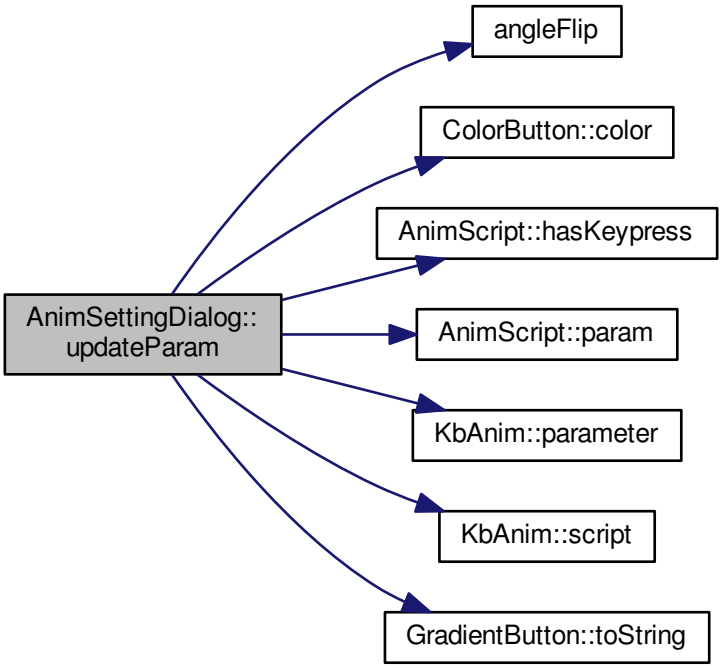
References _anim, AnimScript::Param::AGRADIENT, AnimScript::Param::ANGLE, angleFlip(), AnimScript::Param::ARGB, AnimScript::Param::BOOL, ColorButton::color(), AnimScript::Param::DOUBLE, AnimScript::Param::GRADIENT, AnimScript::hasKeypress(), hasRepeat, kpStopCheck, AnimScript::Param::LONG, AnimScript::param(), KbAnim::parameter(), AnimScript::Param::RGB, KbAnim::script(), settingWidgets, stopCheck, AnimScript::Param::STRING, GradientButton::toString(), and AnimScript::Param::type.

Referenced by angleDialChanged(), angleSpinnerChanged(), AnimSettingDialog(), newDuration(), on_delayBox_valueChanged(), on_kpDelayBox_valueChanged(), on_kpReleaseBox_stateChanged(), on_kpRepeatBox_valueChanged(), on_repeatBox_valueChanged(), and updateStops().

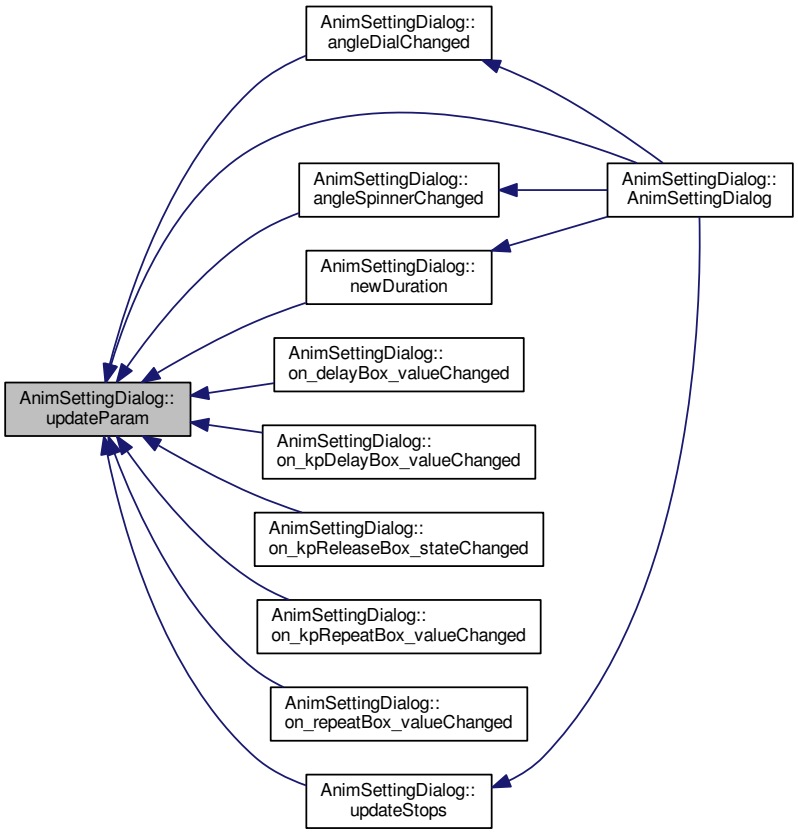
```
401                                                  {
402     if(!settingWidgets.contains(name))
403         return;
404     // stop and kpstop have defeat switches
405     if((name == "stop" && stopCheck->isChecked() == hasRepeat)
406             || (name == "kpstop" && kpStopCheck->isChecked() ==
    hasRepeat)){
407         _anim->parameter(name, -1);
408         return;
409     } else if(name == "kpmode"){
410         // kpmode uses a drop-down, and selection is inverted
411         QComboBox* widget = (QComboBox*)settingWidgets[name];
412         int selected = widget->currentIndex();
413         if(!_anim->script()->hasKeypress())
414             selected++;
415         _anim->parameter(name, selected);
416         return;
417     }
418     // Read value based on type
```

```
419     switch(_anim->script()->param(name).type){
420     case AnimScript::Param::BOOL:{
421         QCheckBox* widget = (QCheckBox*)settingWidgets[name];
422         _anim->parameter(name, (int)widget->isChecked());
423         break;
424     }
425     case AnimScript::Param::LONG:{
426         QSpinBox* widget = (QSpinBox*)settingWidgets[name];
427         _anim->parameter(name, widget->value());
428         break;
429     }
430     case AnimScript::Param::DOUBLE:{
431         QDoubleSpinBox* widget = (QDoubleSpinBox*)settingWidgets[
    name];
432         _anim->parameter(name, widget->value());
433         break;
434     }
435     case AnimScript::Param::RGB:{
436         ColorButton* widget = (ColorButton*)settingWidgets[
    name];
437         QColor color = widget->color();
438         char hex[7];
439         snprintf(hex, sizeof(hex), "%02x%02x%02x", color.red(), color.green(), color.blue());
440         _anim->parameter(name, QString(hex));
441         break;
442     }
443     case AnimScript::Param::ARGB:{
444         ColorButton* widget = (ColorButton*)settingWidgets[
    name];
445         QColor color = widget->color();
446         char hex[9];
447         snprintf(hex, sizeof(hex), "%02x%02x%02x%02x", color.alpha(), color.red(), color.green(), color.
    blue());
448         _anim->parameter(name, QString(hex));
449         break;
450     }
451     case AnimScript::Param::GRADIENT:
452     case AnimScript::Param::AGRADIENT:{
453         GradientButton* widget = (GradientButton*)
    settingWidgets[name];
454         _anim->parameter(name, widget->toString());
455         break;
456     }
457     case AnimScript::Param::ANGLE:{
458         QDial* widget = (QDial*)settingWidgets[name];
459         _anim->parameter(name, angleFlip(widget->value()));
460         break;
461     }
462     case AnimScript::Param::STRING:{
463         QLineEdit* widget = (QLineEdit*)settingWidgets[name];
464         _anim->parameter(name, widget->text());
465         break;
466     }
467     default:
468         break;
469     }
470 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.5.3.12 void AnimSettingDialog::updateStops ( ) [private], [slot]

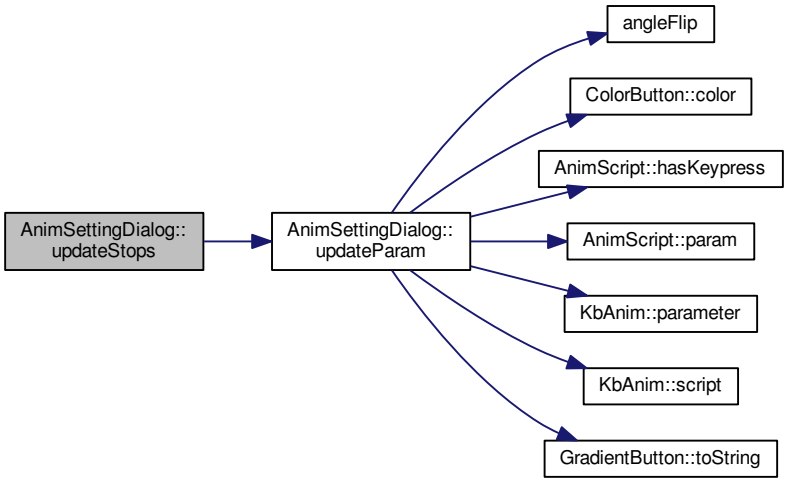
Definition at line 375 of file animsettingdialog.cpp.

References hasRepeat, kpStopCheck, settingWidgets, stopCheck, and updateParam().

Referenced by AnimSettingDialog().

```
375                                  {
376     if(hasRepeat){
377         // When repeat is enabled, the check box is "repeat forever", so disable stop count if checked
378         settingWidgets.value("stop")->setDisabled(stopCheck->isChecked());
379         settingWidgets.value("kpstop")->setDisabled(kpStopCheck->isChecked());
380     } else {
381         // When repeat is disabled, the check box is "stop after", so disable stop count UNLESS checked
382         settingWidgets.value("stop")->setEnabled(stopCheck->isChecked());
383         settingWidgets.value("kpstop")->setEnabled(kpStopCheck->isChecked());
384     }
385     updateParam("stop");
386     updateParam("kpstop");
387 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.5.4 Field Documentation

12.5.4.1 KbAnim* AnimSettingDialog::_anim [private]

Definition at line 32 of file animsettingdialog.h.

Referenced by updateParam().

12.5.4.2 QSignalMapper AnimSettingDialog::angleDialMapper [private]

Definition at line 38 of file animsettingdialog.h.

Referenced by AnimSettingDialog().

12.5.4.3 QSignalMapper AnimSettingDialog::angleSpinnerMapper [private]

Definition at line 38 of file animsettingdialog.h.

Referenced by AnimSettingDialog().

12.5.4.4 QMap<QString, QSpinBox*> AnimSettingDialog::angleSpinners [private]

Definition at line 37 of file animsettingdialog.h.

Referenced by angleDialChanged(), angleSpinnerChanged(), and AnimSettingDialog().

12.5.4.5 bool AnimSettingDialog::hasRepeat [private]

Definition at line 30 of file animsettingdialog.h.

Referenced by AnimSettingDialog(), updateParam(), and updateStops().

12.5.4.6 QCheckBox * AnimSettingDialog::kpStopCheck [private]

Definition at line 29 of file animsettingdialog.h.

Referenced by AnimSettingDialog(), updateParam(), and updateStops().

12.5.4.7 double AnimSettingDialog::lastDuration [private]

Definition at line 33 of file animsettingdialog.h.

Referenced by AnimSettingDialog(), and newDuration().

12.5.4.8 QMap<QString, QWidget*> AnimSettingDialog::settingWidgets [private]

Definition at line 34 of file animsettingdialog.h.

Referenced by angleDialChanged(), angleSpinnerChanged(), AnimSettingDialog(), newDuration(), updateParam(), and updateStops().

12.5.4.9 QCheckBox* AnimSettingDialog::stopCheck [private]

Definition at line 29 of file animsettingdialog.h.

Referenced by AnimSettingDialog(), updateParam(), and updateStops().

12.5.4.10 Ui::AnimSettingDialog* AnimSettingDialog::ui [private]

Definition at line 28 of file animsettingdialog.h.

Referenced by AnimSettingDialog(), name(), and ~AnimSettingDialog().

12.5.4.11 QSignalMapper AnimSettingDialog::updateMapper [private]

Definition at line 35 of file animsettingdialog.h.

Referenced by AnimSettingDialog().

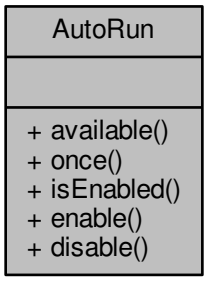
The documentation for this class was generated from the following files:

- src/ckb/animsettingdialog.h
- src/ckb/animsettingdialog.cpp

12.6 AutoRun Class Reference

`#include <src/ckb/autorun.h>`

Collaboration diagram for AutoRun:



Static Public Member Functions

- static bool available ()
- static bool once ()
- static bool isEnabled ()
- static void enable ()
- static void disable ()

12.6.1 Detailed Description

Definition at line 8 of file autorun.h.

12.6.2 Member Function Documentation

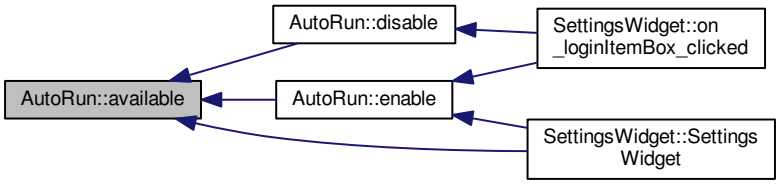
12.6.2.1 bool AutoRun::available ( ) [static]

Definition at line 20 of file autorun.cpp.

Referenced by disable(), enable(), and SettingsWidget::SettingsWidget().

```
20                       {
21     // Allow autostart if the program is located in a system path
22 #ifdef Q_OS_LINUX
23     return QDir::root().absoluteFilePath(QStandardPaths::findExecutable("ckb")) == qApp->
       applicationFilePath();
24 #elif defined(Q_OS_MACX)
25     return qApp->applicationFilePath().startsWith("/Applications/ckb.app", Qt::CaseInsensitive);
26 #endif
27 }
```

Here is the caller graph for this function:



12.6.2.2 void AutoRun::disable ( ) [static]

Definition at line 53 of file autorun.cpp.

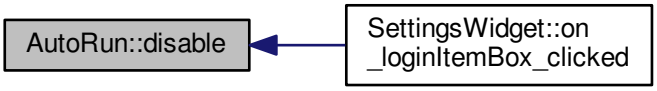
References available().

Referenced by SettingsWidget::on_loginItemBox_clicked().

```
53                      {
54     if(!available())
55         return;
56     // Remove file
57     QFile::remove(path.absoluteFilePath(file));
58 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.6.2.3 void AutoRun::enable ( ) [static]

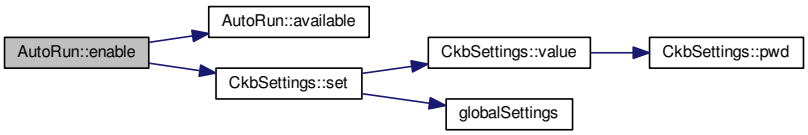
Definition at line 42 of file autorun.cpp.

References available(), and CkbSettings::set().

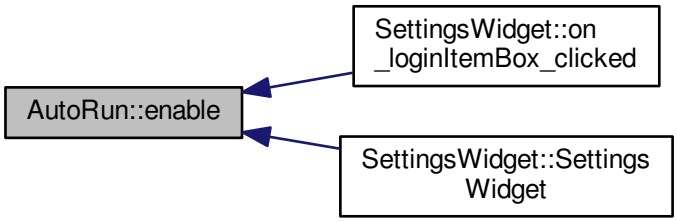
Referenced by SettingsWidget::on_loginItemBox_clicked(), and SettingsWidget::SettingsWidget().

```
42                      {
43     if(!available())
44         return;
45     // Copy file into place
46     if(!path.exists())
47         QDir::home().mkpath(path.absolutePath());
48     QFile::copy(internalFile, path.absoluteFilePath(file));
49     // Mark once() as done
50     CkbSettings::set(settingPath, true);
51 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.6.2.4 bool AutoRun::isEnabled ( ) [static]

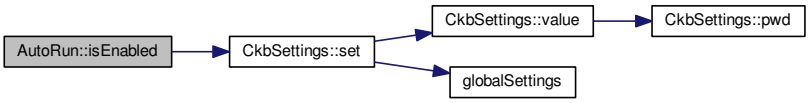
Definition at line 33 of file autorun.cpp.

References CkbSettings::set().

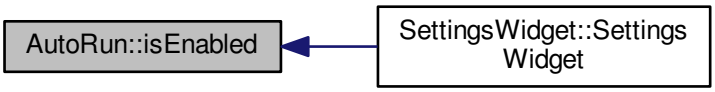
Referenced by SettingsWidget::SettingsWidget().

```
33                      {
34     // Check if the file exists. If not, autostart is disabled.
35     if(!path.exists() || !path.exists(file))
36         return false;
37     // If autostart is enabled, set the flag from once() (in case it hasn't been done yet)
38     CkbSettings::set(settingPath, true);
39     return true;
40 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.6.2.5 bool AutoRun::once ( ) [static]

Definition at line 29 of file autorun.cpp.

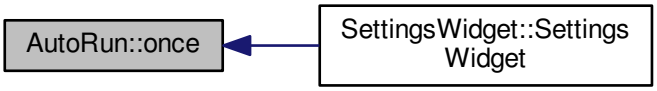
References CkbSettings::get().

Referenced by SettingsWidget::SettingsWidget().

```
29                  {
30      return CkbSettings::get(settingPath).toBool();
31  }
```

Here is the call graph for this function:

Here is the caller graph for this function:



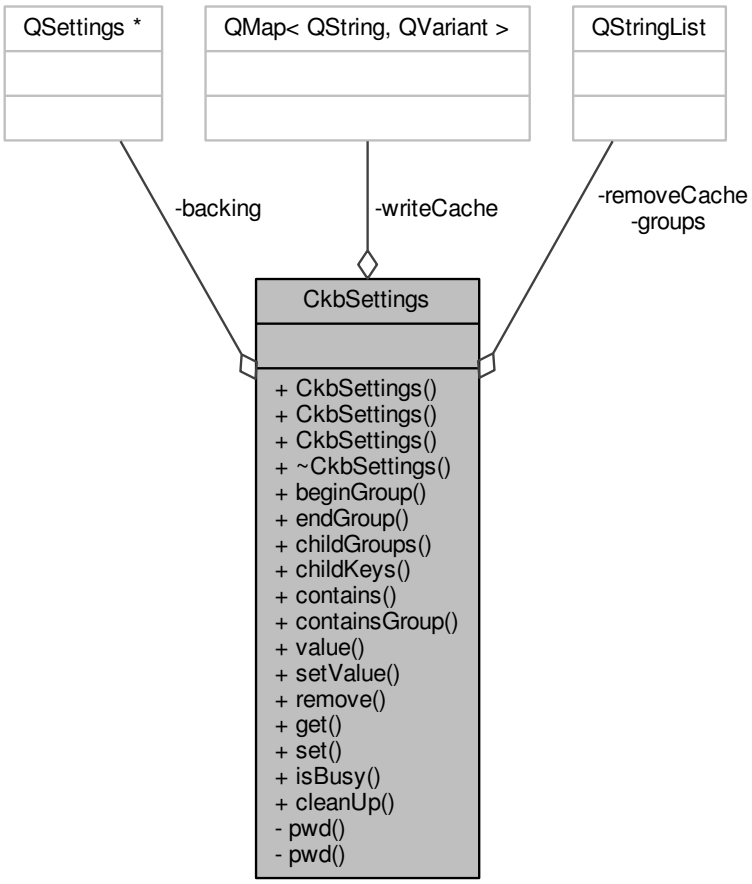
The documentation for this class was generated from the following files:

- src/ckb/autorun.h
- src/ckb/autorun.cpp

12.7 CkbSettings Class Reference

`#include <src/ckb/ckbsettings.h>`

Collaboration diagram for CkbSettings:



Public Member Functions

- CkbSettings ()
- CkbSettings (const QString &basePath, bool eraseExisting=false)
- CkbSettings (QSettings &settings)
- ~CkbSettings ()
- void beginGroup (const QString &prefix)
- void endGroup ()
- QStringList childGroups () const
- QStringList childKeys () const
- bool contains (const QString &key) const
- bool containsGroup (const QString &group)
- QVariant value (const QString &key, const QVariant &defaultValue=QVariant()) const
- void setValue (const QString &key, const QVariant &value)
- void remove (const QString &key)

Static Public Member Functions

- static QVariant get (const QString &key, const QVariant &defaultValue=QVariant())

- static void set (const QString &key, const QVariant &value)
- static bool isBusy ()
- static void cleanUp ()

Private Member Functions

- QString pwd () const
- QString pwd (const QString &key) const

Private Attributes

- QSettings * backing
- QStringList groups
- QStringList removeCache
- QMap< QString, QVariant > writeCache

12.7.1 Detailed Description

Definition at line 11 of file ckbsettings.h.

12.7.2 Constructor & Destructor Documentation

12.7.2.1 CkbSettings::CkbSettings ( )

Definition at line 61 of file ckbsettings.cpp.

```
61                          :
62     backing(globalSettings()) {
63 }
```

12.7.2.2 CkbSettings::CkbSettings (const QString & *basePath*, bool *eraseExisting* = `false`)

Definition at line 65 of file ckbsettings.cpp.

References beginGroup().

```
65                                                                  :
66     backing(globalSettings()) {
67     if(basePath.isEmpty()){
68         if(eraseExisting)
69             qDebug() << "CkbSettings created with basePath = \"\" and eraseExisting = true. This is a
       mistake.";
70         return;
71     }
72     if(eraseExisting)
73         remove(basePath);
74     beginGroup(basePath);
75 }
```

Here is the call graph for this function:

12.7.2.3 CkbSettings::CkbSettings (QSettings & *settings*)

Definition at line 77 of file ckbsettings.cpp.

```
77                                              :
78     backing(&settings) {
79 }
```

12.7.2.4 CkbSettings::~CkbSettings ( )

Definition at line 145 of file ckbsettings.cpp.

References backing, removeCache, and writeCache.

```
145                           {
146     if(removeCache.isEmpty() && writeCache.isEmpty())
147         return;
148     // Save the settings from the settings thread.
149     // They have to be saved from that thread specifically to avoid performance issues
150     CkbSettingsWriter* writer = new CkbSettingsWriter(
      backing, removeCache, writeCache);
151     writer->moveToThread(globalThread);
152     QObject::staticMetaObject.invokeMethod(writer, "run", Qt::QueuedConnection);
153 }
```

12.7.3 Member Function Documentation

12.7.3.1 void CkbSettings::beginGroup (const QString & *prefix*)

Definition at line 81 of file ckbsettings.cpp.

References groups.

Referenced by CkbSettings(), KbAnim::save(), and SGroup::SGroup().

```
81                                                  {
82     groups.append(prefix);
83 }
```

Here is the caller graph for this function:

12.7.3.2 QStringList CkbSettings::childGroups ( ) const

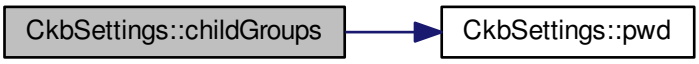
Definition at line 89 of file ckbsettings.cpp.

References backing, current, lockMutex, and pwd().

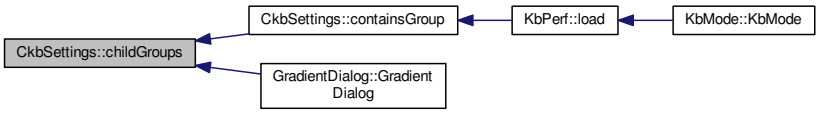
Referenced by containsGroup(), and GradientDialog::GradientDialog().

```
89                                              {
90     QString current = pwd();
91     lockMutex;
92     if(!current.isEmpty())
93         backing->beginGroup(current);
94     QStringList res = backing->childGroups();
95     if(!current.isEmpty())
96         backing->endGroup();
97     return res;
98 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.7.3.3 QStringList CkbSettings::childKeys ( ) const

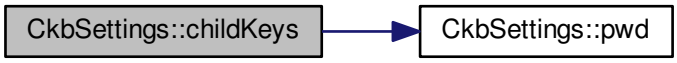
Definition at line 100 of file ckbsettings.cpp.

References backing, current, lockMutex, and pwd().

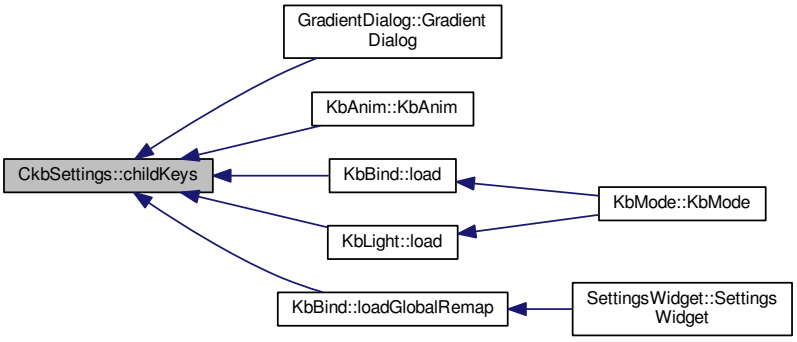
Referenced by GradientDialog::GradientDialog(), KbAnim::KbAnim(), KbBind::load(), KbLight::load(), and KbBind::loadGlobalRemap().

```
100                                         {
101     QString current = pwd();
102     lockMutex;
103     if(!current.isEmpty())
104         backing->beginGroup(current);
105     QStringList res = backing->childKeys();
106     if(!current.isEmpty())
107         backing->endGroup();
108     return res;
109 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.7.3.4 void CkbSettings::cleanUp ( ) [static]

Definition at line 46 of file ckbsettings.cpp.

References _globalSettings, cacheWritesInProgress(), and globalThread.

Referenced by MainWindow::cleanup().

```
46                          {
47     if(!_globalSettings)
48         return;
49     // Wait for all writers to finish
50     while(cacheWritesInProgress.load() > 0)
51         QThread::yieldCurrentThread();
52     // Stop thread and delete objects
53     globalThread->quit();
54     globalThread->wait();
55     delete globalThread;
56     delete _globalSettings;
57     globalThread = 0;
58     _globalSettings = 0;
59 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.7.3.5 bool CkbSettings::contains (const QString & *key*) const

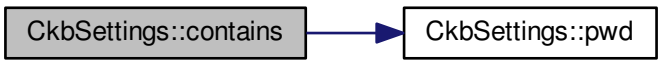
Definition at line 111 of file ckbsettings.cpp.

References backing, lockMutex, and pwd().

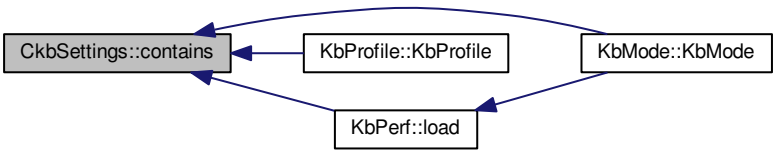
Referenced by KbMode::KbMode(), KbProfile::KbProfile(), and KbPerf::load().

```
111                                                 {
112     lockMutex;
113     return backing->contains(pwd(key));
114 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.7.3.6 bool CkbSettings::containsGroup (const QString & *group*)

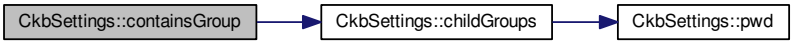
Definition at line 116 of file ckbsettings.cpp.

References childGroups().

Referenced by KbPerf::load().

```
116                                                 {
117     QStringList components = group.split("/");
118     if(components.length() > 1){
119         // Find sub-group
120         SGroup group(*this, components[0]);
121         return containsGroup(QStringList(components.mid(1)).join('/'));
122     }
123     return childGroups().contains(group);
124 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.7.3.7 void CkbSettings::endGroup ( )

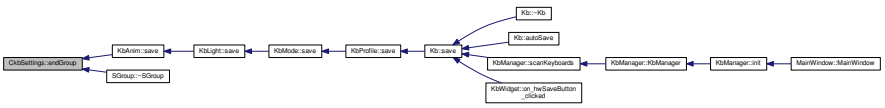
Definition at line 85 of file ckbsettings.cpp.

References groups.

Referenced by KbAnim::save(), and SGroup::~SGroup().

```
85                          {
86     groups.removeLast();
87 }
```

Here is the caller graph for this function:



12.7.3.8 QVariant CkbSettings::get (const QString & *key*, const QVariant & *defaultValue* = QVariant()) [static]

Definition at line 155 of file ckbsettings.cpp.

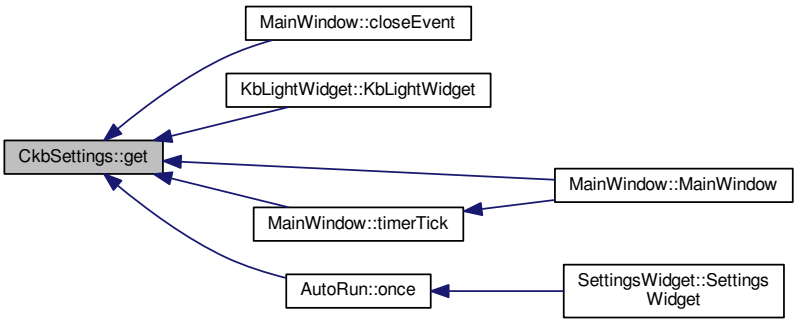
References globalCache, globalSettings(), lockMutexCache, and lockMutexStatic2.

Referenced by MainWindow::closeEvent(), KbLightWidget::KbLightWidget(), MainWindow::MainWindow(), AutoRun::once(), and MainWindow::timerTick().

```
155                                                                     {
156     // Store these settings in a memory cache
157     lockMutexCache;
158     if(globalCache.contains(key))
159         return globalCache.value(key);
160     // If it wasn't found in the memory cache, look for it in QSettings
161     lockMutexStatic2;
162     QSettings* settings = globalSettings();
163     return globalCache[key] = settings->value(key, defaultValue);
164 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.7.3.9 bool CkbSettings::isBusy ( ) [static]

Definition at line 42 of file ckbsettings.cpp.

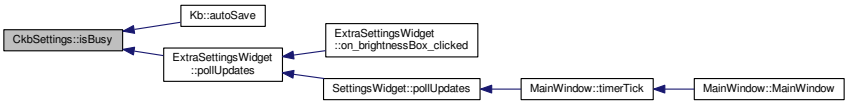
References cacheWritesInProgress().

Referenced by Kb::autoSave(), and ExtraSettingsWidget::pollUpdates().

```
42                         {
43     return cacheWritesInProgress.load() > 0;
44 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.7.3.10 QString CkbSettings::pwd ( ) const [inline], [private]

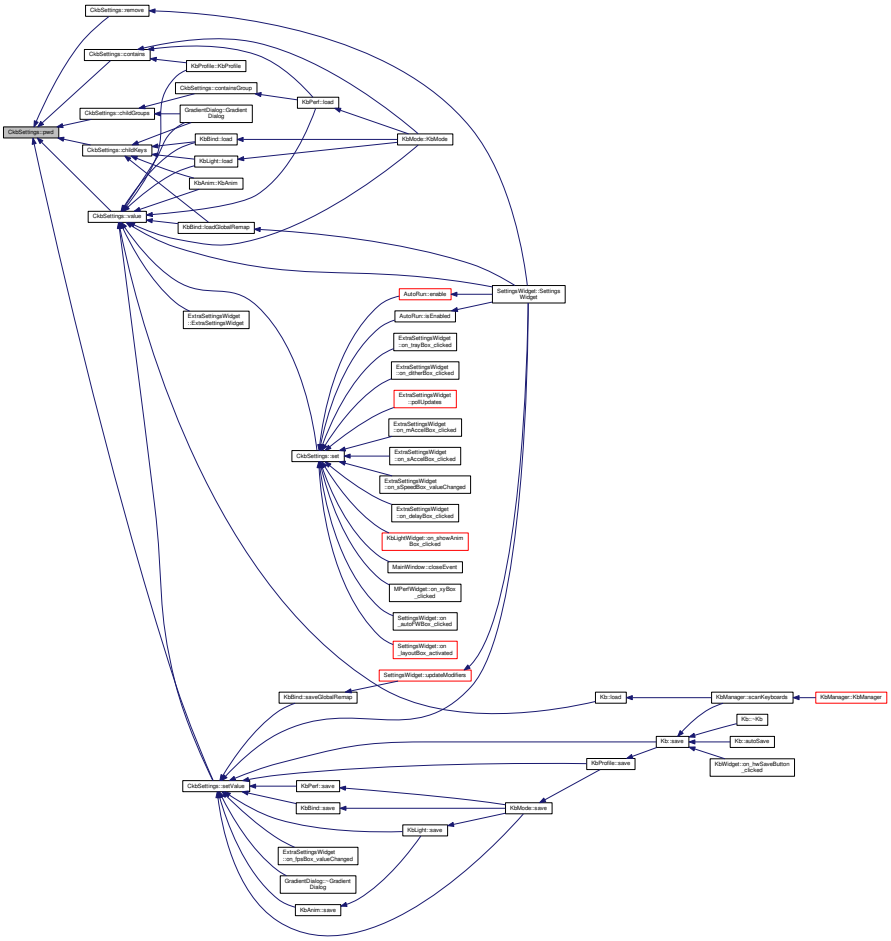
Definition at line 51 of file ckbsettings.h.

References groups.

Referenced by childGroups(), childKeys(), contains(), remove(), setValue(), and value().

```
51 { return groups.join("/"); }
```

Here is the caller graph for this function:



12.7.3.11 QString CkbSettings::pwd (const QString & *key*) const [inline], [private]

Definition at line 52 of file ckbsettings.h.

References groups, and pwd().

Referenced by pwd().

```
52 { return pwd() + (groups.isEmpty() ? "" : "/") + key; }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.7.3.12 void CkbSettings::remove (const QString & *key*)

Definition at line 141 of file ckbsettings.cpp.

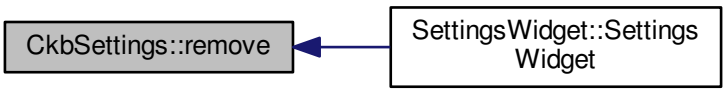
References pwd(), and removeCache.

Referenced by SettingsWidget::SettingsWidget().

```
141                                          {
142     removeCache.append(pwd(key));
143 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.7.3.13 void CkbSettings::set (const QString & *key*, const QVariant & *value*) [static]

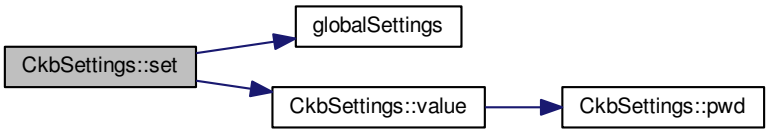
Definition at line 166 of file ckbsettings.cpp.

References globalCache, globalSettings(), lockMutexCache, lockMutexStatic, and value().

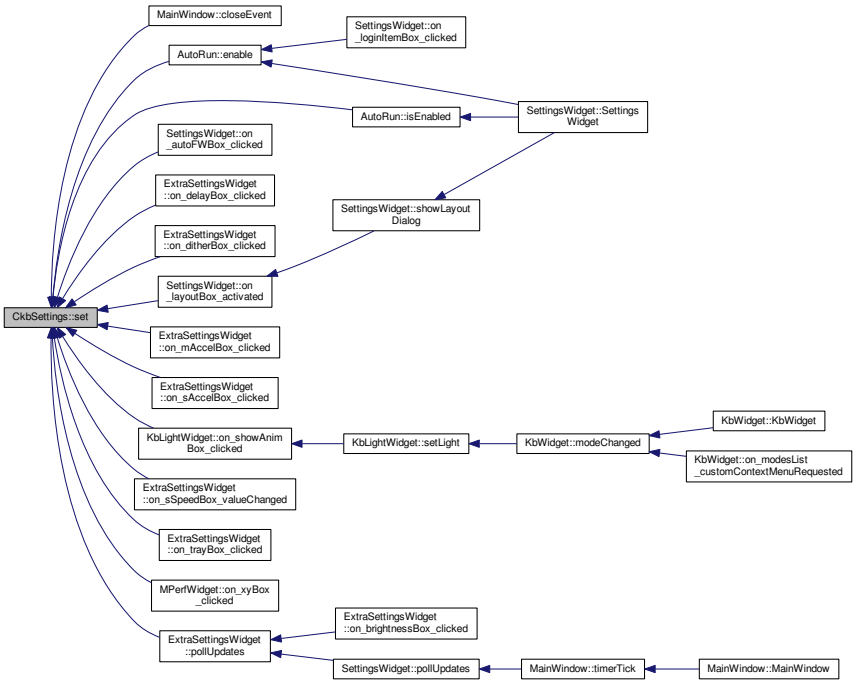
Referenced by MainWindow::closeEvent(), AutoRun::enable(), AutoRun::isEnabled(), SettingsWidget::on_autoFWBox_clicked(), ExtraSettingsWidget::on_delayBox_clicked(), ExtraSettingsWidget::on_ditherBox_clicked(), SettingsWidget::on_layoutBox_activated(), ExtraSettingsWidget::on_mAccelBox_clicked(), ExtraSettingsWidget::on_sAccelBox_clicked(), KbLightWidget::on_showAnimBox_clicked(), ExtraSettingsWidget::on_sSpeedBox_valueChanged(), ExtraSettingsWidget::on_trayBox_clicked(), MPerfWidget::on_xyBox_clicked(), and ExtraSettingsWidget::pollUpdates().

```
166                                                                 {
167     {
168         lockMutexCache;
169         if(globalCache.value(key) == value)
170             return;
171         globalCache[key] = value;
172     }
173     lockMutexStatic;
174     globalSettings()->setValue(key, value);
175 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.7.3.14 void CkbSettings::setValue (const QString & *key,* const QVariant & *value*)

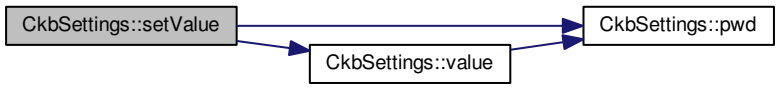
Definition at line 131 of file ckbsettings.cpp.

References globalCache, lockMutexCache, pwd(), value(), and writeCache.

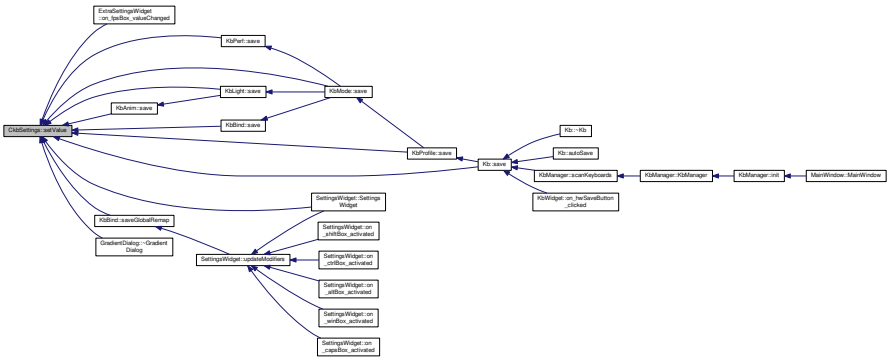
Referenced by ExtraSettingsWidget::on_fpsBox_valueChanged(), KbProfile::save(), KbPerf::save(), KbAnim::save(), KbBind::save(), KbMode::save(), KbLight::save(), Kb::save(), KbBind::saveGlobalRemap(), SettingsWidget::SettingsWidget(), and GradientDialog::~GradientDialog().

```
131                                                                 {
132     // Cache the write values, save them when the object is destroyed
133     QString realKey = pwd(key);
134     writeCache[realKey] = value;
135     // Update global cache if needed
136     lockMutexCache;
137     if(globalCache.contains(realKey))
138         globalCache[realKey] = value;
139 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.7.3.15 QVariant CkbSettings::value (const QString & *key,* const QVariant & *defaultValue* = `QVariant()`) const

Definition at line 126 of file ckbsettings.cpp.

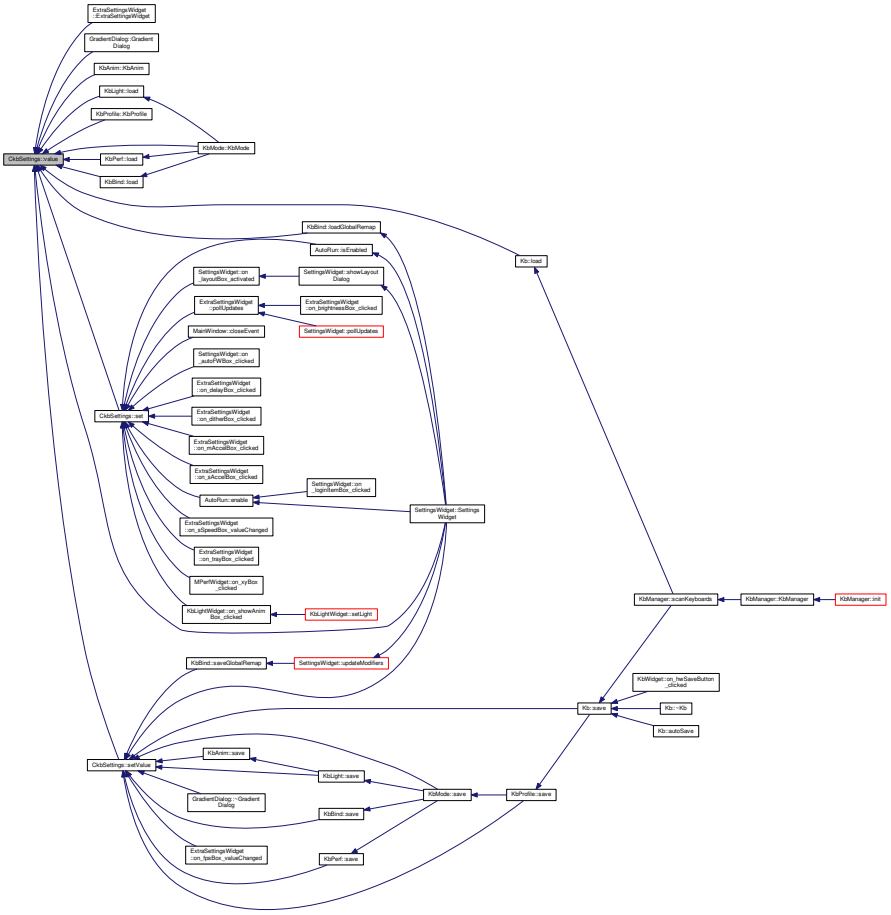
References backing, lockMutex, and pwd().

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), GradientDialog::GradientDialog(), KbAnim::KbAnim(), KbMode::KbMode(), KbProfile::KbProfile(), KbPerf::load(), KbBind::load(), KbLight::load(), Kb::load(), KbBind::loadGlobalRemap(), set(), SettingsWidget::SettingsWidget(), and setValue().

```
126                                                                 {
127     lockMutex;
128     return backing->value(pwd(key), defaultValue);
129 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.7.4 Field Documentation

12.7.4.1 QSettings* CkbSettings::backing [private]

Definition at line 46 of file ckbsettings.h.

Referenced by childGroups(), childKeys(), contains(), value(), and ~CkbSettings().

12.7.4.2 QStringList CkbSettings::groups [private]

Definition at line 47 of file ckbsettings.h.

Referenced by beginGroup(), endGroup(), and pwd().

12.7.4.3 QStringList CkbSettings::removeCache [private]

Definition at line 48 of file ckbsettings.h.

Referenced by remove(), and ~CkbSettings().

12.7.4.4 QMap<QString, QVariant> CkbSettings::writeCache [private]

Definition at line 49 of file ckbsettings.h.

Referenced by setValue(), and ~CkbSettings().

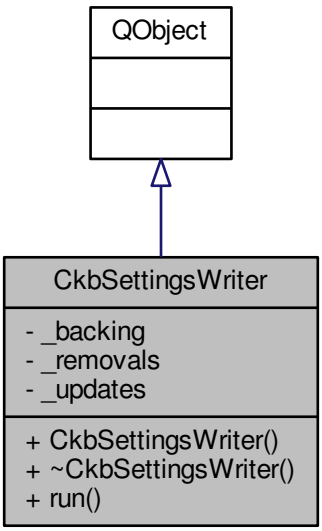
The documentation for this class was generated from the following files:

- src/ckb/ckbsettings.h
- src/ckb/ckbsettings.cpp

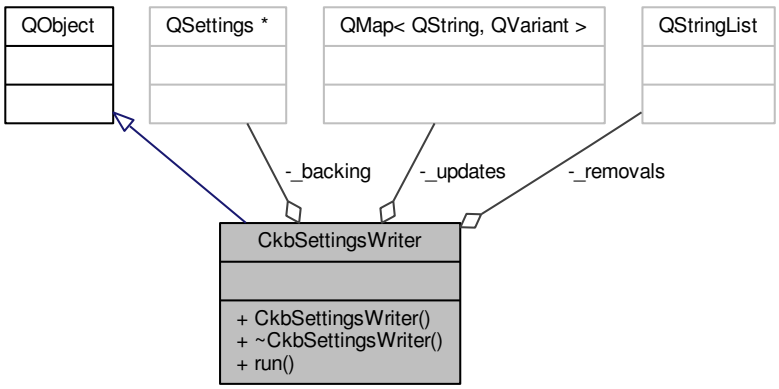
12.8 CkbSettingsWriter Class Reference

`#include <src/ckb/ckbsettingswriter.h>`

Inheritance diagram for CkbSettingsWriter:



Collaboration diagram for CkbSettingsWriter:



Public Member Functions

- CkbSettingsWriter (QSettings ∗backing, const QStringList &removals, const QMap< QString, QVariant > &updates)
- ∼CkbSettingsWriter ()
- Q_SLOT void run ()

Private Attributes

- QSettings ∗ _backing
- QStringList _removals
- QMap< QString, QVariant > _updates

12.8.1 Detailed Description

Definition at line 12 of file ckbsettingswriter.h.

12.8.2 Constructor & Destructor Documentation

12.8.2.1 CkbSettingsWriter::CkbSettingsWriter (QSettings ∗ *backing*, const QStringList & *removals*, const QMap< QString, QVariant > & *updates*)

Definition at line 12 of file ckbsettingswriter.cpp.

References cacheWritesInProgress.

```
12
                    :
13     _backing(backing), _removals(removals), _updates(updates) {
14     cacheWritesInProgress.ref();
15 }
```

12.8.2.2 CkbSettingsWriter::~CkbSettingsWriter ( )

Definition at line 17 of file ckbsettingswriter.cpp.

References cacheWritesInProgress.

```
17                                      {
18     cacheWritesInProgress.deref();
19 }
```

12.8.3 Member Function Documentation

12.8.3.1 void CkbSettingsWriter::run ( )

Definition at line 21 of file ckbsettingswriter.cpp.

References _backing, _removals, _updates, and lockMutexStatic.

```
21                                {
22     lockMutexStatic;
23     // Process key removals
24     foreach(const QString& rm, _removals){
25         _backing->remove(rm);
26     }
27     // Process writes
28     QMapIterator<QString, QVariant> i(_updates);
29     while(i.hasNext()){
30         i.next();
31         _backing->setValue(i.key(), i.value());
32         // Updating the global cache was done above
33     }
34     _backing->sync();
35     deleteLater();
36 }
```

12.8.4 Field Documentation

12.8.4.1 QSettings* CkbSettingsWriter::_backing [private]

Definition at line 21 of file ckbsettingswriter.h.

Referenced by run().

12.8.4.2 QStringList CkbSettingsWriter::_removals [private]

Definition at line 22 of file ckbsettingswriter.h.

Referenced by run().

12.8.4.3 QMap<QString, QVariant> CkbSettingsWriter::_updates [private]

Definition at line 23 of file ckbsettingswriter.h.

Referenced by run().

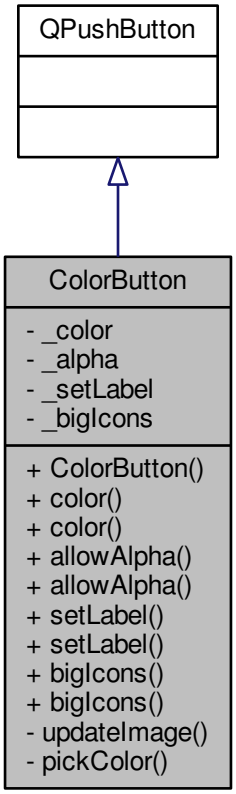
The documentation for this class was generated from the following files:

- src/ckb/ckbsettingswriter.h
- src/ckb/ckbsettingswriter.cpp

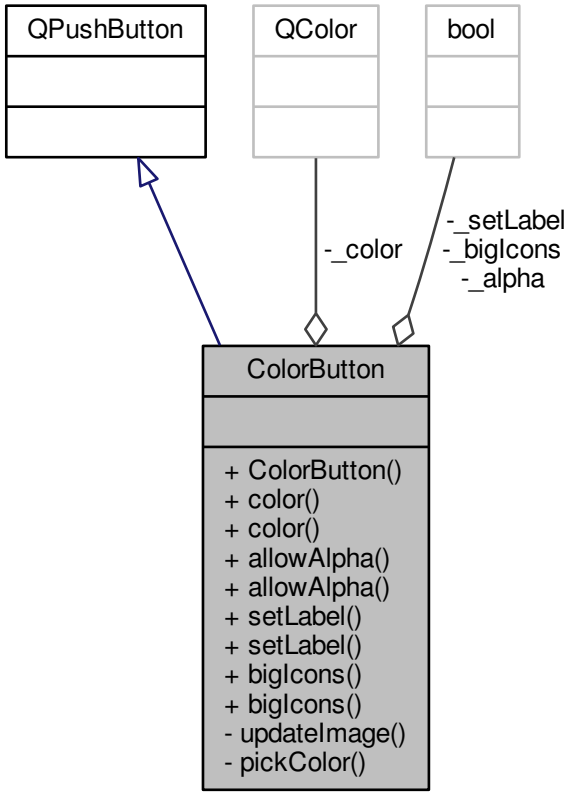
12.9 ColorButton Class Reference

`#include <src/ckb/colorbutton.h>`

Inheritance diagram for ColorButton:



Collaboration diagram for ColorButton:



Signals

- void colorChanged (QColor newColor)

Public Member Functions

- ColorButton (QWidget *parent=0, bool allowAlpha=false)
- const QColor & color () const
- void color (const QColor &newColor)
- bool allowAlpha () const
- void allowAlpha (bool newAllowAlpha)
- bool setLabel () const
- void setLabel (bool newSetLabel)
- bool bigIcons () const
- void bigIcons (bool newBigIcons)

Private Slots

- void pickColor ()

Private Member Functions

- void updateImage ()

Private Attributes

- QColor _color
- bool _alpha
- bool _setLabel
- bool _bigIcons

12.9.1 Detailed Description

Definition at line 6 of file colorbutton.h.

12.9.2 Constructor & Destructor Documentation

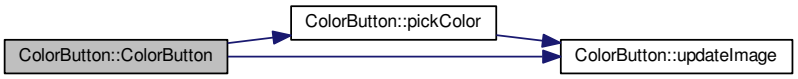
12.9.2.1 ColorButton::ColorButton (QWidget * *parent* = 0, bool *allowAlpha* = false) [explicit]

Definition at line 5 of file colorbutton.cpp.

References pickColor(), and updateImage().

```
5                                                       :
6     QPushButton(parent), _alpha(allowAlpha),
      _setLabel(true), _bigIcons(false)
7 {
8     setAutoDefault(false);
9     setDefault(false);
10     updateImage();
11     // Pick color on click (use queued connection so that any on_*_clicked() events can be processed first)
12     connect(this, SIGNAL(clicked()), this, SLOT(pickColor()), Qt::QueuedConnection);
13 }
```

Here is the call graph for this function:



12.9.3 Member Function Documentation

12.9.3.1 bool ColorButton::allowAlpha ( ) const [inline]

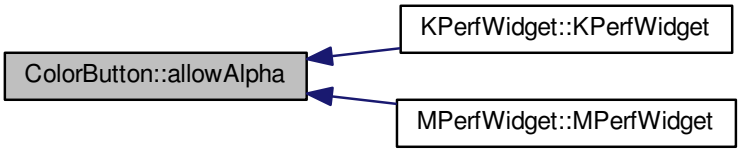
Definition at line 17 of file colorbutton.h.

References _alpha.

Referenced by KPerfWidget::KPerfWidget(), and MPerfWidget::MPerfWidget().

```
17 { return _alpha; }
```

Here is the caller graph for this function:



12.9.3.2 void ColorButton::allowAlpha (bool *newAllowAlpha*)

Definition at line 22 of file colorbutton.cpp.

References _alpha, and updateImage().

```
22                                              {
23     _alpha = newAllowAlpha;
24     updateImage();
25 }
```

Here is the call graph for this function:

12.9.3.3 bool ColorButton::bigIcons ( ) const [inline]

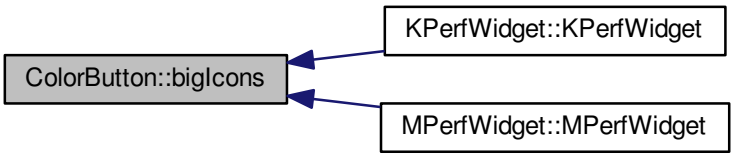
Definition at line 23 of file colorbutton.h.

References _bigIcons.

Referenced by KPerfWidget::KPerfWidget(), and MPerfWidget::MPerfWidget().

```
23 { return _bigIcons; }
```

Here is the caller graph for this function:



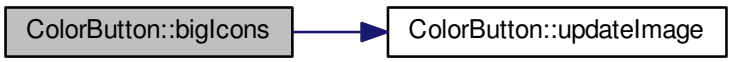
12.9.3.4 void ColorButton::bigIcons (bool *newBigIcons*)

Definition at line 27 of file colorbutton.cpp.

References _bigIcons, and updateImage().

```
27                                          {
28     _bigIcons = newBigIcons;
29     updateImage();
30 }
```

Here is the call graph for this function:



12.9.3.5 const QColor& ColorButton::color ( ) const [inline]

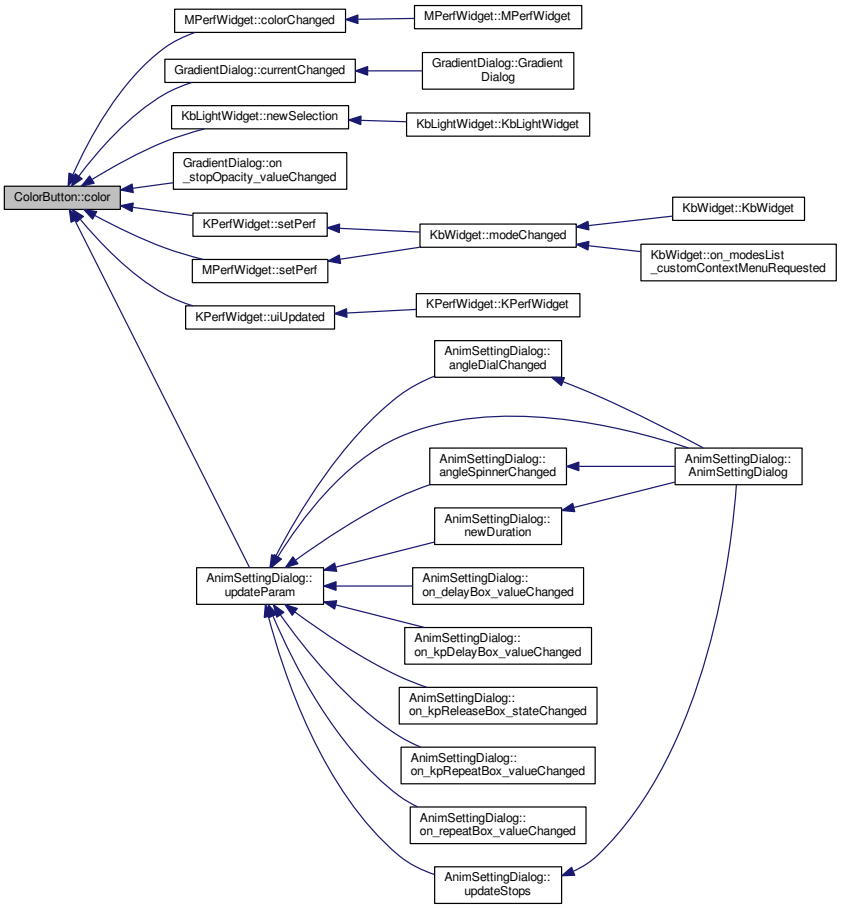
Definition at line 13 of file colorbutton.h.

References _color.

Referenced by MPerfWidget::colorChanged(), GradientDialog::currentChanged(), KbLightWidget::newSelection(), GradientDialog::on_stopOpacity_valueChanged(), KPerfWidget::setPerf(), MPerfWidget::setPerf(), KPerfWidget::uiUpdated(), and AnimSettingDialog::updateParam().

```
13 { return _color; }
```

Here is the caller graph for this function:



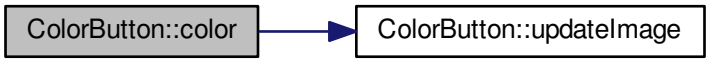
12.9.3.6 void ColorButton::color (const QColor & *newColor*)

Definition at line 15 of file colorbutton.cpp.

References _alpha, _color, and updateImage().

```
15                                          {
16     _color = newColor;
17     if(!_alpha)
18         _color.setAlpha(255);
19     updateImage();
20 }
```

Here is the call graph for this function:



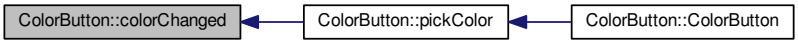
12.9.3.7 void ColorButton::colorChanged (QColor *newColor*) [signal]

Definition at line 132 of file moc_colorbutton.cpp.

Referenced by pickColor().

```
133 {
134     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
135     QMetaObject::activate(this, &staticMetaObject, 0, _a);
136 }
```

Here is the caller graph for this function:



12.9.3.8 void ColorButton::pickColor ( ) [private], [slot]

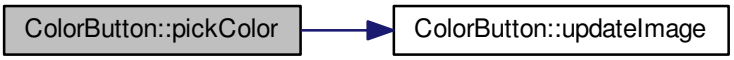
Definition at line 60 of file colorbutton.cpp.

References _alpha, _color, colorChanged(), and updateImage().

Referenced by ColorButton().

```
60                            {
61     QColor newColor = QColorDialog::getColor(_color, this, QString(),
      QColorDialog::ColorDialogOptions(_alpha ? QColorDialog::ShowAlphaChannel : 0));
62     if(newColor.isValid()){
63         _color = newColor;
64         updateImage();
65         emit colorChanged(_color);
66     }
67 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.9.3.9 bool ColorButton::setLabel ( ) const [inline]

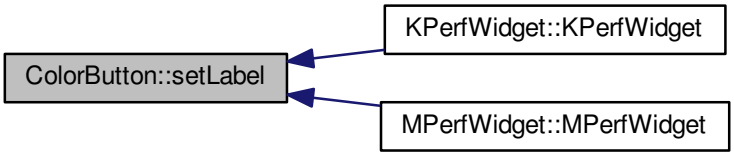
Definition at line 20 of file colorbutton.h.

References _setLabel.

Referenced by KPerfWidget::KPerfWidget(), and MPerfWidget::MPerfWidget().

```
20 { return _setLabel; }
```

Here is the caller graph for this function:



12.9.3.10 void ColorButton::setLabel (bool *newSetLabel*) [inline]

Definition at line 21 of file colorbutton.h.

References _setLabel.

```
21 { _setLabel = newSetLabel; }
```

12.9.3.11 void ColorButton::updateImage ( ) [private]

Definition at line 32 of file colorbutton.cpp.

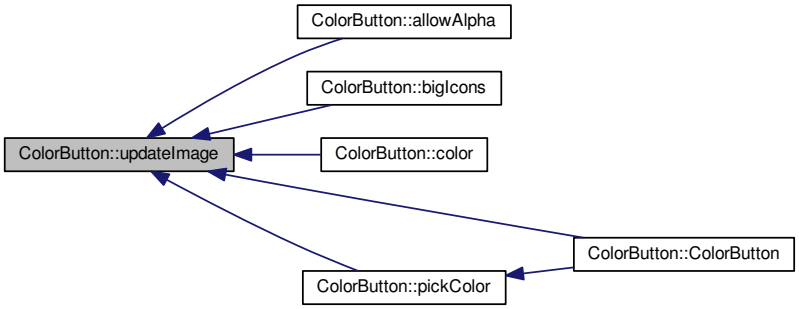
References _alpha, _bigIcons, _color, and _setLabel.

Referenced by allowAlpha(), bigIcons(), color(), ColorButton(), and pickColor().

```
32                              {
33     if(!_color.isValid()){
34         setIcon(QIcon());
35         if(_setLabel)
36             setText("Change color...");
37         return;
38     }
39     const int w = 24, h = _bigIcons ? 24 : 12;
40     QImage image(w, h, QImage::Format_RGB888);
41     QPainter painter(&image);
42     painter.setPen(Qt::NoPen);
43     painter.fillRect(0, 0, w, h, QColor(0, 0, 0));
44     if(_alpha && _color.alpha() != 255){
45         painter.fillRect(1, 1, w / 2 - 1, h / 2 - 1, QColor(255, 255, 255));
46         painter.fillRect(w / 2, 1, w / 2 - 1, h / 2 - 1, QColor(192, 192, 192));
47         painter.fillRect(1, h / 2, w / 2 - 1, h / 2 - 1, QColor(192, 192, 192));
48         painter.fillRect(w / 2, h / 2, w / 2 - 1, h / 2 - 1, QColor(255, 255, 255));
49     }
50     painter.fillRect(1, 1, w - 2, h - 2, _color);
51     setIcon(QIcon(QPixmap::fromImage(image)));
52     if(_setLabel){
53         QString text = QString(" (%1, %2, %3)").arg(_color.red()).arg(
       _color.green()).arg(_color.blue());
54         if(_alpha)
```

```
55             text += QString(", %4%").arg(QString::number(_color.alphaF() * 100., 'f', 0));
56         setText(text);
57     }
58 }
```

Here is the caller graph for this function:



12.9.4 Field Documentation

12.9.4.1 bool ColorButton::_alpha [private]

Definition at line 34 of file colorbutton.h.

Referenced by allowAlpha(), color(), pickColor(), and updateImage().

12.9.4.2 bool ColorButton::_bigIcons [private]

Definition at line 34 of file colorbutton.h.

Referenced by bigIcons(), and updateImage().

12.9.4.3 QColor ColorButton::_color [private]

Definition at line 33 of file colorbutton.h.

Referenced by color(), pickColor(), and updateImage().

12.9.4.4 bool ColorButton::_setLabel [private]

Definition at line 34 of file colorbutton.h.

Referenced by setLabel(), and updateImage().

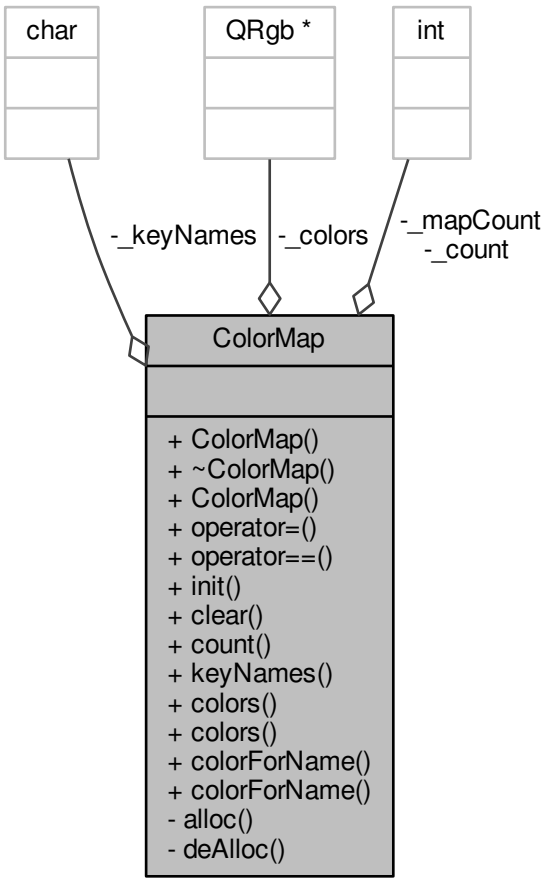
The documentation for this class was generated from the following files:

- src/ckb/colorbutton.h
- src/ckb/colorbutton.cpp
- src/ckb/release/moc_colorbutton.cpp

12.10 ColorMap Class Reference

`#include <src/ckb/colormap.h>`

Collaboration diagram for ColorMap:



Public Member Functions

- ColorMap ()
- ~ColorMap ()
- ColorMap (const ColorMap &rhs)
- const ColorMap & operator= (const ColorMap &rhs)
- bool operator== (const ColorMap &rhs) const
- void init (const KeyMap &map)
- void clear ()
- int count () const
- const char *const * keyNames () const
- QRgb * colors ()
- const QRgb * colors () const
- QRgb * colorForName (const char *name)
- const QRgb * colorForName (const char *name) const

Private Member Functions

- void alloc (int count)
- void deAlloc ()

Private Attributes

- const char ** _keyNames
- QRgb * _colors
- int _count
- int _mapCount

12.10.1 Detailed Description

Definition at line 15 of file colormap.h.

12.10.2 Constructor & Destructor Documentation

12.10.2.1 ColorMap::ColorMap ( )

Definition at line 3 of file colormap.cpp.

```
3                   :
4     _keyNames(0), _colors(0), _count(0), _mapCount(0)
5 {
6 }
```

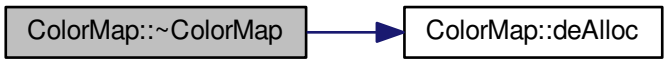
12.10.2.2 ColorMap::~ColorMap ( )

Definition at line 8 of file colormap.cpp.

References deAlloc().

```
8                   {
9     deAlloc();
10 }
```

Here is the call graph for this function:



12.10.2.3 ColorMap::ColorMap (const ColorMap & *rhs*)

Definition at line 12 of file colormap.cpp.

```
12                                  :
13     _keyNames(0), _colors(0), _count(0), _mapCount(0)
14 {
15     *this = rhs;
16 }
```

12.10.3 Member Function Documentation

12.10.3.1 void ColorMap::alloc (int *count*) [private]

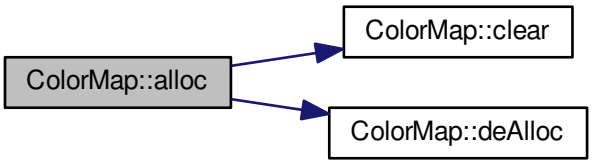
Definition at line 43 of file colormap.cpp.

References _colors, _count, _keyNames, _mapCount, clear(), and deAlloc().

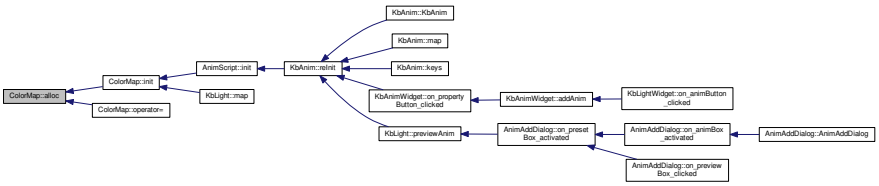
Referenced by init(), and operator=().

```
43                                {
44     if(newCount > _mapCount){
45         // ^ map never shrinks, only expands
46         deAlloc();
47         _mapCount = newCount;
48         _keyNames = new const char*[_mapCount];
49         _colors = new QRgb[_mapCount];
50     }
51     _count = newCount;
52     clear();
53 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.10.3.2 void ColorMap::clear ( )

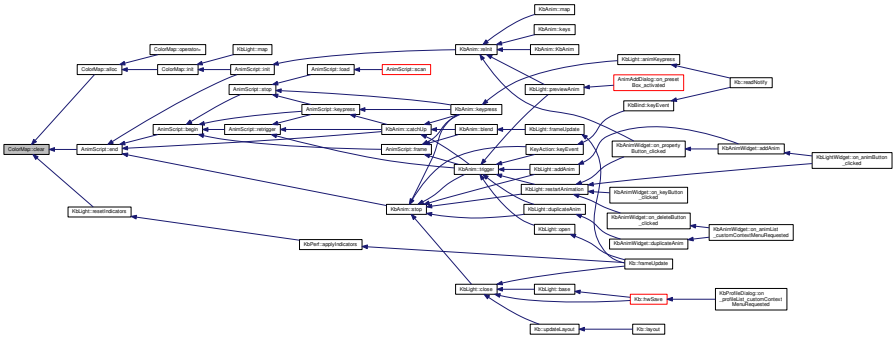
Definition at line 61 of file colormap.cpp.

References _colors, and _count.

Referenced by alloc(), AnimScript::end(), and KbLight::resetIndicators().

```
61                      {
62     memset(_colors, 0, _count * sizeof(QRgb));
63 }
```

Here is the caller graph for this function:



12.10.3.3 QRgb * ColorMap::colorForName (const char * *name*)

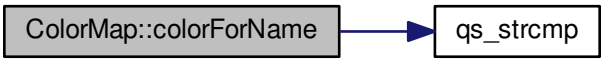
Definition at line 88 of file colormap.cpp.

References _colors, _count, _keyNames, and qs_strcmp().

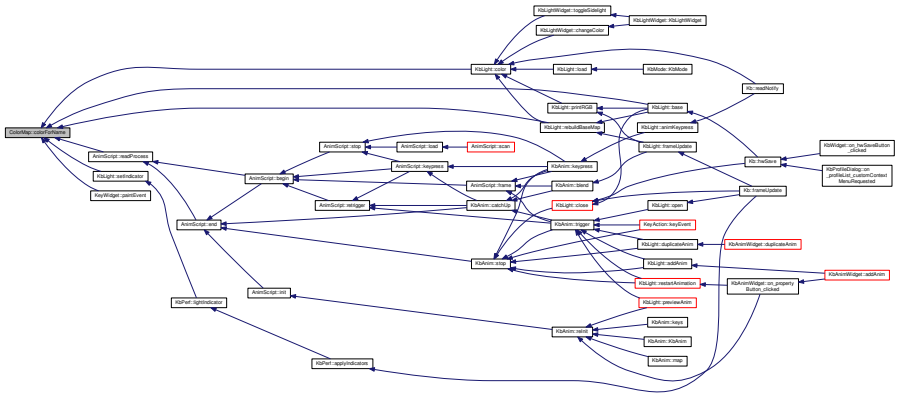
Referenced by KbLight::base(), KbLight::color(), KeyWidget::paintEvent(), AnimScript::readProcess(), KbLight::rebuildBaseMap(), and KbLight::setIndicator().

```
88                                          {
89     if(!_keyNames)
90         return 0;
91     const char** namePtr = (const char**)bsearch(&name, _keyNames,
      _count, sizeof(const char*), qs_strcmp);
92     if(!namePtr)
93         return 0;
94     ptrdiff_t position = namePtr - _keyNames;
95     return _colors + position;
96 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



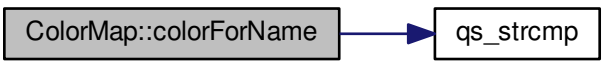
12.10.3.4 const QRgb * ColorMap::colorForName (const char * *name*) const

Definition at line 98 of file colormap.cpp.

References _colors, _count, _keyNames, and qs_strcmp().

```
98                                                          {
99     if(!_keyNames)
100         return 0;
101     const char** namePtr = (const char**)bsearch(&name, _keyNames,
    _count, sizeof(const char*), qs_strcmp);
102     if(!namePtr)
103         return 0;
104     ptrdiff_t position = namePtr - _keyNames;
105     return _colors + position;
106 }
```

Here is the call graph for this function:



12.10.3.5 QRgb* ColorMap::colors ( ) [inline]

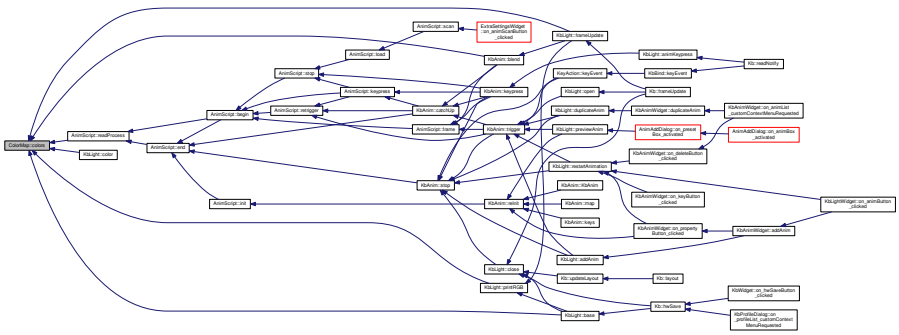
Definition at line 33 of file colormap.h.

References _colors.

Referenced by KbLight::base(), KbAnim::blend(), KbLight::color(), KbLight::frameUpdate(), KbLight::printRGB(), and AnimScript::readProcess().

```
33 { return _colors; }
```

Here is the caller graph for this function:



12.10.3.6 const QRgb* ColorMap::colors ( ) const [inline]

Definition at line 34 of file colormap.h.

References _colors.

```
34 { return _colors; }
```

12.10.3.7 int ColorMap::count ( ) const [inline]

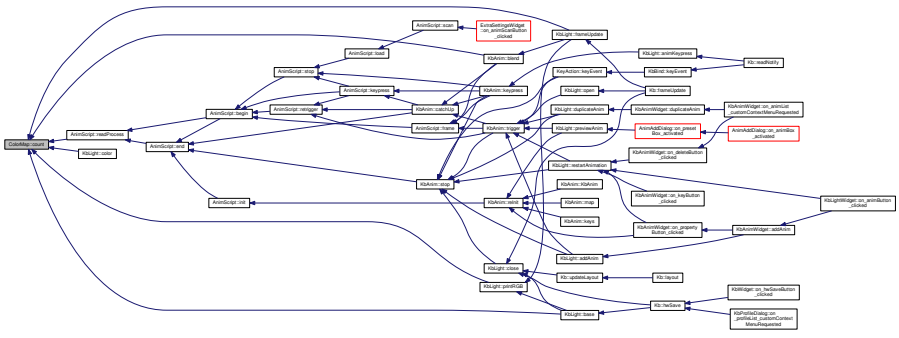
Definition at line 31 of file colormap.h.

References _count.

Referenced by KbLight::base(), KbAnim::blend(), KbLight::color(), KbLight::frameUpdate(), KbLight::printRGB(), and AnimScript::readProcess().

```
31 { return _count; }
```

Here is the caller graph for this function:



12.10.3.8 void ColorMap::deAlloc ( ) [private]

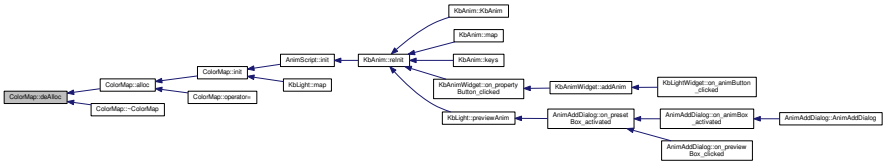
Definition at line 55 of file colormap.cpp.

References _colors, _count, _keyNames, and _mapCount.

Referenced by alloc(), and ~ColorMap().

```
55                      {
56     _count = _mapCount = 0;
57     if(_keyNames) delete[] _keyNames;
58     if(_colors) delete[] _colors;
59 }
```

Here is the caller graph for this function:



12.10.3.9 void ColorMap::init (const KeyMap & *map*)

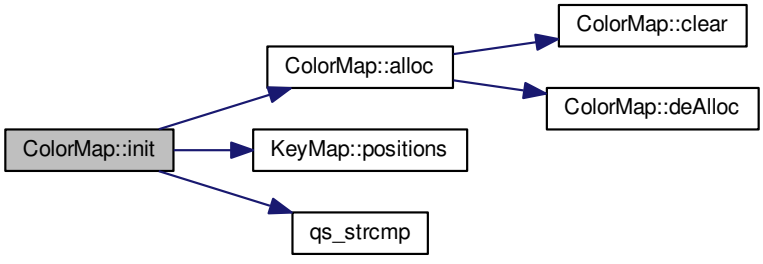
Definition at line 69 of file colormap.cpp.

References _count, _keyNames, alloc(), Key::hasLed, Key::name, KeyMap::positions(), and qs_strcmp().

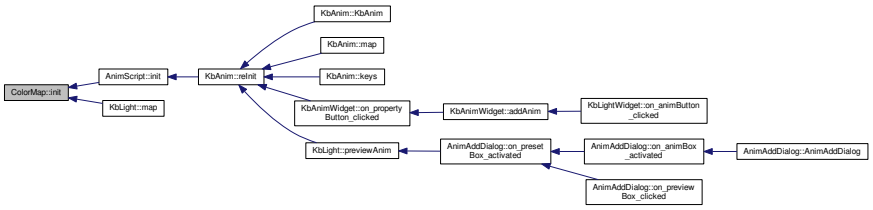
Referenced by AnimScript::init(), and KbLight::map().

```
69                                      {
70     QList<Key> newKeys = map.positions();
71     // There's no point including keys that don't have LEDs, so remove them now
72     QMutableListIterator<Key> i(newKeys);
73     while(i.hasNext()){
74         Key key = i.next();
75         if(!key.hasLed)
76             i.remove();
77     }
78     // Now that we know how many keys we'll have, check memory
79     alloc(newKeys.count());
80     // Copy key names
81     int keyPos = 0;
82     foreach(const Key& key, newKeys)
83         _keyNames[keyPos++] = key.name;      // as above, it's safe to copy these since the
       strings are constants
84     // Sort names for quick access
85     qsort(_keyNames, _count, sizeof(const char*), qs_strcmp);
86 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.10.3.10 const char* const* ColorMap::keyNames ( ) const [inline]

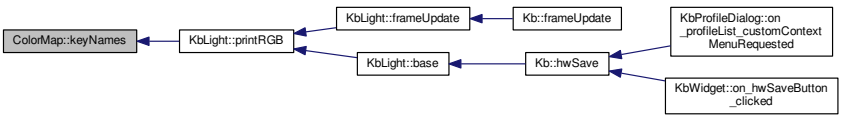
Definition at line 32 of file colormap.h.

References _keyNames.

Referenced by KbLight::printRGB().

```
32 { return _keyNames; }
```

Here is the caller graph for this function:



12.10.3.11 const ColorMap & ColorMap::operator= (const ColorMap & *rhs*)

Definition at line 18 of file colormap.cpp.

References _colors, _count, _keyNames, and alloc().

```
18                                                  {
19     alloc(rhs._count);
20     // Copy key names and colors
21     // (Note: it's safe to copy the const char*'s because key names are constants and will never move or be
       deleted)
22     memcpy(_keyNames, rhs._keyNames, sizeof(const char*) *
      _count);
23     memcpy(_colors, rhs._colors, sizeof(QRgb) * _count);
24     return rhs;
25 }
```

Here is the call graph for this function:

12.10.3.12 bool ColorMap::operator== (const ColorMap & *rhs*) const

Definition at line 27 of file colormap.cpp.

References _colors, _count, _keyNames, and x.

```
27                                                  {
28     if(_count != rhs._count)
29         return false;
30
31     // Compare just the pointers because key names are sorted and the strings are constants.
32     if(memcmp(_keyNames, rhs._keyNames, sizeof(const char*) *
       _count) != 0)
33         return false;
34
35     for(int x = 0;x < _count;x++){
36         if(_colors[x] != rhs._colors[x])
37             return false;
38     }
39
40     return true;
41 }
```

12.10.4 Field Documentation

12.10.4.1 QRgb* ColorMap::_colors [private]

Definition at line 45 of file colormap.h.

Referenced by alloc(), clear(), colorForName(), colors(), deAlloc(), operator=(), and operator==().

12.10.4.2 int ColorMap::_count [private]

Definition at line 46 of file colormap.h.

Referenced by alloc(), clear(), colorForName(), count(), deAlloc(), init(), operator=(), and operator==().

12.10.4.3 const char** ColorMap::_keyNames [private]

Definition at line 44 of file colormap.h.

Referenced by alloc(), colorForName(), deAlloc(), init(), keyNames(), operator=(), and operator==().

12.10.4.4 int ColorMap::_mapCount [private]

Definition at line 46 of file colormap.h.

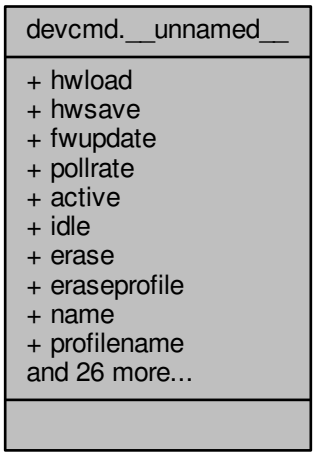
Referenced by alloc(), and deAlloc().

The documentation for this class was generated from the following files:

- src/ckb/colormap.h
- src/ckb/colormap.cpp

12.11 devcmd.__unnamed__ Struct Reference

Collaboration diagram for devcmd.__unnamed__:



Data Fields

- cmdhandler_io hwload
- cmdhandler_io hwsave
- cmdhandler_io fwupdate
- cmdhandler_io pollrate
- cmdhandler_io active
- cmdhandler_io idle
- cmdhandler erase
- cmdhandler eraseprofile
- cmdhandler name
- cmdhandler profilename
- cmdhandler id
- cmdhandler profileid
- cmdhandler rgb
- cmdhandler ioff
- cmdhandler ion
- cmdhandler iauto
- cmdhandler bind
- cmdhandler unbind
- cmdhandler rebind
- cmdhandler_mac macro

- cmdhandler_mac dpi
- cmdhandler dpisel
- cmdhandler lift
- cmdhandler snap
- cmdhandler notify
- cmdhandler inotify
- cmdhandler get
- cmdhandler restart
- int(∗ start)(usbdevice ∗kb, int makeactive)
- void(∗ setmodeindex)(usbdevice ∗kb, int index)
- void(∗ allocprofile)(usbdevice ∗kb)
- int(∗ loadprofile)(usbdevice ∗kb)
- void(∗ freeprofile)(usbdevice ∗kb)
- int(∗ updatergb)(usbdevice ∗kb, int force)
- void(∗ updateindicators)(usbdevice ∗kb, int force)
- int(∗ updatedpi)(usbdevice ∗kb, int force)

12.11.1 Detailed Description

Definition at line 78 of file command.h.

12.11.2 Field Documentation

12.11.2.1

12.11.2.2

12.11.2.3

12.11.2.4

12.11.2.5

12.11.2.6

12.11.2.7

12.11.2.8

12.11.2.9

12.11.2.10

12.11.2.11

12.11.2.12

12.11.2.13

12.11.2.14

12.11.2.15

12.11.2.16

12.11.2.17

12.11.2.18

12.11.2.19

12.11.2.20

12.11.2.21

12.11.2.22

12.11.2.23

12.11.2.24

12.11.2.25

12.11.2.26

12.11.2.27

12.11.2.28

12.11.2.29

12.11.2.30

12.11.2.31

12.11.2.32

12.11.2.33

12.11.2.34

12.11.2.35

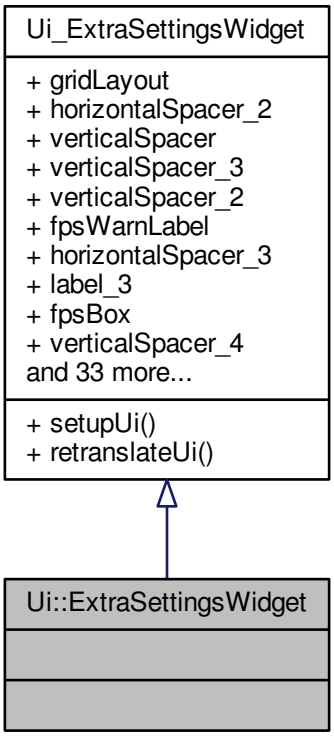
12.11.2.36

The documentation for this struct was generated from the following files:

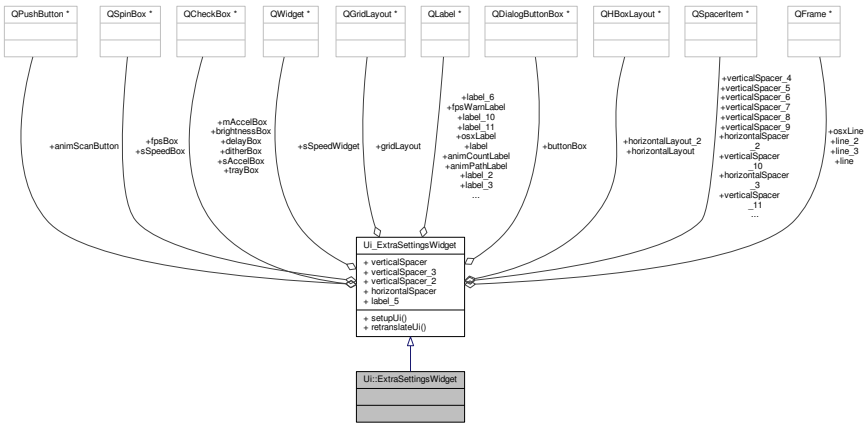
12.12 Ui::ExtraSettingsWidget Class Reference

`#include <src/ckb/ui_extrasettingswidget.h>`

Inheritance diagram for Ui::ExtraSettingsWidget:



Collaboration diagram for Ui::ExtraSettingsWidget:



Public Member Functions

- void setupUi (QDialog *ExtraSettingsWidget)
- void retranslateUi (QDialog *ExtraSettingsWidget)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer_2
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_3
- QSpacerItem * verticalSpacer_2
- QLabel * fpsWarnLabel
- QSpacerItem * horizontalSpacer_3
- QLabel * label_3
- QSpinBox * fpsBox
- QSpacerItem * verticalSpacer_4
- QSpacerItem * verticalSpacer_9
- QLabel * label_6
- QSpacerItem * verticalSpacer_6
- QSpacerItem * verticalSpacer_5
- QCheckBox * sAccelBox
- QSpacerItem * verticalSpacer_7
- QSpacerItem * verticalSpacer_8
- QSpacerItem * verticalSpacer_10
- QSpacerItem * verticalSpacer_11
- QSpacerItem * horizontalSpacer
- QFrame * line_2
- QHBoxLayout * horizontalLayout
- QLabel * animPathLabel
- QLabel * animCountLabel
- QPushButton * animScanButton
- QCheckBox * brightnessBox
- QCheckBox * trayBox
- QLabel * label
- QFrame * line
- QLabel * label_5
- QFrame * line_3
- QLabel * label_2
- QCheckBox * ditherBox
- QCheckBox * delayBox
- QLabel * osxLabel
- QFrame * osxLine
- QCheckBox * mAccelBox
- QWidget * sSpeedWidget
- QHBoxLayout * horizontalLayout_2
- QLabel * label_10
- QSpinBox * sSpeedBox
- QLabel * label_11
- QDialogButtonBox * buttonBox

12.12.1 Detailed Description

Definition at line 374 of file ui_extrasettingswidget.h.

12.12.2 Member Function Documentation

12.12.2.1 void Ui_ExtraSettingsWidget::retranslateUi (QDialog * *ExtraSettingsWidget*) [inline],[inherited]

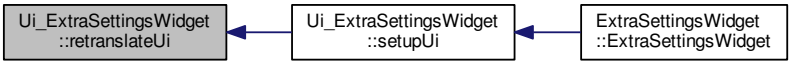
Definition at line 330 of file ui_extrasettingswidget.h.

References Ui_ExtraSettingsWidget::animCountLabel, Ui_ExtraSettingsWidget::animPathLabel, Ui_ExtraSettingsWidget::animScanButton, Ui_ExtraSettingsWidget::brightnessBox, Ui_ExtraSettingsWidget::delayBox, Ui_ExtraSettingsWidget::ditherBox, Ui_ExtraSettingsWidget::fpsWarnLabel, Ui_ExtraSettingsWidget::label, Ui_ExtraSettingsWidget::label_10, Ui_ExtraSettingsWidget::label_11, Ui_ExtraSettingsWidget::label_2, Ui_ExtraSettingsWidget::label_3, Ui_ExtraSettingsWidget::label_5, Ui_ExtraSettingsWidget::label_6, Ui_ExtraSettingsWidget::mAccelBox, Ui_ExtraSettingsWidget::osxLabel, Ui_ExtraSettingsWidget::sAccelBox, and Ui_ExtraSettingsWidget::trayBox.

Referenced by Ui_ExtraSettingsWidget::setupUi().

```
331     {
332         ExtraSettingsWidget->setWindowTitle(QApplication::translate("ExtraSettingsWidget", "Application
       Settings", Q_NULLPTR));
333         fpsWarnLabel->setText(QApplication::translate("ExtraSettingsWidget", "Warning: high
       frame rates may cause stability issues", Q_NULLPTR));
334         label_3->setText(QApplication::translate("ExtraSettingsWidget", "Frame rate (FPS):",
      Q_NULLPTR));
335         label_6->setText(QApplication::translate("ExtraSettingsWidget", "Location:", Q_NULLPTR));
336 #ifndef QT_NO_TOOLTIP
337         sAccelBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "Try this if you're
       having problems with the scroll wheel.", Q_NULLPTR));
338 #endif // QT_NO_TOOLTIP
339         sAccelBox->setText(QApplication::translate("ExtraSettingsWidget", "Disable scroll
       acceleration", Q_NULLPTR));
340         animPathLabel->setText(QApplication::translate("ExtraSettingsWidget", "
      /path/to/animations", Q_NULLPTR));
341         animCountLabel->setText(QApplication::translate("ExtraSettingsWidget", "0 animations
       found", Q_NULLPTR));
342         animScanButton->setText(QApplication::translate("ExtraSettingsWidget", "Re-scan",
      Q_NULLPTR));
343 #ifndef QT_NO_TOOLTIP
344         brightnessBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "By default,
       the same brightness level will be applied to all profiles and all devices. Enable this to store it with the
       lighting mode instead.", Q_NULLPTR));
345 #endif // QT_NO_TOOLTIP
346         brightnessBox->setText(QApplication::translate("ExtraSettingsWidget", "Set brightness
       per-mode", Q_NULLPTR));
347 #ifndef QT_NO_TOOLTIP
348         trayBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "The tray icon will not
       be displayed. The application will still run in the background; re-launch the app to see the GUI again.",
      Q_NULLPTR));
349 #endif // QT_NO_TOOLTIP
350         trayBox->setText(QApplication::translate("ExtraSettingsWidget", "Hide tray icon", Q_NULLPTR)
     );
351         label->setText(QApplication::translate("ExtraSettingsWidget", "Behavior", Q_NULLPTR));
352         label_5->setText(QApplication::translate("ExtraSettingsWidget", "Animation scripts",
      Q_NULLPTR));
353         label_2->setText(QApplication::translate("ExtraSettingsWidget", "Hardware", Q_NULLPTR));
354 #ifndef QT_NO_TOOLTIP
355         ditherBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "May improve
       appearance on some keyboards.", Q_NULLPTR));
356 #endif // QT_NO_TOOLTIP
357         ditherBox->setText(QApplication::translate("ExtraSettingsWidget", "Use spatial dithering
       to simulate extra color resolution", Q_NULLPTR));
358 #ifndef QT_NO_TOOLTIP
359         delayBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "When using macros with
       strings longer than 25 chars, some OS may lose characters (e.g. Mint 17.2). Select to prevent that bug.",
      Q_NULLPTR));
360 #endif // QT_NO_TOOLTIP
361         delayBox->setText(QApplication::translate("ExtraSettingsWidget", "Use delay for very long
       macros", Q_NULLPTR));
362         osxLabel->setText(QApplication::translate("ExtraSettingsWidget", "OS X tweaks", Q_NULLPTR))
     ;
363 #ifndef QT_NO_TOOLTIP
364         mAccelBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "Try this if you're
       having problems with mouse movement.", Q_NULLPTR));
365 #endif // QT_NO_TOOLTIP
366         mAccelBox->setText(QApplication::translate("ExtraSettingsWidget", "Disable mouse
       acceleration", Q_NULLPTR));
367         label_10->setText(QApplication::translate("ExtraSettingsWidget", "Scroll", Q_NULLPTR));
368         label_11->setText(QApplication::translate("ExtraSettingsWidget", "line(s) at a time",
      Q_NULLPTR));
369     } // retranslateUi
```

Here is the caller graph for this function:



12.12.2.2 void Ui_ExtraSettingsWidget::setupUi (QDialog * *ExtraSettingsWidget*) [inline],[inherited]

Definition at line 78 of file ui_extrasettingswidget.h.

References Ui_ExtraSettingsWidget::animCountLabel, Ui_ExtraSettingsWidget::animPathLabel, Ui_ExtraSettingsWidget::animScanButton, Ui_ExtraSettingsWidget::brightnessBox, Ui_ExtraSettingsWidget::buttonBox, Ui_ExtraSettingsWidget::delayBox, Ui_ExtraSettingsWidget::ditherBox, Ui_ExtraSettingsWidget::fpsBox, Ui_ExtraSettingsWidget::fpsWarnLabel, Ui_ExtraSettingsWidget::gridLayout, Ui_ExtraSettingsWidget::horizontalLayout, Ui_ExtraSettingsWidget::horizontalLayout_2, Ui_ExtraSettingsWidget::horizontalSpacer, Ui_ExtraSettingsWidget::horizontalSpacer_2, Ui_ExtraSettingsWidget::horizontalSpacer_3, Ui_ExtraSettingsWidget::label, Ui_ExtraSettingsWidget::label_10, Ui_ExtraSettingsWidget::label_11, Ui_ExtraSettingsWidget::label_2, Ui_ExtraSettingsWidget::label_3, Ui_ExtraSettingsWidget::label_5, Ui_ExtraSettingsWidget::label_6, Ui_ExtraSettingsWidget::line, Ui_ExtraSettingsWidget::line_2, Ui_ExtraSettingsWidget::line_3, Ui_ExtraSettingsWidget::mAccelBox, Ui_ExtraSettingsWidget::osxLabel, Ui_ExtraSettingsWidget::osxLine, Ui_ExtraSettingsWidget::retranslateUi(), Ui_ExtraSettingsWidget::sAccelBox, Ui_ExtraSettingsWidget::sSpeedBox, Ui_ExtraSettingsWidget::sSpeedWidget, Ui_ExtraSettingsWidget::trayBox, Ui_ExtraSettingsWidget::verticalSpacer, Ui_ExtraSettingsWidget::verticalSpacer_10, Ui_ExtraSettingsWidget::verticalSpacer_11, Ui_ExtraSettingsWidget::verticalSpacer_2, Ui_ExtraSettingsWidget::verticalSpacer_3, Ui_ExtraSettingsWidget::verticalSpacer_4, Ui_ExtraSettingsWidget::verticalSpacer_5, Ui_ExtraSettingsWidget::verticalSpacer_6, Ui_ExtraSettingsWidget::verticalSpacer_7, Ui_ExtraSettingsWidget::verticalSpacer_8, and Ui_ExtraSettingsWidget::verticalSpacer_9.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget().

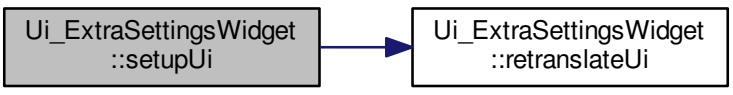
```
79     {
80         if (ExtraSettingsWidget->objectName().isEmpty())
81             ExtraSettingsWidget->setObjectName(QStringLiteral("ExtraSettingsWidget"));
82         ExtraSettingsWidget->resize(650, 515);
83         gridLayout = new QGridLayout(ExtraSettingsWidget);
84         gridLayout->setObjectName(QStringLiteral("gridLayout"));
85         horizontalSpacer_2 = new QSpacerItem(30, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
86
87         gridLayout->addItem(horizontalSpacer_2, 12, 2, 1, 1);
88
89         verticalSpacer = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed);
90
91         gridLayout->addItem(verticalSpacer, 2, 0, 1, 1);
92
93         verticalSpacer_3 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
94
95         gridLayout->addItem(verticalSpacer_3, 3, 0, 1, 1);
96
97         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
98
99         gridLayout->addItem(verticalSpacer_2, 19, 5, 1, 1);
100
101         fpsWarnLabel = new QLabel(ExtraSettingsWidget);
102         fpsWarnLabel->setObjectName(QStringLiteral("fpsWarnLabel"));
103         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
104         sizePolicy.setHorizontalStretch(0);
105         sizePolicy.setVerticalStretch(0);
106         sizePolicy.setHeightForWidth(fpsWarnLabel->sizePolicy().hasHeightForWidth());
107         fpsWarnLabel->setSizePolicy(sizePolicy);
108
109         gridLayout->addWidget(fpsWarnLabel, 12, 5, 1, 1);
110
111         horizontalSpacer_3 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
```

```
    QSizePolicy::Minimum);
112
113         gridLayout->addItem(horizontalSpacer_3, 12, 4, 1, 1);
114
115         label_3 = new QLabel(ExtraSettingsWidget);
116         label_3->setObjectName(QStringLiteral("label_3"));
117
118         gridLayout->addWidget(label_3, 12, 1, 1, 1);
119
120         fpsBox = new QSpinBox(ExtraSettingsWidget);
121         fpsBox->setObjectName(QStringLiteral("fpsBox"));
122         fpsBox->setMinimum(5);
123         fpsBox->setMaximum(60);
124         fpsBox->setValue(30);
125
126         gridLayout->addWidget(fpsBox, 12, 3, 1, 1);
127
128         verticalSpacer_4 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
129
130         gridLayout->addItem(verticalSpacer_4, 7, 0, 1, 1);
131
132         verticalSpacer_9 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
133
134         gridLayout->addItem(verticalSpacer_9, 12, 0, 1, 1);
135
136         label_6 = new QLabel(ExtraSettingsWidget);
137         label_6->setObjectName(QStringLiteral("label_6"));
138
139         gridLayout->addWidget(label_6, 7, 1, 1, 1);
140
141         verticalSpacer_6 = new QSpacerItem(20, 10, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
142
143         gridLayout->addItem(verticalSpacer_6, 8, 1, 1, 1);
144
145         verticalSpacer_5 = new QSpacerItem(20, 10, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
146
147         gridLayout->addItem(verticalSpacer_5, 4, 1, 1, 1);
148
149         sAccelBox = new QCheckBox(ExtraSettingsWidget);
150         sAccelBox->setObjectName(QStringLiteral("sAccelBox"));
151
152         gridLayout->addWidget(sAccelBox, 18, 1, 1, 4);
153
154         verticalSpacer_7 = new QSpacerItem(20, 10, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
155
156         gridLayout->addItem(verticalSpacer_7, 14, 1, 1, 1);
157
158         verticalSpacer_8 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
159
160         gridLayout->addItem(verticalSpacer_8, 13, 0, 1, 1);
161
162         verticalSpacer_10 = new QSpacerItem(0, 28, QSizePolicy::Minimum,
    QSizePolicy::Fixed);
163
164         gridLayout->addItem(verticalSpacer_10, 17, 0, 1, 1);
165
166         verticalSpacer_11 = new QSpacerItem(0, 28, QSizePolicy::Minimum,
    QSizePolicy::Fixed);
167
168         gridLayout->addItem(verticalSpacer_11, 18, 0, 1, 1);
169
170         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
171
172         gridLayout->addItem(horizontalSpacer, 12, 6, 1, 1);
173
174         line_2 = new QFrame(ExtraSettingsWidget);
175         line_2->setObjectName(QStringLiteral("line_2"));
176         line_2->setFrameShape(QFrame::HLine);
177         line_2->setFrameShadow(QFrame::Sunken);
178
179         gridLayout->addWidget(line_2, 10, 0, 1, 7);
180
181         horizontalLayout = new QHBoxLayout();
182         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
183         animPathLabel = new QLabel(ExtraSettingsWidget);
184         animPathLabel->setObjectName(QStringLiteral("animPathLabel"));
185
186         horizontalLayout->addWidget(animPathLabel);
187
188         animCountLabel = new QLabel(ExtraSettingsWidget);
```

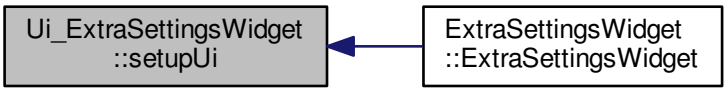
```
189         animCountLabel->setObjectName(QStringLiteral("animCountLabel"));
190         sizePolicy.setHeightForWidth(animCountLabel->sizePolicy().hasHeightForWidth());
191         animCountLabel->setSizePolicy(sizePolicy);
192         animCountLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
193 
194         horizontalLayout->addWidget(animCountLabel);
195 
196         animScanButton = new QPushButton(ExtraSettingsWidget);
197         animScanButton->setObjectName(QStringLiteral("animScanButton"));
198 
199         horizontalLayout->addWidget(animScanButton);
200 
201 
202         gridLayout->addLayout(horizontalLayout, 7, 3, 1, 4);
203 
204         brightnessBox = new QCheckBox(ExtraSettingsWidget);
205         brightnessBox->setObjectName(QStringLiteral("brightnessBox"));
206 
207         gridLayout->addWidget(brightnessBox, 3, 1, 1, 6);
208 
209         trayBox = new QCheckBox(ExtraSettingsWidget);
210         trayBox->setObjectName(QStringLiteral("trayBox"));
211 
212         gridLayout->addWidget(trayBox, 2, 1, 1, 6);
213 
214         label = new QLabel(ExtraSettingsWidget);
215         label->setObjectName(QStringLiteral("label"));
216         QFont font;
217         font.setBold(true);
218         font.setWeight(75);
219         label->setFont(font);
220 
221         gridLayout->addWidget(label, 0, 0, 1, 7);
222 
223         line = new QFrame(ExtraSettingsWidget);
224         line->setObjectName(QStringLiteral("line"));
225         line->setFrameShape(QFrame::HLine);
226         line->setFrameShadow(QFrame::Sunken);
227 
228         gridLayout->addWidget(line, 1, 0, 1, 7);
229 
230         label_5 = new QLabel(ExtraSettingsWidget);
231         label_5->setObjectName(QStringLiteral("label_5"));
232         label_5->setFont(font);
233 
234         gridLayout->addWidget(label_5, 5, 0, 1, 7);
235 
236         line_3 = new QFrame(ExtraSettingsWidget);
237         line_3->setObjectName(QStringLiteral("line_3"));
238         line_3->setFrameShape(QFrame::HLine);
239         line_3->setFrameShadow(QFrame::Sunken);
240 
241         gridLayout->addWidget(line_3, 6, 0, 1, 7);
242 
243         label_2 = new QLabel(ExtraSettingsWidget);
244         label_2->setObjectName(QStringLiteral("label_2"));
245         label_2->setFont(font);
246 
247         gridLayout->addWidget(label_2, 9, 0, 1, 7);
248 
249         ditherBox = new QCheckBox(ExtraSettingsWidget);
250         ditherBox->setObjectName(QStringLiteral("ditherBox"));
251 
252         gridLayout->addWidget(ditherBox, 13, 1, 1, 6);
253 
254         delayBox = new QCheckBox(ExtraSettingsWidget);
255         delayBox->setObjectName(QStringLiteral("delayBox"));
256 
257         gridLayout->addWidget(delayBox, 14, 1, 1, 6);
258 
259         osxLabel = new QLabel(ExtraSettingsWidget);
260         osxLabel->setObjectName(QStringLiteral("osxLabel"));
261         osxLabel->setFont(font);
262 
263         gridLayout->addWidget(osxLabel, 15, 0, 1, 7);
264 
265         osxLine = new QFrame(ExtraSettingsWidget);
266         osxLine->setObjectName(QStringLiteral("osxLine"));
267         osxLine->setFrameShape(QFrame::HLine);
268         osxLine->setFrameShadow(QFrame::Sunken);
269 
270         gridLayout->addWidget(osxLine, 16, 0, 1, 7);
271 
272         mAccelBox = new QCheckBox(ExtraSettingsWidget);
273         mAccelBox->setObjectName(QStringLiteral("mAccelBox"));
274 
275         gridLayout->addWidget(mAccelBox, 17, 1, 1, 6);
```

```
276
277         sSpeedWidget = new QWidget(ExtraSettingsWidget);
278         sSpeedWidget->setObjectName(QStringLiteral("sSpeedWidget"));
279         sizePolicy.setHeightForWidth(sSpeedWidget->sizePolicy().hasHeightForWidth());
280         sSpeedWidget->setSizePolicy(sizePolicy);
281         horizontalLayout_2 = new QHBoxLayout(sSpeedWidget);
282         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
283         horizontalLayout_2->setContentsMargins(0, 0, 0, 0);
284         label_10 = new QLabel(sSpeedWidget);
285         label_10->setObjectName(QStringLiteral("label_10"));
286
287         horizontalLayout_2->addWidget(label_10);
288
289         sSpeedBox = new QSpinBox(sSpeedWidget);
290         sSpeedBox->setObjectName(QStringLiteral("sSpeedBox"));
291         sSpeedBox->setMinimum(1);
292         sSpeedBox->setMaximum(10);
293         sSpeedBox->setValue(3);
294
295         horizontalLayout_2->addWidget(sSpeedBox);
296
297         label_11 = new QLabel(sSpeedWidget);
298         label_11->setObjectName(QStringLiteral("label_11"));
299         sizePolicy.setHeightForWidth(label_11->sizePolicy().hasHeightForWidth());
300         label_11->setSizePolicy(sizePolicy);
301
302         horizontalLayout_2->addWidget(label_11);
303
304
305         gridLayout->addWidget(sSpeedWidget, 18, 5, 1, 2);
306
307         buttonBox = new QDialogButtonBox(ExtraSettingsWidget);
308         buttonBox->setObjectName(QStringLiteral("buttonBox"));
309         buttonBox->setOrientation(Qt::Horizontal);
310         buttonBox->setStandardButtons(QDialogButtonBox::Ok);
311
312         gridLayout->addWidget(buttonBox, 20, 0, 1, 7);
313
314         QWidget::setTabOrder(trayBox, brightnessBox);
315         QWidget::setTabOrder(brightnessBox, animScanButton);
316         QWidget::setTabOrder(animScanButton, fpsBox);
317         QWidget::setTabOrder(fpsBox, ditherBox);
318         QWidget::setTabOrder(ditherBox, delayBox);
319         QWidget::setTabOrder(delayBox, mAccelBox);
320         QWidget::setTabOrder(mAccelBox, sAccelBox);
321         QWidget::setTabOrder(sAccelBox, sSpeedBox);
322
323         retranslateUi(ExtraSettingsWidget);
324         QObject::connect(buttonBox, SIGNAL(accepted()), ExtraSettingsWidget, SLOT(accept()));
325         QObject::connect(buttonBox, SIGNAL(rejected()), ExtraSettingsWidget, SLOT(reject()));
326
327         QMetaObject::connectSlotsByName(ExtraSettingsWidget);
328     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.12.3 Field Documentation

12.12.3.1 QLabel* Ui_ExtraSettingsWidget::animCountLabel [inherited]

Definition at line 57 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::on_animScanButton_clicked(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.2 QLabel* Ui_ExtraSettingsWidget::animPathLabel [inherited]

Definition at line 56 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.3 QPushButton* Ui_ExtraSettingsWidget::animScanButton [inherited]

Definition at line 58 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.4 QCheckBox* Ui_ExtraSettingsWidget::brightnessBox [inherited]

Definition at line 59 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.5 QDialogButtonBox* Ui_ExtraSettingsWidget::buttonBox [inherited]

Definition at line 76 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.6 QCheckBox* Ui_ExtraSettingsWidget::delayBox [inherited]

Definition at line 67 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.7 QCheckBox* Ui_ExtraSettingsWidget::ditherBox [inherited]

Definition at line 66 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.8 QSpinBox* Ui_ExtraSettingsWidget::fpsBox [inherited]

Definition at line 42 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.9 QLabel* Ui_ExtraSettingsWidget::fpsWarnLabel [inherited]

Definition at line 39 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_fpsBox_valueChanged(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.10 QGridLayout* Ui_ExtraSettingsWidget::gridLayout [inherited]

Definition at line 34 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.11 QHBoxLayout* Ui_ExtraSettingsWidget::horizontalLayout [inherited]

Definition at line 55 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.12 QHBoxLayout* Ui_ExtraSettingsWidget::horizontalLayout_2 [inherited]

Definition at line 72 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.13 QSpacerItem* Ui_ExtraSettingsWidget::horizontalSpacer [inherited]

Definition at line 53 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.14 QSpacerItem* Ui_ExtraSettingsWidget::horizontalSpacer_2 [inherited]

Definition at line 35 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.15 QSpacerItem* Ui_ExtraSettingsWidget::horizontalSpacer_3 [inherited]

Definition at line 40 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.16 QLabel* Ui_ExtraSettingsWidget::label [inherited]

Definition at line 61 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.17 QLabel* Ui_ExtraSettingsWidget::label_10 [inherited]

Definition at line 73 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.18 QLabel* Ui_ExtraSettingsWidget::label_11 [inherited]

Definition at line 75 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.19 QLabel* Ui_ExtraSettingsWidget::label_2 [inherited]

Definition at line 65 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.20 QLabel* Ui_ExtraSettingsWidget::label_3 [inherited]

Definition at line 41 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.21 QLabel* Ui_ExtraSettingsWidget::label_5 [inherited]

Definition at line 63 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.22 QLabel* Ui_ExtraSettingsWidget::label_6 [inherited]

Definition at line 45 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.23 QFrame* Ui_ExtraSettingsWidget::line [inherited]

Definition at line 62 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.24 QFrame* Ui_ExtraSettingsWidget::line_2 [inherited]

Definition at line 54 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.25 QFrame* Ui_ExtraSettingsWidget::line_3 [inherited]

Definition at line 64 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.26 QCheckBox* Ui_ExtraSettingsWidget::mAccelBox [inherited]

Definition at line 70 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.27 QLabel* Ui_ExtraSettingsWidget::osxLabel [inherited]

Definition at line 68 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.28 QFrame* Ui_ExtraSettingsWidget::osxLine [inherited]

Definition at line 69 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.29 QCheckBox* Ui_ExtraSettingsWidget::sAccelBox [inherited]

Definition at line 48 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sSpeedBox_valueChanged(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.30 QSpinBox* Ui_ExtraSettingsWidget::sSpeedBox [inherited]

Definition at line 74 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sAccelBox_clicked(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.31 QWidget* Ui_ExtraSettingsWidget::sSpeedWidget [inherited]

Definition at line 71 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sAccelBox_clicked(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.32 QCheckBox* Ui_ExtraSettingsWidget::trayBox [inherited]

Definition at line 60 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), Ui_ExtraSettingsWidget::retranslateUi(), and Ui_ExtraSettingsWidget::setupUi().

12.12.3.33 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer [inherited]

Definition at line 36 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.34 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_10 [inherited]

Definition at line 51 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.35 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_11 [inherited]

Definition at line 52 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.36 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_2 [inherited]

Definition at line 38 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.37 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_3 [inherited]

Definition at line 37 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.38 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_4 [inherited]

Definition at line 43 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.39 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_5 [inherited]

Definition at line 47 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.40 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_6 [inherited]

Definition at line 46 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.41 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_7 [inherited]

Definition at line 49 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.42 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_8 [inherited]

Definition at line 50 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

12.12.3.43 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_9 [inherited]

Definition at line 44 of file ui_extrasettingswidget.h.

Referenced by Ui_ExtraSettingsWidget::setupUi().

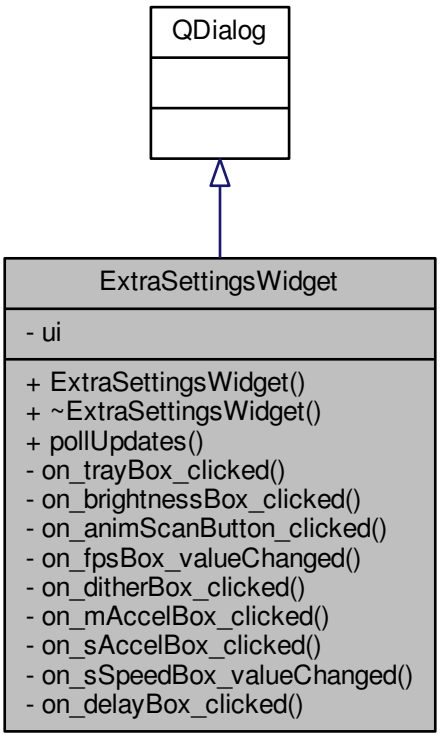
The documentation for this class was generated from the following file:

- src/ckb/ui_extrasettingswidget.h

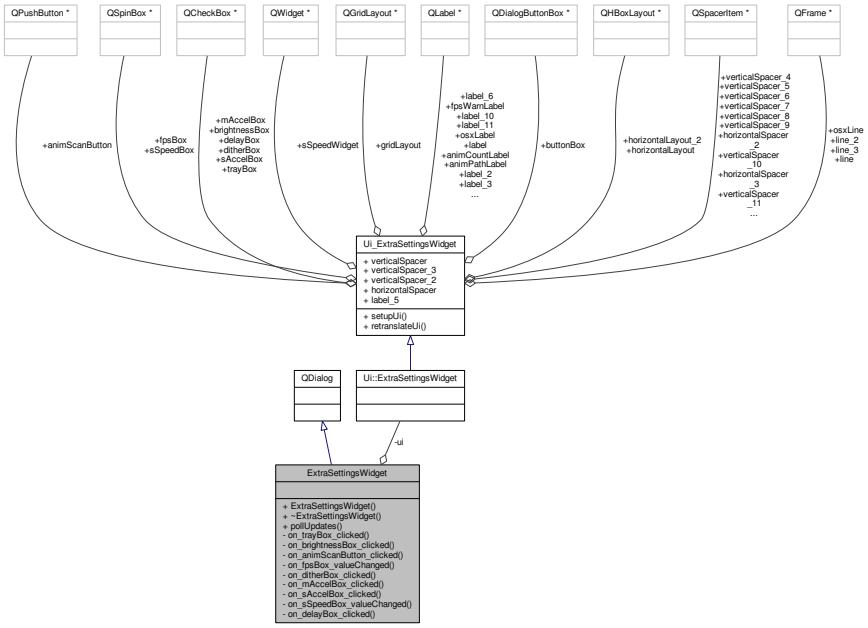
12.13 ExtraSettingsWidget Class Reference

`#include <src/ckb/extrasettingswidget.h>`

Inheritance diagram for ExtraSettingsWidget:



Collaboration diagram for ExtraSettingsWidget:



Public Member Functions

- ExtraSettingsWidget (QWidget *parent=0)
- ~ExtraSettingsWidget ()
- void pollUpdates ()

Private Slots

- void on_trayBox_clicked (bool checked)
- void on_brightnessBox_clicked (bool checked)
- void on_animScanButton_clicked ()
- void on_fpsBox_valueChanged (int arg1)
- void on_ditherBox_clicked (bool checked)
- void on_mAccelBox_clicked (bool checked)
- void on_sAccelBox_clicked (bool checked)
- void on_sSpeedBox_valueChanged (int arg1)
- void on_delayBox_clicked (bool checked)

Private Attributes

- Ui::ExtraSettingsWidget * ui

12.13.1 Detailed Description

Definition at line 10 of file extrasettingswidget.h.

12.13.2 Constructor & Destructor Documentation

12.13.2.1 ExtraSettingsWidget::ExtraSettingsWidget (QWidget * *parent* = 0) [explicit]

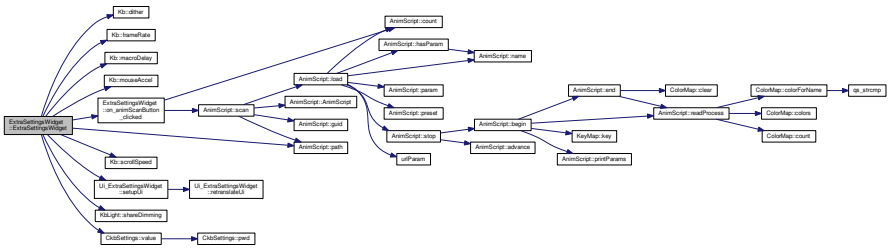
Read Macro Delay setting, update UI and Kb-internal flag

Definition at line 10 of file extrasettingswidget.cpp.

References Ui_ExtraSettingsWidget::animPathLabel, Ui_ExtraSettingsWidget::brightnessBox, Ui_ExtraSettingsWidget::delayBox, Kb::dither(), Ui_ExtraSettingsWidget::ditherBox, Ui_ExtraSettingsWidget::fpsBox, Ui_ExtraSettingsWidget::fpsWarnLabel, Kb::frameRate(), lastSharedDimming, Ui_ExtraSettingsWidget::mAccelBox, Kb::macroDelay(), KbLight::MAX_DIM, Kb::mouseAccel(), on_animScanButton_clicked(), Ui_ExtraSettingsWidget::osxLabel, Ui_ExtraSettingsWidget::osxLine, AnimScript::path(), Ui_ExtraSettingsWidget::sAccelBox, Kb::scrollSpeed(), Ui_ExtraSettingsWidget::setupUi(), KbLight::shareDimming(), Ui_ExtraSettingsWidget::sSpeedBox, Ui_ExtraSettingsWidget::sSpeedWidget, Ui_ExtraSettingsWidget::trayBox, ui, and CkbSettings::value().

```
10                                                      :
11     QDialog(parent),
12     ui(new Ui::ExtraSettingsWidget)
13 {
14     ui->setupUi(this);
15     CkbSettings settings("Program");
16
17     // Read frame rate from settings
18     int rate = settings.value("framerate").toInt();
19     if(rate <= 0 || rate > 60)
20         rate = 30;
21     ui->fpsBox->setValue(rate);
22     Kb::frameRate(rate);
23     ui->fpsWarnLabel->setVisible(rate > 30);
24
25     // Read global brightness setting (default = on, 100% brightness)
26     int dimming = settings.value("GlobalBrightness").toInt();
27     if(dimming < -1 || dimming > KbLight::MAX_DIM)
28         dimming = 0;
29     lastSharedDimming = dimming;
30     KbLight::shareDimming(dimming);
31     // Set checkbox value (-1 = don't share)
32     ui->brightnessBox->setChecked(dimming == -1);
33
34     // Read dither
35     bool dither = settings.value("Dither").toBool();
36     Kb::dither(dither);
37     ui->ditherBox->setChecked(dither);
38
39 #ifdef Q_OS_MACX
40     // Read OSX settings
41     bool noAccel = settings.value("DisableMouseAccel").toBool();
42     Kb::mouseAccel(!noAccel);
43     ui->mAccelBox->setChecked(noAccel);
44     bool noScrollAccel = settings.value("DisableScrollAccel").toBool();
45     int scrollSpeed = settings.value("ScrollSpeed", 3).toInt();
46     ui->sAccelBox->setChecked(noScrollAccel);
47     ui->sSpeedBox->setValue(scrollSpeed);
48     Kb::scrollSpeed(noScrollAccel ? scrollSpeed : 0);
49     if(!noScrollAccel)
50         ui->sSpeedWidget->hide();
51 #else
52     // Hide all OSX options on Linux
53     ui->osxLabel->hide();
54     ui->osxLine->hide();
55     ui->mAccelBox->hide();
56     ui->sAccelBox->hide();
57     ui->sSpeedWidget->hide();
58 #endif
59
60     // Read tray icon setting
61     ui->trayBox->setChecked(settings.value("SuppressTrayIcon").toBool());
62
63     // Update animation info
64     ui->animPathLabel->setText(AnimScript::path());
65     on_animScanButton_clicked();
66
67     bool macroDelay = settings.value("MacroDelay").toBool();
68     Kb::macroDelay(macroDelay);
69     ui->delayBox->setChecked(macroDelay);
70 }
```

Here is the call graph for this function:



12.13.2.2 ExtraSettingsWidget::~ExtraSettingsWidget ( )

Definition at line 73 of file extrasettingswidget.cpp.

References ui.

```
73                                          {
74     delete ui;
75 }
```

12.13.3 Member Function Documentation

12.13.3.1 void ExtraSettingsWidget::on_animScanButton_clicked ( ) [private], [slot]

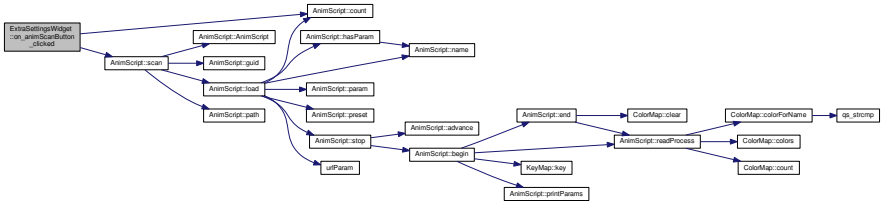
Definition at line 87 of file extrasettingswidget.cpp.

References Ui_ExtraSettingsWidget::animCountLabel, AnimScript::count(), AnimScript::scan(), and ui.

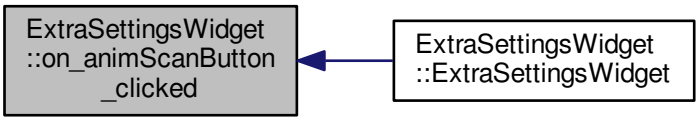
Referenced by ExtraSettingsWidget().

```
87                                                  {
88     AnimScript::scan();
89     int count = AnimScript::count();
90     if(count == 0)
91         ui->animCountLabel->setText("No animations found");
92     else if(count == 1)
93         ui->animCountLabel->setText("1 animation found");
94     else
95         ui->animCountLabel->setText(QString("%1 animations found").arg(count));
96 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



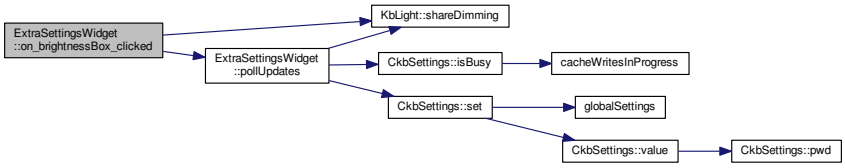
12.13.3.2 void ExtraSettingsWidget::on_brightnessBox_clicked (bool *checked*) [private], [slot]

Definition at line 82 of file extrasettingswidget.cpp.

References pollUpdates(), and KbLight::shareDimming().

```
82                                                          {
83     KbLight::shareDimming(checked ? -1 : 0);
84     pollUpdates();
85 }
```

Here is the call graph for this function:



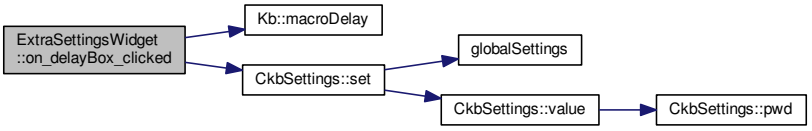
12.13.3.3 void ExtraSettingsWidget::on_delayBox_clicked (bool *checked*) [private], [slot]

Definition at line 143 of file extrasettingswidget.cpp.

References Kb::macroDelay(), and CkbSettings::set().

```
143                                                         {
144     CkbSettings::set("Program/MacroDelay", checked);
145     Kb::macroDelay(checked);
146 }
```

Here is the call graph for this function:



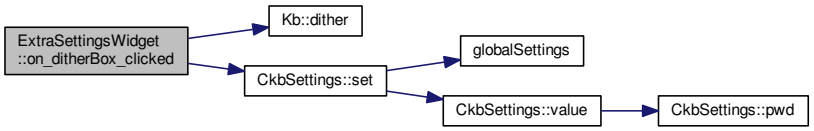
12.13.3.4 void ExtraSettingsWidget::on_ditherBox_clicked (bool *checked*) [private], [slot]

Definition at line 110 of file extrasettingswidget.cpp.

References Kb::dither(), and CkbSettings::set().

```
110                                                     {
111     CkbSettings::set("Program/Dither", checked);
112     Kb::dither(checked);
113 }
```

Here is the call graph for this function:



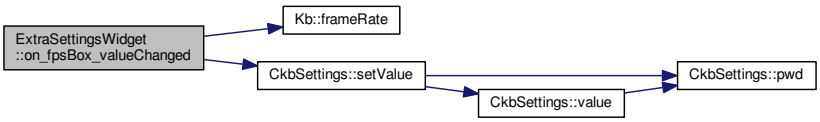
12.13.3.5 void ExtraSettingsWidget::on_fpsBox_valueChanged (int *arg1*) [private], [slot]

Definition at line 98 of file extrasettingswidget.cpp.

References Ui_ExtraSettingsWidget::fpsWarnLabel, Kb::frameRate(), CkbSettings::setValue(), and ui.

```
98                                                      {
99     // Set FPS
100     CkbSettings settings("Program");
101     settings.setValue("framerate", arg1);
102     Kb::frameRate(arg1);
103     // Show warning label if FPS is above 30
104     if(arg1 > 30)
105         ui->fpsWarnLabel->show();
106     else
107         ui->fpsWarnLabel->hide();
108 }
```

Here is the call graph for this function:



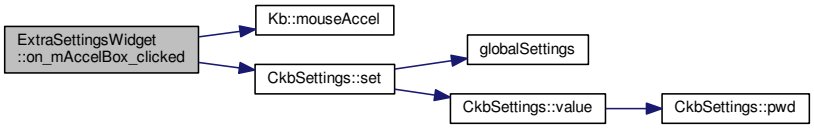
12.13.3.6 void ExtraSettingsWidget::on_mAccelBox_clicked (bool *checked*) [private], [slot]

Definition at line 124 of file extrasettingswidget.cpp.

References Kb::mouseAccel(), and CkbSettings::set().

```
124                                                     {
125     CkbSettings::set("Program/DisableMouseAccel", checked);
126     Kb::mouseAccel(!checked);
127 }
```

Here is the call graph for this function:



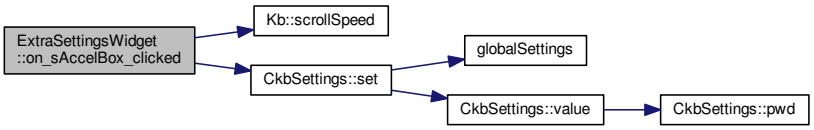
12.13.3.7 void ExtraSettingsWidget::on_sAccelBox_clicked (bool *checked*) [private], [slot]

Definition at line 129 of file extrasettingswidget.cpp.

References Kb::scrollSpeed(), CkbSettings::set(), Ui_ExtraSettingsWidget::sSpeedBox, Ui_ExtraSettingsWidget::sSpeedWidget, and ui.

```
129                                                     {
130     CkbSettings::set("Program/DisableScrollAccel", checked);
131     Kb::scrollSpeed(checked ? ui->sSpeedBox->value() : 0);
132     if(checked)
133         ui->sSpeedWidget->show();
134     else
135         ui->sSpeedWidget->hide();
136 }
```

Here is the call graph for this function:



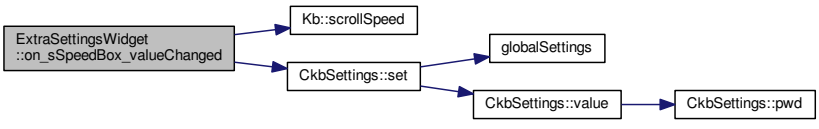
12.13.3.8 void ExtraSettingsWidget::on_sSpeedBox_valueChanged (int *arg1*) [private], [slot]

Definition at line 138 of file extrasettingswidget.cpp.

References Ui_ExtraSettingsWidget::sAccelBox, Kb::scrollSpeed(), CkbSettings::set(), and ui.

```
138                                                     {
139     CkbSettings::set("Program/ScrollSpeed", arg1);
140     Kb::scrollSpeed(ui->sAccelBox->isChecked() ? arg1 : 0);
141 }
```

Here is the call graph for this function:



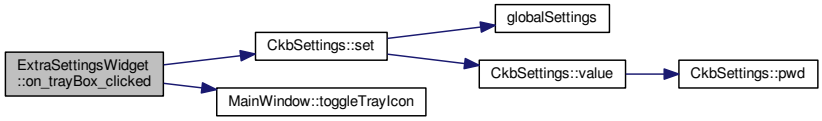
12.13.3.9 void ExtraSettingsWidget::on_trayBox_clicked (bool *checked*) [private], [slot]

Definition at line 77 of file extrasettingswidget.cpp.

References MainWindow::mainWindow, CkbSettings::set(), and MainWindow::toggleTrayIcon().

```
77                                                  {
78     CkbSettings::set("Program/SuppressTrayIcon", checked);
79     MainWindow::mainWindow->toggleTrayIcon(!checked);
80 }
```

Here is the call graph for this function:



12.13.3.10 void ExtraSettingsWidget::pollUpdates ( )

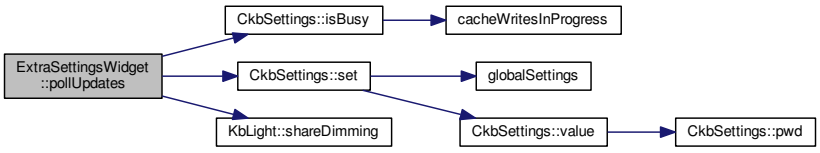
Definition at line 115 of file extrasettingswidget.cpp.

References CkbSettings::isBusy(), lastSharedDimming, CkbSettings::set(), and KbLight::shareDimming().

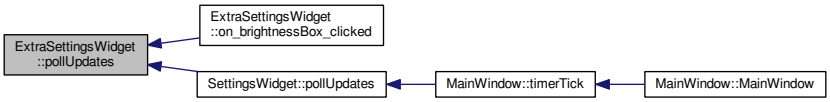
Referenced by on_brightnessBox_clicked(), and SettingsWidget::pollUpdates().

```
115                                     {
116     // Check for changes to shared brightness setting
117     int dimming = KbLight::shareDimming();
118     if(dimming != lastSharedDimming && !CkbSettings::isBusy()){
119         CkbSettings::set("Program/GlobalBrightness", dimming);
120         lastSharedDimming = dimming;
121     }
122 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.13.4 Field Documentation

12.13.4.1 Ui::ExtraSettingsWidget* ExtraSettingsWidget::ui [private]

Definition at line 33 of file extrasettingswidget.h.

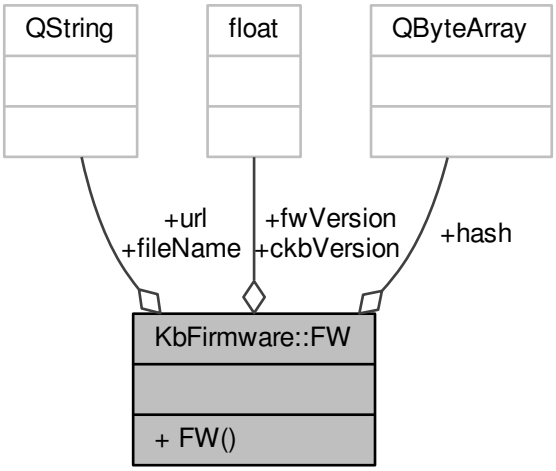
Referenced by ExtraSettingsWidget(), on_animScanButton_clicked(), on_fpsBox_valueChanged(), on_sAccelBox-_clicked(), on_sSpeedBox_valueChanged(), and ~ExtraSettingsWidget().

The documentation for this class was generated from the following files:

- src/ckb/extrasettingswidget.h
- src/ckb/extrasettingswidget.cpp

12.14 KbFirmware::FW Struct Reference

Collaboration diagram for KbFirmware::FW:



Public Member Functions

- FW ()

Data Fields

- QString url
- QString fileName
- QByteArray hash
- float fwVersion
- float ckbVersion

12.14.1 Detailed Description

Definition at line 37 of file kbfirmware.h.

12.14.2 Constructor & Destructor Documentation

12.14.2.1 KbFirmware::FW::FW ( )

Definition at line 12 of file kbfirmware.cpp.

```
12 : fwVersion(0.f), ckbVersion(0.f) {}
```

12.14.3 Field Documentation

12.14.3.1 float KbFirmware::FW::ckbVersion

Definition at line 40 of file kbfirmware.h.

Referenced by KbFirmware::_latestForBoard(), and KbFirmware::processDownload().

12.14.3.2 QString KbFirmware::FW::fileName

Definition at line 38 of file kbfirmware.h.

Referenced by KbFirmware::_fileForBoard(), and KbFirmware::processDownload().

12.14.3.3 float KbFirmware::FW::fwVersion

Definition at line 40 of file kbfirmware.h.

Referenced by KbFirmware::_latestForBoard(), and KbFirmware::processDownload().

12.14.3.4 QByteArray KbFirmware::FW::hash

Definition at line 39 of file kbfirmware.h.

Referenced by KbFirmware::_fileForBoard(), KbFirmware::_latestForBoard(), and KbFirmware::processDownload().

12.14.3.5 QString KbFirmware::FW::url

Definition at line 38 of file kbfirmware.h.

Referenced by KbFirmware::_fileForBoard(), and KbFirmware::processDownload().

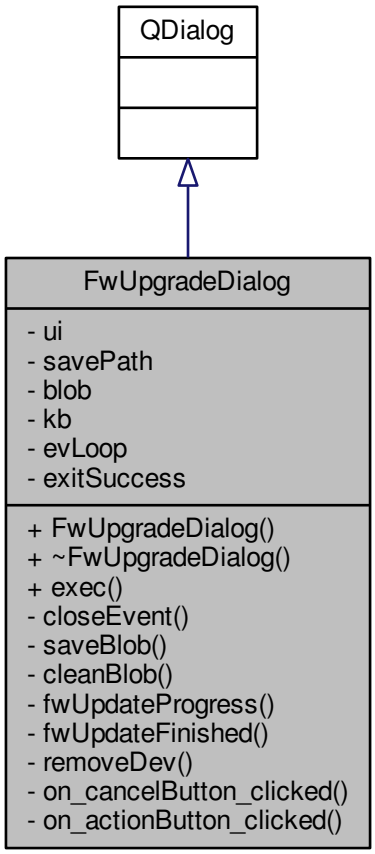
The documentation for this struct was generated from the following files:

- src/ckb/kbfirmware.h
- src/ckb/kbfirmware.cpp

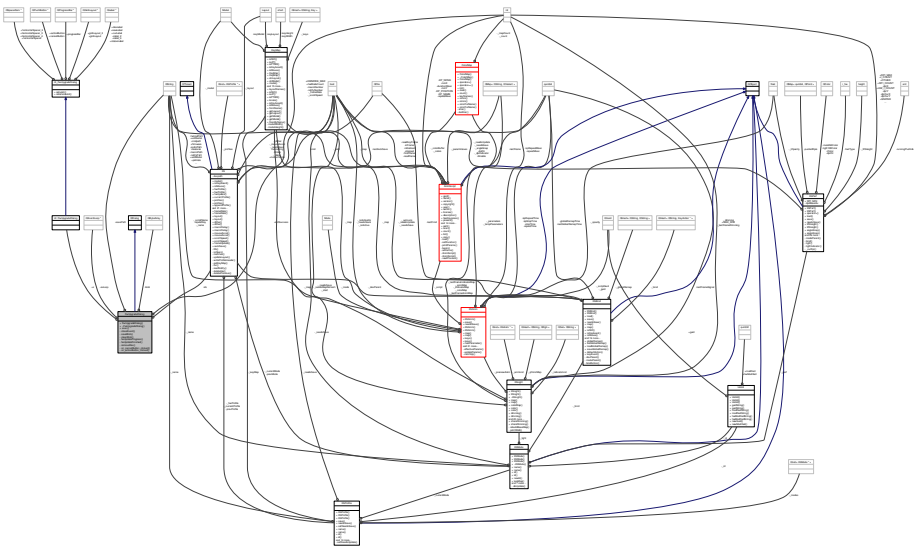
12.15 FwUpgradeDialog Class Reference

`#include <src/ckb/fwupgradedialog.h>`

Inheritance diagram for FwUpgradeDialog:



Collaboration diagram for FwUpgradeDialog:



Public Member Functions

- FwUpgradeDialog (QWidget *parent, float newV, const QByteArray &fwBlob, Kb *device)
- ~FwUpgradeDialog ()
- int exec ()

Private Slots

- void fwUpdateProgress (int current, int total)
- void fwUpdateFinished (bool succeeded)
- void removeDev ()
- void on_cancelButton_clicked ()
- void on_actionButton_clicked ()

Private Member Functions

- void closeEvent (QCloseEvent *event)
- const QString & saveBlob ()
- void cleanBlob ()

Private Attributes

- Ui::FwUpgradeDialog * ui
- QString savePath
- QByteArray blob
- Kb * kb
- QEventLoop * evLoop
- bool exitSuccess

12.15.1 Detailed Description

Definition at line 13 of file fwupgradedialog.h.

12.15.2 Constructor & Destructor Documentation

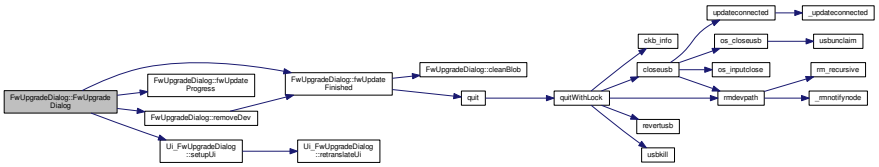
12.15.2.1 FwUpgradeDialog::FwUpgradeDialog (QWidget * *parent,* float *newV,* const QByteArray & *fwBlob,* Kb * *device*)

Definition at line 32 of file fwupgradedialog.cpp.

References Ui_FwUpgradeDialog::curLabel, Ui_FwUpgradeDialog::devLabel, DIALOG_HEIGHT_MIN, DIALOG_WIDTH, Kb::firmware, fwUpdateFinished(), fwUpdateProgress(), kb, Ui_FwUpgradeDialog::newLabel, removeDev(), Ui_FwUpgradeDialog::setupUi(), ui, and Kb::usbModel.

```
32                                                                                                    :
33     QDialog(parent),
34     ui(new Ui::FwUpgradeDialog),
35     blob(fwBlob), kb(device), evLoop(0), exitSuccess(true)
36 {
37     ui->setupUi(this);
38     ui->curLabel->setText(kb->firmware);
39     ui->newLabel->setText(QString::number(newV, 'f', 2));
40     ui->devLabel->setText(kb->usbModel);
41
42     connect(device, SIGNAL(destroyed()), this, SLOT(removeDev()));
43     connect(device, SIGNAL(fwUpdateProgress(int,int)), this, SLOT(
       fwUpdateProgress(int,int)));
44     connect(device, SIGNAL(fwUpdateFinished(bool)), this, SLOT(
       fwUpdateFinished(bool)));
45
46     setFixedSize(DIALOG_WIDTH, DIALOG_HEIGHT_MIN);
47 }
```

Here is the call graph for this function:



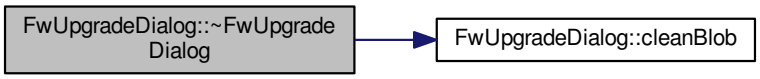
12.15.2.2 FwUpgradeDialog::~FwUpgradeDialog ( )

Definition at line 49 of file fwupgradedialog.cpp.

References cleanBlob(), and ui.

```
49                                      {
50     cleanBlob();
51     delete ui;
52 }
```

Here is the call graph for this function:



12.15.3 Member Function Documentation

12.15.3.1 void FwUpgradeDialog::cleanBlob ( ) [private]

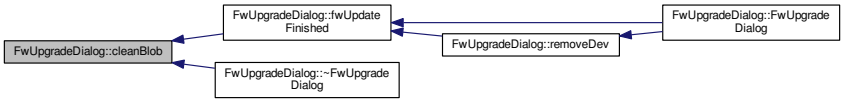
Definition at line 78 of file fwupgradedialog.cpp.

References savePath.

Referenced by fwUpdateFinished(), and ~FwUpgradeDialog().

```
78                                      {
79     if(savePath.isEmpty())
80         return;
81     QFile(savePath).remove();
82     savePath = "";
83 }
```

Here is the caller graph for this function:



12.15.3.2 void FwUpgradeDialog::closeEvent (QCloseEvent * *event*) [private]

Definition at line 54 of file fwupgradedialog.cpp.

```
54                                                {
55     event->ignore();
56 }
```

12.15.3.3 int FwUpgradeDialog::exec ( )

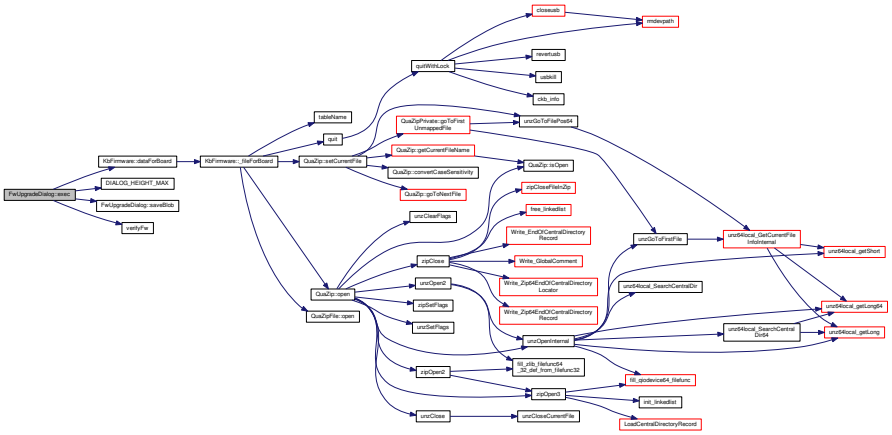
Definition at line 108 of file fwupgradedialog.cpp.

References Ui_FwUpgradeDialog::actionButton, blob, Ui_FwUpgradeDialog::cancelButton, KbFirmware::dataForBoard(), DIALOG_HEIGHT_MAX(), DIALOG_WIDTH, evLoop, exitSuccess, Kb::features, kb, Ui_FwUpgradeDialog::newLabel, Ui_FwUpgradeDialog::progressBar, saveBlob(), Ui_FwUpgradeDialog::statusLabel, ui, and verifyFw().

Referenced by KbWidget::on_fwUpdButton_clicked().

```
108                          {
109     QString features = kb->features;
110     if(!blob.isEmpty()){
111         // If a blob was already specified, check its version and validity
112         float newV = verifyFw(blob, features);
113         if(newV == 0.f){
114             QMessageBox::warning(parentWidget(), "Error", "<center>Not a valid firmware for this
    device.</center>");
115             return QDialog::Rejected;
116         }
117         ui->newLabel->setText(QString::number(newV, 'f', 2));
118     } else {
119         // Download a new blob file
120         ui->progressBar->show();
121         ui->cancelButton->setEnabled(false);
122         ui->actionButton->setEnabled(false);
123         show();
124         // This can take a while
125         blob = KbFirmware::dataForBoard(features);
126         // Check validity
127         float newV = verifyFw(blob, features);
128         if(newV == 0.f){
129             hide();
130             QMessageBox::warning(parentWidget(), "Error", "<center>There was a problem with the downloaded
    file.<br />Please try again later.</center>");
131             return QDialog::Rejected;
132         }
133     }
134     // Save temporary file
135     if(saveBlob().isEmpty()){
136         hide();
137         QMessageBox::warning(parentWidget(), "Error", "<center>Unable to save temporary file.</center>");
138         return QDialog::Rejected;
139     }
140     // Set up UI
141     ui->progressBar->setValue(0);
142     ui->progressBar->setMaximum(1);
143     ui->progressBar->setTextVisible(false);
144     ui->statusLabel->setText("Ready to install new firmware.<br /><br /><b>Disclaimer:</b>
    ckb-next is not endorsed by Corsair.<br />This is <i>unlikely</i> to cause any damage, however the developers
    of this software do not accept any responsibility in such an event.");
145     ui->cancelButton->setEnabled(true);
146     ui->actionButton->setEnabled(true);
147     setFixedSize(DIALOG_WIDTH, DIALOG_HEIGHT_MAX);
148     show();
149     // Run modal event loop
150     evLoop = new QEventLoop(this);
151     evLoop->exec();
152     delete evLoop;
153     hide();
154     return exitSuccess ? QDialog::Accepted : QDialog::Rejected;
155 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.15.3.4 void FwUpgradeDialog::fwUpdateFinished (bool *succeeded*) [private], [slot]

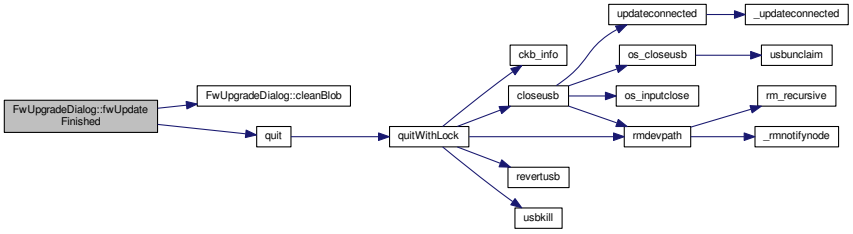
Definition at line 171 of file fwupgradedialog.cpp.

References Ui_FwUpgradeDialog::actionButton, cleanBlob(), evLoop, Ui_FwUpgradeDialog::progressBar, quit(), Ui_FwUpgradeDialog::statusLabel, and ui.

Referenced by FwUpgradeDialog(), and removeDev().

```
171                                                 {
172     cleanBlob();
173     if(succeeded)
174         ui->statusLabel->setText("Update successful!");
175     else
176         ui->statusLabel->setText("Update failed.");
177     ui->actionButton->setText("OK");
178     ui->actionButton->setEnabled(true);
179     ui->progressBar->setMaximum(1);
180     ui->progressBar->setValue(1);
181     // Exit after 10s
182     if(evLoop)
183         QTimer::singleShot(10000, evLoop, SLOT(quit()));
184 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.15.3.5 void FwUpgradeDialog::fwUpdateProgress (int *current*, int *total*) [private], [slot]

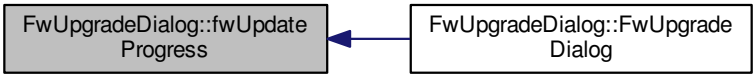
Definition at line 164 of file fwupgradedialog.cpp.

References Ui_FwUpgradeDialog::progressBar, and ui.

Referenced by FwUpgradeDialog().

```
164                                                         {
165     if(current > 0 && total > 0){
166         ui->progressBar->setMaximum(total);
167         ui->progressBar->setValue(current);
168     }
169 }
```

Here is the caller graph for this function:



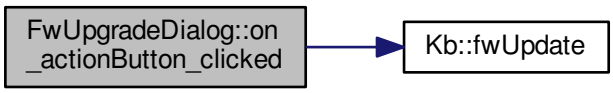
12.15.3.6 void FwUpgradeDialog::on_actionButton_clicked ( ) [private], [slot]

Definition at line 192 of file fwupgradedialog.cpp.

References Ui_FwUpgradeDialog::actionButton, Ui_FwUpgradeDialog::cancelButton, DIALOG_HEIGHT_MIN, DIALOG_WIDTH, evLoop, Kb::fwUpdate(), kb, Ui_FwUpgradeDialog::progressBar, savePath, Ui_FwUpgradeDialog::statusLabel, and ui.

```
192                                         {
193     if(!savePath.isEmpty() && kb){
194         // Start upgrade
195         setFixedSize(DIALOG_WIDTH, DIALOG_HEIGHT_MIN);
196         ui->progressBar->show();
197         ui->progressBar->setValue(0);
198         ui->progressBar->setMaximum(0);
199         ui->progressBar->setTextVisible(true);
200         ui->cancelButton->hide();
201         ui->actionButton->setEnabled(false);
202         ui->actionButton->setText("Please wait");
203         ui->statusLabel->setText("Installing firmware...");
204         kb->fwUpdate(savePath);
205     } else {
206         // Finished, close dialog.
207         if(evLoop)
208             evLoop->quit();
209     }
210 }
```

Here is the call graph for this function:



12.15.3.7 void FwUpgradeDialog::on_cancelButton_clicked ( ) [private],[slot]

Definition at line 186 of file fwupgradedialog.cpp.

References evLoop, and exitSuccess.

```
186                                         {
187     exitSuccess = false;
188     if(evLoop)
189         evLoop->quit();
190 }
```

12.15.3.8 void FwUpgradeDialog::removeDev ( ) [private],[slot]

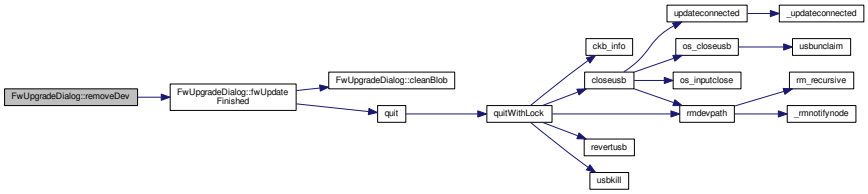
Definition at line 157 of file fwupgradedialog.cpp.

References fwUpdateFinished(), kb, and savePath.

Referenced by FwUpgradeDialog().

```
157                                 {
158     kb = 0;
159     // Assume success if upgrade in progress
160     if(!savePath.isEmpty())
161         fwUpdateFinished(true);
162 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.15.3.9 const QString & FwUpgradeDialog::saveBlob ( ) [private]

Definition at line 58 of file fwupgradedialog.cpp.

References blob, and savePath.

Referenced by exec().

```
58                                          {
59     if(!savePath.isEmpty())
60         return savePath;
61     QDir tmp = QDir::temp();
62     qint64 pid = QCoreApplication::applicationPid();
63     QString path = tmp.absoluteFilePath(QString("ckb-%1-fwblob.bin").arg(pid));
64     QFile output(path);
65     if(!output.open(QIODevice::WriteOnly)){
66         return savePath;
67     }
68     if(!output.write(blob)){
69         output.close();
70         tmp.remove(path);
71         return savePath;
72     }
73     output.close();
74     savePath = path;
75     return savePath;
76 }
```

Here is the caller graph for this function:

12.15.4 Field Documentation

12.15.4.1 QByteArray FwUpgradeDialog::blob [private]

Definition at line 41 of file fwupgradedialog.h.

Referenced by exec(), and saveBlob().

12.15.4.2 QEventLoop* FwUpgradeDialog::evLoop [private]

Definition at line 45 of file fwupgradedialog.h.

Referenced by exec(), fwUpdateFinished(), on_actionButton_clicked(), and on_cancelButton_clicked().

12.15.4.3 bool FwUpgradeDialog::exitSuccess [private]

Definition at line 46 of file fwupgradedialog.h.

Referenced by exec(), and on_cancelButton_clicked().

12.15.4.4 Kb* FwUpgradeDialog::kb [private]

Definition at line 42 of file fwupgradedialog.h.

Referenced by exec(), FwUpgradeDialog(), on_actionButton_clicked(), and removeDev().

12.15.4.5 QString FwUpgradeDialog::savePath [private]

Definition at line 37 of file fwupgradedialog.h.

Referenced by cleanBlob(), on_actionButton_clicked(), removeDev(), and saveBlob().

12.15.4.6 Ui::FwUpgradeDialog* FwUpgradeDialog::ui [private]

Definition at line 32 of file fwupgradedialog.h.

Referenced by exec(), fwUpdateFinished(), fwUpdateProgress(), FwUpgradeDialog(), on_actionButton_clicked(), and ~FwUpgradeDialog().

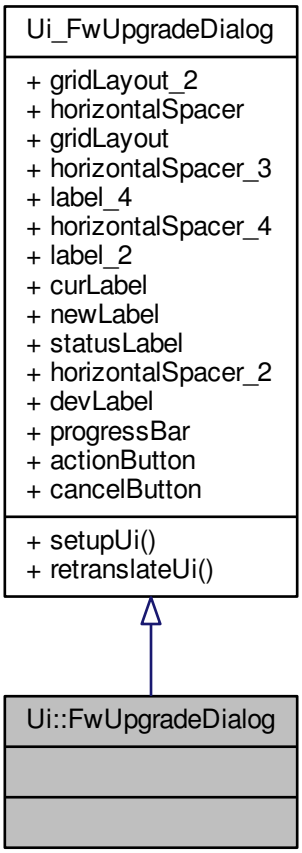
The documentation for this class was generated from the following files:

- src/ckb/fwupgradedialog.h
- src/ckb/fwupgradedialog.cpp

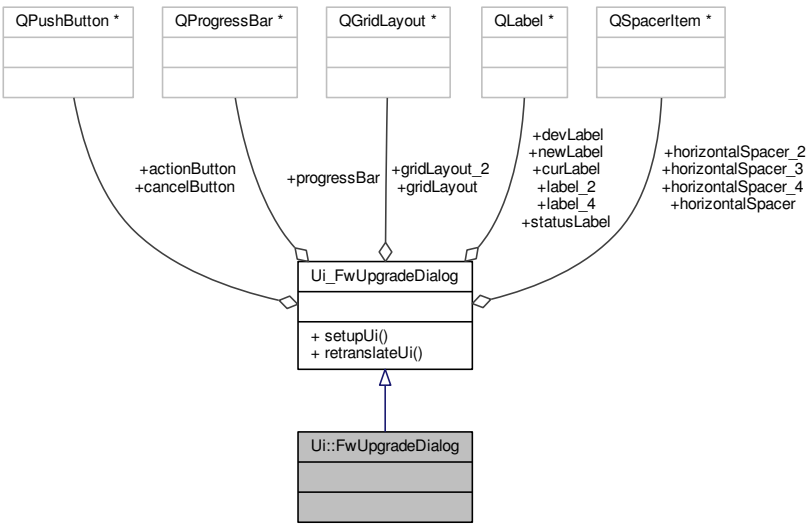
12.16 Ui::FwUpgradeDialog Class Reference

`#include <src/ckb/ui_fwupgradedialog.h>`

Inheritance diagram for Ui::FwUpgradeDialog:



Collaboration diagram for Ui::FwUpgradeDialog:



Public Member Functions

- void setupUi (QDialog *FwUpgradeDialog)
- void retranslateUi (QDialog *FwUpgradeDialog)

Data Fields

- QGridLayout * gridLayout_2
- QSpacerItem * horizontalSpacer
- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer_3
- QLabel * label_4
- QSpacerItem * horizontalSpacer_4
- QLabel * label_2
- QLabel * curLabel
- QLabel * newLabel
- QLabel * statusLabel
- QSpacerItem * horizontalSpacer_2
- QLabel * devLabel
- QProgressBar * progressBar
- QPushButton * actionButton
- QPushButton * cancelButton

12.16.1 Detailed Description

Definition at line 172 of file ui_fwupgradedialog.h.

12.16.2 Member Function Documentation

12.16.2.1 void Ui_FwUpgradeDialog::retranslateUi (QDialog * *FwUpgradeDialog*) [inline],[inherited]

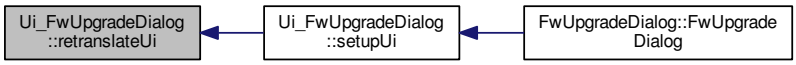
Definition at line 156 of file ui_fwupgradedialog.h.

References Ui_FwUpgradeDialog::actionButton, Ui_FwUpgradeDialog::cancelButton, Ui_FwUpgradeDialog::curLabel, Ui_FwUpgradeDialog::devLabel, Ui_FwUpgradeDialog::label_2, Ui_FwUpgradeDialog::label_4, Ui_FwUpgradeDialog::newLabel, and Ui_FwUpgradeDialog::statusLabel.

Referenced by Ui_FwUpgradeDialog::setupUi().

```
157     {
158         FwUpgradeDialog->setWindowTitle(QApplication::translate("FwUpgradeDialog", "Firmware update",
    Q_NULLPTR));
159         label_4->setText(QApplication::translate("FwUpgradeDialog", "New:", Q_NULLPTR));
160         label_2->setText(QApplication::translate("FwUpgradeDialog", "Current:", Q_NULLPTR));
161         curLabel->setText(QApplication::translate("FwUpgradeDialog", "0.00", Q_NULLPTR));
162         newLabel->setText(QApplication::translate("FwUpgradeDialog", "0.00", Q_NULLPTR));
163         statusLabel->setText(QApplication::translate("FwUpgradeDialog", "Downloading new
     firmware, please wait...", Q_NULLPTR));
164         devLabel->setText(QApplication::translate("FwUpgradeDialog", "Device", Q_NULLPTR));
165         actionButton->setText(QApplication::translate("FwUpgradeDialog", "Start", Q_NULLPTR));
166         cancelButton->setText(QApplication::translate("FwUpgradeDialog", "Cancel", Q_NULLPTR));
167     } // retranslateUi
```

Here is the caller graph for this function:



12.16.2.2 void Ui_FwUpgradeDialog::setupUi (QDialog * *FwUpgradeDialog*) [inline],[inherited]

Definition at line 45 of file ui_fwupgradedialog.h.

References Ui_FwUpgradeDialog::actionButton, Ui_FwUpgradeDialog::cancelButton, Ui_FwUpgradeDialog::curLabel, Ui_FwUpgradeDialog::devLabel, Ui_FwUpgradeDialog::gridLayout, Ui_FwUpgradeDialog::gridLayout_2, Ui_FwUpgradeDialog::horizontalSpacer, Ui_FwUpgradeDialog::horizontalSpacer_2, Ui_FwUpgradeDialog::horizontalSpacer_3, Ui_FwUpgradeDialog::horizontalSpacer_4, Ui_FwUpgradeDialog::label_2, Ui_FwUpgradeDialog::label_4, Ui_FwUpgradeDialog::newLabel, Ui_FwUpgradeDialog::progressBar, Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::statusLabel.

Referenced by FwUpgradeDialog::FwUpgradeDialog().

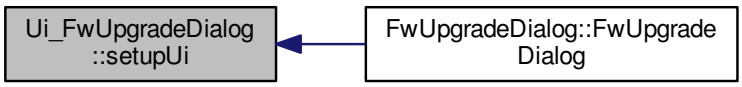
```
46     {
47         if (FwUpgradeDialog->objectName().isEmpty())
48             FwUpgradeDialog->setObjectName(QStringLiteral("FwUpgradeDialog"));
49         FwUpgradeDialog->resize(420, 280);
50         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
51         sizePolicy.setHorizontalStretch(0);
52         sizePolicy.setVerticalStretch(0);
53         sizePolicy.setHeightForWidth(FwUpgradeDialog->sizePolicy().hasHeightForWidth());
54         FwUpgradeDialog->setSizePolicy(sizePolicy);
55         gridLayout_2 = new QGridLayout(FwUpgradeDialog);
56         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
57         horizontalSpacer = new QSpacerItem(127, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
58
59         gridLayout_2->addItem(horizontalSpacer, 1, 0, 1, 1);
60
61         gridLayout = new QGridLayout();
62         gridLayout->setObjectName(QStringLiteral("gridLayout"));
63         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
```

```
64
65          gridLayout->addItem(horizontalSpacer_3, 1, 0, 1, 1);
66
67          label_4 = new QLabel(FwUpgradeDialog);
68          label_4->setObjectName(QStringLiteral("label_4"));
69          QFont font;
70          font.setBold(true);
71          font.setWeight(75);
72          label_4->setFont(font);
73
74          gridLayout->addWidget(label_4, 1, 4, 1, 1);
75
76          horizontalSpacer_4 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
      QSizePolicy::Minimum);
77
78          gridLayout->addItem(horizontalSpacer_4, 1, 6, 1, 1);
79
80          label_2 = new QLabel(FwUpgradeDialog);
81          label_2->setObjectName(QStringLiteral("label_2"));
82          label_2->setFont(font);
83
84          gridLayout->addWidget(label_2, 1, 1, 1, 1);
85
86          curLabel = new QLabel(FwUpgradeDialog);
87          curLabel->setObjectName(QStringLiteral("curLabel"));
88
89          gridLayout->addWidget(curLabel, 1, 2, 1, 1);
90
91          newLabel = new QLabel(FwUpgradeDialog);
92          newLabel->setObjectName(QStringLiteral("newLabel"));
93
94          gridLayout->addWidget(newLabel, 1, 5, 1, 1);
95
96          statusLabel = new QLabel(FwUpgradeDialog);
97          statusLabel->setObjectName(QStringLiteral("statusLabel"));
98          QSizePolicy sizePolicy1(QSizePolicy::Preferred, QSizePolicy::Expanding);
99          sizePolicy1.setHorizontalStretch(0);
100          sizePolicy1.setVerticalStretch(0);
101          sizePolicy1.setHeightForWidth(statusLabel->sizePolicy().hasHeightForWidth());
102          statusLabel->setSizePolicy(sizePolicy1);
103          statusLabel->setAlignment(Qt::AlignCenter);
104          statusLabel->setWordWrap(true);
105
106          gridLayout->addWidget(statusLabel, 2, 0, 1, 7);
107
108          horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
      QSizePolicy::Minimum);
109
110          gridLayout->addItem(horizontalSpacer_2, 1, 3, 1, 1);
111
112          devLabel = new QLabel(FwUpgradeDialog);
113          devLabel->setObjectName(QStringLiteral("devLabel"));
114          QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Minimum);
115          sizePolicy2.setHorizontalStretch(0);
116          sizePolicy2.setVerticalStretch(0);
117          sizePolicy2.setHeightForWidth(devLabel->sizePolicy().hasHeightForWidth());
118          devLabel->setSizePolicy(sizePolicy2);
119          devLabel->setFont(font);
120          devLabel->setAlignment(Qt::AlignCenter);
121
122          gridLayout->addWidget(devLabel, 0, 0, 1, 7);
123
124          progressBar = new QProgressBar(FwUpgradeDialog);
125          progressBar->setObjectName(QStringLiteral("progressBar"));
126          progressBar->setMaximum(0);
127          progressBar->setValue(0);
128
129          gridLayout->addWidget(progressBar, 3, 0, 1, 7);
130
131
132          gridLayout_2->addLayout(gridLayout, 0, 0, 1, 3);
133
134          actionButton = new QPushButton(FwUpgradeDialog);
135          actionButton->setObjectName(QStringLiteral("actionButton"));
136          actionButton->setEnabled(false);
137
138          gridLayout_2->addWidget(actionButton, 1, 2, 1, 1);
139
140          cancelButton = new QPushButton(FwUpgradeDialog);
141          cancelButton->setObjectName(QStringLiteral("cancelButton"));
142          cancelButton->setEnabled(false);
143
144          gridLayout_2->addWidget(cancelButton, 1, 1, 1, 1);
145
146          QWidget::setTabOrder(actionButton, cancelButton);
147
148          retranslateUi(FwUpgradeDialog);
```

```
149
150         actionButton->setDefault(true);
151
152
153         QMetaObject::connectSlotsByName(FwUpgradeDialog);
154     } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.16.3 Field Documentation

12.16.3.1 QPushButton* Ui_FwUpgradeDialog::actionButton [inherited]

Definition at line 42 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::fwUpdateFinished(), FwUpgradeDialog::on_actionButton_clicked(), Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.2 QPushButton* Ui_FwUpgradeDialog::cancelButton [inherited]

Definition at line 43 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::on_actionButton_clicked(), Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.3 QLabel* Ui_FwUpgradeDialog::curLabel [inherited]

Definition at line 36 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.4 QLabel* Ui_FwUpgradeDialog::devLabel [inherited]

Definition at line 40 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.5 QGridLayout* Ui_FwUpgradeDialog::gridLayout [inherited]

Definition at line 31 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::setupUi().

12.16.3.6 QGridLayout* Ui_FwUpgradeDialog::gridLayout_2 [inherited]

Definition at line 29 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::setupUi().

12.16.3.7 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer [inherited]

Definition at line 30 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::setupUi().

12.16.3.8 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer_2 [inherited]

Definition at line 39 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::setupUi().

12.16.3.9 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer_3 [inherited]

Definition at line 32 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::setupUi().

12.16.3.10 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer_4 [inherited]

Definition at line 34 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::setupUi().

12.16.3.11 QLabel* Ui_FwUpgradeDialog::label_2 [inherited]

Definition at line 35 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.12 QLabel* Ui_FwUpgradeDialog::label_4 [inherited]

Definition at line 33 of file ui_fwupgradedialog.h.

Referenced by Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.13 QLabel* Ui_FwUpgradeDialog::newLabel [inherited]

Definition at line 37 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::FwUpgradeDialog(), Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.14 QProgressBar* Ui_FwUpgradeDialog::progressBar [inherited]

Definition at line 41 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::fwUpdateFinished(), FwUpgradeDialog::fwUpdateProgress(), FwUpgradeDialog::on_actionButton_clicked(), and Ui_FwUpgradeDialog::setupUi().

12.16.3.15 QLabel* Ui_FwUpgradeDialog::statusLabel [inherited]

Definition at line 38 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::fwUpdateFinished(), FwUpgradeDialog::on_actionButton_clicked(), Ui_FwUpgradeDialog::retranslateUi(), and Ui_FwUpgradeDialog::setupUi().

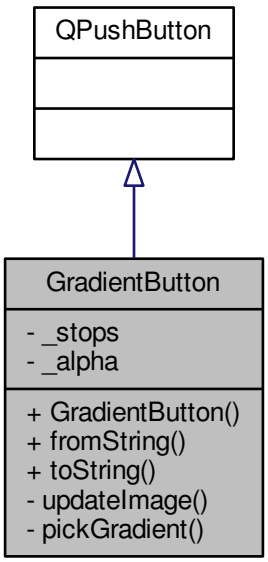
The documentation for this class was generated from the following file:

- src/ckb/ui_fwupgradedialog.h

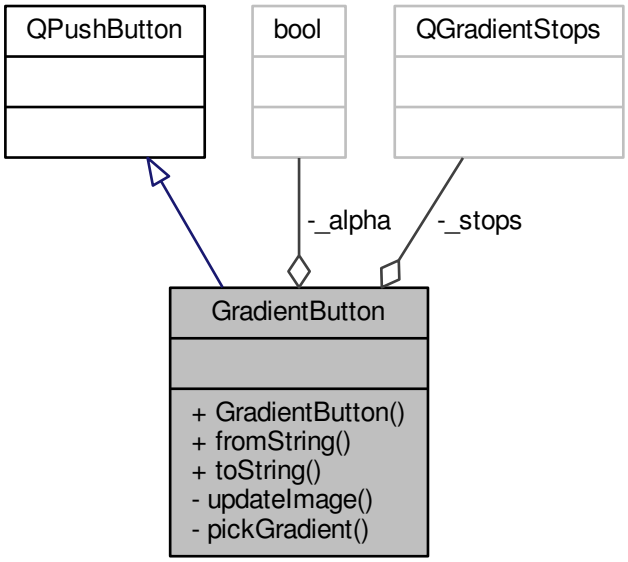
12.17 GradientButton Class Reference

`#include <src/ckb/gradientbutton.h>`

Inheritance diagram for GradientButton:



Collaboration diagram for GradientButton:



Signals

- void gradientChanged ()

Public Member Functions

- GradientButton (QWidget *parent=0, bool allowAlpha=false)
- void fromString (const QString &string)
- QString toString () const

Private Slots

- void pickGradient ()

Private Member Functions

- void updateImage ()

Private Attributes

- QGradientStops _stops
- bool _alpha

12.17.1 Detailed Description

Definition at line 6 of file gradientbutton.h.

12.17.2 Constructor & Destructor Documentation

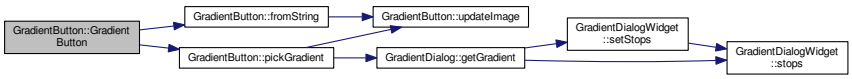
12.17.2.1 GradientButton::GradientButton (QWidget * *parent* = 0, bool *allowAlpha* = false) [explicit]

Definition at line 7 of file gradientbutton.cpp.

References fromString(), and pickGradient().

```
7                                                                   :
8     QPushButton(parent), _alpha(allowAlpha)
9 {
10     connect(this, SIGNAL(clicked()), this, SLOT(pickGradient()));
11     setAutoDefault(false);
12     setDefault(false);
13     fromString("");
14     setText("");
15 }
```

Here is the call graph for this function:



12.17.3 Member Function Documentation

12.17.3.1 void GradientButton::fromString (const QString & *string*)

Definition at line 38 of file gradientbutton.cpp.

References _alpha, _stops, and updateImage().

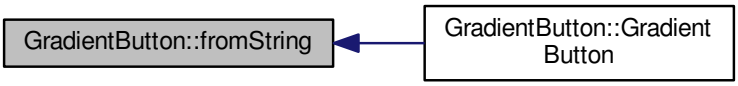
Referenced by GradientButton().

```
38                                                  {
39     // Clear existing stops
40     _stops.clear();
41     // Parse string with sscanf
42     QByteArray cString = string.toLatin1();
43     const char* data = cString.data();
44     char pos = -1;
45     uchar a, r, g, b;
46     while(1){
47         int scanned = 0;
48         char newpos;
49         if(sscanf(data, "%hhd:%2hhx%2hhx%2hhx%2hhx%n", &newpos, &a, &r, &g, &b, &scanned) != 5)
50             break;
51         data += scanned;
52         // Don't allow stops out-of-order or past 100
53         if(newpos <= pos || newpos > 100)
54             break;
55         pos = newpos;
56         if(!_alpha)
57             a = 255;
58         _stops.append(QGradientStop(pos / 100., QColor(r, g, b, a)));
59     }
60     if(_stops.count() == 0){
61         // If nothing was read, try a single ARGB constant.
62         if(sscanf(data, "%2hhx%2hhx%2hhx%2hhx", &a, &r, &g, &b) == 4){
63             _stops.append(QGradientStop(0., QColor(r, g, b, a)));
64             _stops.append(QGradientStop(1., QColor(r, g, b, 0.)));
65         }
66     }
67     // If that still didn't work, fill with white
68     if(_stops.count() == 0){
69         _stops.append(QGradientStop(0., QColor(255, 255, 255)));
70         _stops.append(QGradientStop(1., QColor(255, 255, 255)));
71     }
72     // Make sure there are stops at 0 and at 100
73     if(_stops[0].first != 0.)
74         _stops.prepend(QGradientStop(0., _stops[0].second));
```

```
75     if(_stops.last().first != 1.)
76         _stops.prepend(QGradientStop(1., _stops.last().second));
77     updateImage();
78 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.17.3.2 void GradientButton::gradientChanged ( ) [signal]

Definition at line 132 of file moc_gradientbutton.cpp.

Referenced by pickGradient().

```
133 {
134     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
135 }
```

Here is the caller graph for this function:

12.17.3.3 void GradientButton::pickGradient ( ) [private], [slot]

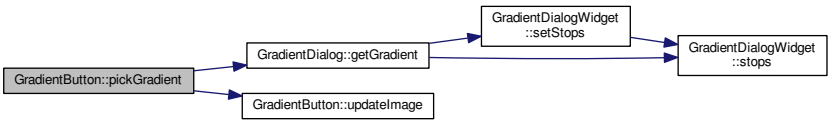
Definition at line 90 of file gradientbutton.cpp.

References _stops, GradientDialog::getGradient(), gradientChanged(), and updateImage().

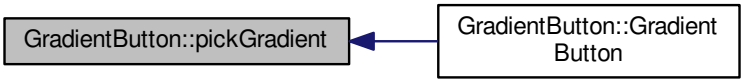
Referenced by GradientButton().

```
90                                  {
91     GradientDialog dialog(this);
92     _stops = dialog.getGradient(_stops);
93     updateImage();
94     emit gradientChanged();
95 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.17.3.4 QString GradientButton::toString ( ) const

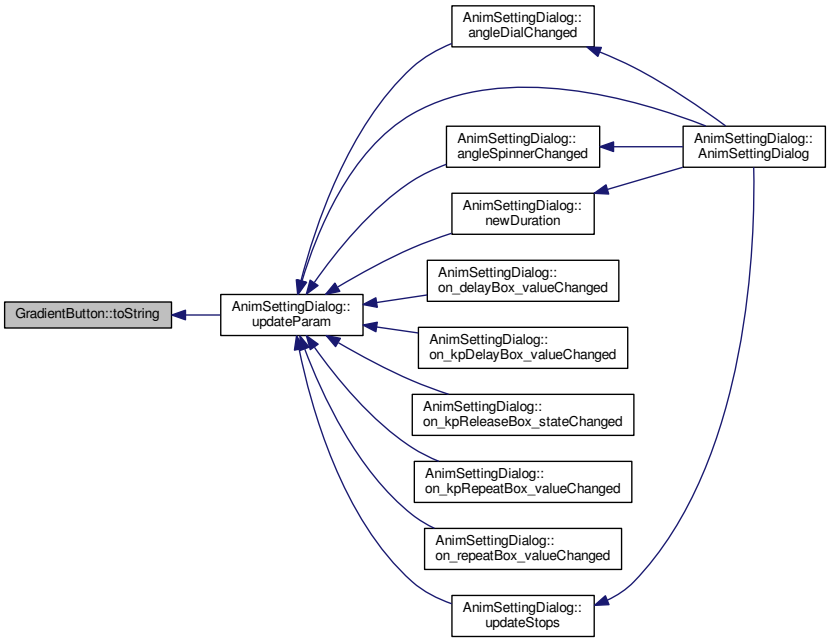
Definition at line 80 of file gradientbutton.cpp.

References _stops.

Referenced by AnimSettingDialog::updateParam().

```
80                                      {
81     QStringList result;
82     foreach(const QGradientStop& stop, _stops){
83         QString string;
84         const QColor& color = stop.second;
85         result << string.sprintf("%d:%02x%02x%02x%02x", (int)round(stop.first * 100.f), color.alpha(),
       color.red(), color.green(), color.blue());
86     }
87     return result.join(" ");
88 }
```

Here is the caller graph for this function:



12.17.3.5 void GradientButton::updateImage ( ) [private]

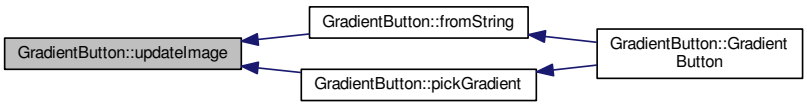
Definition at line 17 of file gradientbutton.cpp.

References _alpha, _stops, and x.

Referenced by fromString(), and pickGradient().

```
17                                {
18     const int w = 130, h = 16;
19     QImage image(w, h, QImage::Format_RGB888);
20     QPainter painter(&image);
21     painter.setPen(Qt::NoPen);
22     painter.fillRect(0, 0, w, h, QColor(0, 0, 0));
23     if(_alpha){
24         for(int x = 1; x < w - 1; x += h){
25             painter.fillRect(x, 1, h / 2, h / 2 - 1, QColor(255, 255, 255));
26             painter.fillRect(x + h / 2, 1, h / 2, h / 2 - 1, QColor(192, 192, 192));
27             painter.fillRect(x, h / 2, h / 2, h / 2 - 1, QColor(192, 192, 192));
28             painter.fillRect(x + h / 2, h / 2, h / 2, h / 2 - 1, QColor(255, 255, 255));
29         }
30     }
31     QLinearGradient gradient(1., 0., w - 1, 0.);
32     gradient.setStops(_stops);
33     painter.fillRect(1, 1, w - 2, h - 2, QBrush(gradient));
34     setIconSize(QSize(w, h));
35     setIcon(QIcon(QPixmap::fromImage(image)));
36 }
```

Here is the caller graph for this function:



12.17.4 Field Documentation

12.17.4.1 bool GradientButton::_alpha [private]

Definition at line 23 of file gradientbutton.h.

Referenced by fromString(), and updateImage().

12.17.4.2 QGradientStops GradientButton::_stops [private]

Definition at line 22 of file gradientbutton.h.

Referenced by fromString(), pickGradient(), toString(), and updateImage().

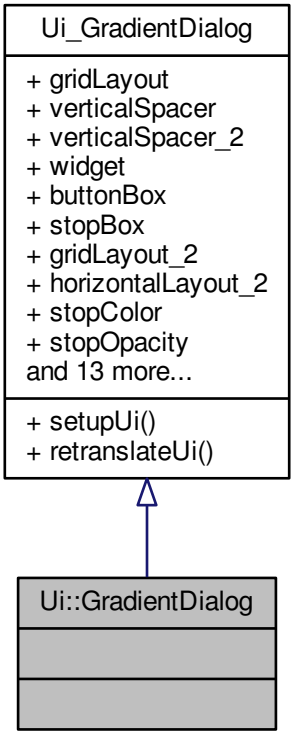
The documentation for this class was generated from the following files:

- src/ckb/gradientbutton.h
- src/ckb/gradientbutton.cpp
- src/ckb/release/moc_gradientbutton.cpp

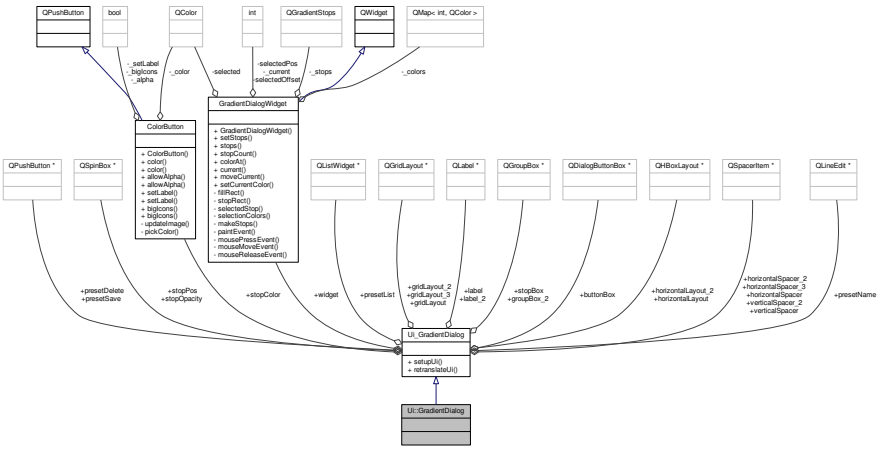
12.18 Ui::GradientDialog Class Reference

`#include <src/ckb/ui_gradientdialog.h>`

Inheritance diagram for Ui::GradientDialog:



Collaboration diagram for Ui::GradientDialog:



Public Member Functions

- void setupUi (QDialog ∗GradientDialog)
- void retranslateUi (QDialog ∗GradientDialog)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_2
- GradientDialogWidget * widget
- QDialogButtonBox * buttonBox
- QGroupBox * stopBox
- QGridLayout * gridLayout_2
- QHBoxLayout * horizontalLayout_2
- ColorButton * stopColor
- QSpinBox * stopOpacity
- QSpacerItem * horizontalSpacer_2
- QLabel * label
- QHBoxLayout * horizontalLayout
- QSpinBox * stopPos
- QSpacerItem * horizontalSpacer
- QLabel * label_2
- QSpacerItem * horizontalSpacer_3
- QGroupBox * groupBox_2
- QGridLayout * gridLayout_3
- QListWidget * presetList
- QLineEdit * presetName
- QPushButton * presetSave
- QPushButton * presetDelete

12.18.1 Detailed Description

Definition at line 231 of file ui_gradientdialog.h.

12.18.2 Member Function Documentation

12.18.2.1 void Ui_GradientDialog::retranslateUi (QDialog * *GradientDialog*) [inline], [inherited]

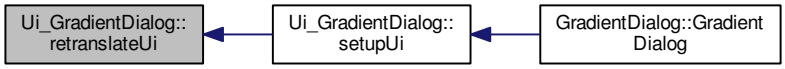
Definition at line 213 of file ui_gradientdialog.h.

References Ui_GradientDialog::groupBox_2, Ui_GradientDialog::label, Ui_GradientDialog::label_2, Ui_GradientDialog::presetDelete, Ui_GradientDialog::presetName, Ui_GradientDialog::presetSave, Ui_GradientDialog::stopBox, Ui_GradientDialog::stopColor, Ui_GradientDialog::stopOpacity, and Ui_GradientDialog::stopPos.

Referenced by Ui_GradientDialog::setupUi().

```
214     {
215         GradientDialog->setWindowTitle(QApplication::translate("GradientDialog", "Pick Gradient", Q_NULLPTR
    ));
216         stopBox->setTitle(QApplication::translate("GradientDialog", "Point X of Y", Q_NULLPTR));
217         stopColor->setText(QApplication::translate("GradientDialog", "Pick Color...", Q_NULLPTR));
218         stopOpacity->setSuffix(QApplication::translate("GradientDialog", "%", Q_NULLPTR));
219         label->setText(QApplication::translate("GradientDialog", "Position:", Q_NULLPTR));
220         stopPos->setSuffix(QApplication::translate("GradientDialog", "%", Q_NULLPTR));
221         label_2->setText(QApplication::translate("GradientDialog", "Color:", Q_NULLPTR));
222         groupBox_2->setTitle(QApplication::translate("GradientDialog", "Presets", Q_NULLPTR));
223         presetName->setPlaceholderText(QApplication::translate("GradientDialog", "Name",
    Q_NULLPTR));
224         presetSave->setText(QApplication::translate("GradientDialog", "Save", Q_NULLPTR));
225         presetDelete->setText(QApplication::translate("GradientDialog", "Delete", Q_NULLPTR));
226     } // retranslateUi
```

Here is the caller graph for this function:



12.18.2.2 void Ui_GradientDialog::setupUi (QDialog * *GradientDialog*) [inline],[inherited]

Definition at line 60 of file ui_gradientdialog.h.

References Ui_GradientDialog::buttonBox, Ui_GradientDialog::gridLayout, Ui_GradientDialog::gridLayout_2, Ui_GradientDialog::gridLayout_3, Ui_GradientDialog::groupBox_2, Ui_GradientDialog::horizontalLayout, Ui_GradientDialog::horizontalLayout_2, Ui_GradientDialog::horizontalSpacer, Ui_GradientDialog::horizontalSpacer_2, Ui_GradientDialog::horizontalSpacer_3, Ui_GradientDialog::label, Ui_GradientDialog::label_2, Ui_GradientDialog::presetDelete, Ui_GradientDialog::presetList, Ui_GradientDialog::presetName, Ui_GradientDialog::presetSave, Ui_GradientDialog::retranslateUi(), Ui_GradientDialog::stopBox, Ui_GradientDialog::stopColor, Ui_GradientDialog::stopOpacity, Ui_GradientDialog::stopPos, Ui_GradientDialog::verticalSpacer, Ui_GradientDialog::verticalSpacer_2, and Ui_GradientDialog::widget.

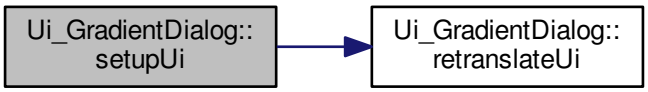
Referenced by GradientDialog::GradientDialog().

```
61     {
62         if (GradientDialog->objectName().isEmpty())
63             GradientDialog->setObjectName(QStringLiteral("GradientDialog"));
64         GradientDialog->resize(400, 440);
65         GradientDialog->setMinimumSize(QSize(400, 440));
66         GradientDialog->setMaximumSize(QSize(400, 440));
67         GradientDialog->setFocusPolicy(Qt::ClickFocus);
68         gridLayout = new QGridLayout(GradientDialog);
69         gridLayout->setObjectName(QStringLiteral("gridLayout"));
70         verticalSpacer = new QSpacerItem(0, 0, QSizePolicy::Minimum, QSizePolicy::Expanding);
71 
72         gridLayout->addItem(verticalSpacer, 1, 0, 1, 1);
73 
74         verticalSpacer_2 = new QSpacerItem(0, 15, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
75 
76         gridLayout->addItem(verticalSpacer_2, 3, 0, 1, 1);
77 
78         widget = new GradientDialogWidget(GradientDialog);
79         widget->setObjectName(QStringLiteral("widget"));
80         QSizePolicy sizePolicy(QSizePolicy::Preferred, QSizePolicy::Fixed);
81         sizePolicy.setHorizontalStretch(0);
82         sizePolicy.setVerticalStretch(0);
83         sizePolicy.setHeightForWidth(widget->sizePolicy().hasHeightForWidth());
84         widget->setSizePolicy(sizePolicy);
85         widget->setMinimumSize(QSize(0, 74));
86         widget->setMaximumSize(QSize(16777215, 74));
87 
88         gridLayout->addWidget(widget, 2, 0, 1, 1);
89 
90         buttonBox = new QDialogButtonBox(GradientDialog);
91         buttonBox->setObjectName(QStringLiteral("buttonBox"));
92         buttonBox->setOrientation(Qt::Horizontal);
93         buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
94 
95         gridLayout->addWidget(buttonBox, 5, 0, 1, 1);
96 
97         stopBox = new QGroupBox(GradientDialog);
98         stopBox->setObjectName(QStringLiteral("stopBox"));
99         sizePolicy.setHeightForWidth(stopBox->sizePolicy().hasHeightForWidth());
100         stopBox->setSizePolicy(sizePolicy);
101         gridLayout_2 = new QGridLayout(stopBox);
102         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
103         horizontalLayout_2 = new QHBoxLayout();
104         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
105         stopColor = new ColorButton(stopBox);
106         stopColor->setObjectName(QStringLiteral("stopColor"));
107 
```

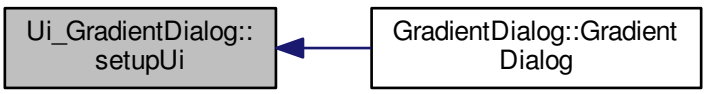
```
108         horizontalLayout_2->addWidget(stopColor);
109 
110         stopOpacity = new QSpinBox(stopBox);
111         stopOpacity->setObjectName(QStringLiteral("stopOpacity"));
112         stopOpacity->setMaximum(100);
113 
114         horizontalLayout_2->addWidget(stopOpacity);
115 
116         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
117 
118         horizontalLayout_2->addItem(horizontalSpacer_2);
119 
120 
121         gridLayout_2->addLayout(horizontalLayout_2, 1, 2, 1, 1);
122 
123         label = new QLabel(stopBox);
124         label->setObjectName(QStringLiteral("label"));
125 
126         gridLayout_2->addWidget(label, 0, 0, 1, 1);
127 
128         horizontalLayout = new QHBoxLayout();
129         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
130         stopPos = new QSpinBox(stopBox);
131         stopPos->setObjectName(QStringLiteral("stopPos"));
132         stopPos->setMaximum(100);
133 
134         horizontalLayout->addWidget(stopPos);
135 
136         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
137 
138         horizontalLayout->addItem(horizontalSpacer);
139 
140 
141         gridLayout_2->addLayout(horizontalLayout, 0, 2, 1, 1);
142 
143         label_2 = new QLabel(stopBox);
144         label_2->setObjectName(QStringLiteral("label_2"));
145 
146         gridLayout_2->addWidget(label_2, 1, 0, 1, 1);
147 
148         horizontalSpacer_3 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
149 
150         gridLayout_2->addItem(horizontalSpacer_3, 0, 1, 1, 1);
151 
152 
153         gridLayout->addWidget(stopBox, 4, 0, 1, 1);
154 
155         groupBox_2 = new QGroupBox(GradientDialog);
156         groupBox_2->setObjectName(QStringLiteral("groupBox_2"));
157         sizePolicy.setHeightForWidth(groupBox_2->sizePolicy().hasHeightForWidth());
158         groupBox_2->setSizePolicy(sizePolicy);
159         gridLayout_3 = new QGridLayout(groupBox_2);
160         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
161         presetList = new QListWidget(groupBox_2);
162         presetList->setObjectName(QStringLiteral("presetList"));
163         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Preferred);
164         sizePolicy1.setHorizontalStretch(0);
165         sizePolicy1.setVerticalStretch(0);
166         sizePolicy1.setHeightForWidth(presetList->sizePolicy().hasHeightForWidth());
167         presetList->setSizePolicy(sizePolicy1);
168         presetList->setFrameShadow(QFrame::Plain);
169         presetList->setHorizontalScrollBarPolicy(Qt::ScrollBarAlwaysOn);
170         presetList->setIconSize(QSize(64, 64));
171         presetList->setHorizontalScrollMode(QAbstractItemView::ScrollPerPixel);
172         presetList->setFlow(QListView::TopToBottom);
173         presetList->setViewMode(QListView::IconMode);
174         presetList->setUniformItemSizes(true);
175 
176         gridLayout_3->addWidget(presetList, 0, 0, 3, 1);
177 
178         presetName = new QLineEdit(groupBox_2);
179         presetName->setObjectName(QStringLiteral("presetName"));
180         presetName->setMaxLength(30);
181 
182         gridLayout_3->addWidget(presetName, 0, 1, 1, 1);
183 
184         presetSave = new QPushButton(groupBox_2);
185         presetSave->setObjectName(QStringLiteral("presetSave"));
186         presetSave->setAutoDefault(false);
187 
188         gridLayout_3->addWidget(presetSave, 1, 1, 1, 1);
189 
190         presetDelete = new QPushButton(groupBox_2);
191         presetDelete->setObjectName(QStringLiteral("presetDelete"));
```

```
192         presetDelete->setAutoDefault(false);
193
194         gridLayout_3->addWidget(presetDelete, 2, 1, 1, 1);
195
196
197         gridLayout->addWidget(groupBox_2, 0, 0, 1, 1);
198
199         QWidget::setTabOrder(presetList, presetName);
200         QWidget::setTabOrder(presetName, presetSave);
201         QWidget::setTabOrder(presetSave, presetDelete);
202         QWidget::setTabOrder(presetDelete, stopPos);
203         QWidget::setTabOrder(stopPos, stopColor);
204         QWidget::setTabOrder(stopColor, stopOpacity);
205
206         retranslateUi(GradientDialog);
207         QObject::connect(buttonBox, SIGNAL(accepted()), GradientDialog, SLOT(accept()));
208         QObject::connect(buttonBox, SIGNAL(rejected()), GradientDialog, SLOT(reject()));
209
210         QMetaObject::connectSlotsByName(GradientDialog);
211     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.18.3 Field Documentation

12.18.3.1 QDialogButtonBox* Ui_GradientDialog::buttonBox [inherited]

Definition at line 40 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.2 QGridLayout* Ui_GradientDialog::gridLayout [inherited]

Definition at line 36 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.3 QGridLayout* Ui_GradientDialog::gridLayout_2 [inherited]

Definition at line 42 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.4 QGridLayout* Ui_GradientDialog::gridLayout_3 [inherited]

Definition at line 54 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.5 QGroupBox* Ui_GradientDialog::groupBox_2 [inherited]

Definition at line 53 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.6 QHBoxLayout* Ui_GradientDialog::horizontalLayout [inherited]

Definition at line 48 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.7 QHBoxLayout* Ui_GradientDialog::horizontalLayout_2 [inherited]

Definition at line 43 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.8 QSpacerItem* Ui_GradientDialog::horizontalSpacer [inherited]

Definition at line 50 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.9 QSpacerItem* Ui_GradientDialog::horizontalSpacer_2 [inherited]

Definition at line 46 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.10 QSpacerItem* Ui_GradientDialog::horizontalSpacer_3 [inherited]

Definition at line 52 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.11 QLabel* Ui_GradientDialog::label [inherited]

Definition at line 47 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.12 QLabel* Ui_GradientDialog::label_2 [inherited]

Definition at line 51 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.13 QPushButton* Ui_GradientDialog::presetDelete [inherited]

Definition at line 58 of file ui_gradientdialog.h.

Referenced by GradientDialog::getGradient(), GradientDialog::on_presetDelete_clicked(), GradientDialog::on_presetSave_clicked(), Ui_GradientDialog::retranslateUi(), GradientDialog::setPreset(), and Ui_GradientDialog::setupUi().

12.18.3.14 QListWidget* Ui_GradientDialog::presetList [inherited]

Definition at line 55 of file ui_gradientdialog.h.

Referenced by GradientDialog::setPreset(), Ui_GradientDialog::setupUi(), and GradientDialog::updatePresets().

12.18.3.15 QLineEdit* Ui_GradientDialog::presetName [inherited]

Definition at line 56 of file ui_gradientdialog.h.

Referenced by GradientDialog::on_presetSave_clicked(), Ui_GradientDialog::retranslateUi(), GradientDialog::setPreset(), and Ui_GradientDialog::setupUi().

12.18.3.16 QPushButton* Ui_GradientDialog::presetSave [inherited]

Definition at line 57 of file ui_gradientdialog.h.

Referenced by GradientDialog::getGradient(), GradientDialog::on_presetDelete_clicked(), GradientDialog::on_presetSave_clicked(), Ui_GradientDialog::retranslateUi(), GradientDialog::setPreset(), and Ui_GradientDialog::setupUi().

12.18.3.17 QGroupBox* Ui_GradientDialog::stopBox [inherited]

Definition at line 41 of file ui_gradientdialog.h.

Referenced by GradientDialog::currentChanged(), Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.18 ColorButton* Ui_GradientDialog::stopColor [inherited]

Definition at line 44 of file ui_gradientdialog.h.

Referenced by GradientDialog::currentChanged(), GradientDialog::GradientDialog(), GradientDialog::on_stopOpacity_valueChanged(), Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.19 QSpinBox* Ui_GradientDialog::stopOpacity [inherited]

Definition at line 45 of file ui_gradientdialog.h.

Referenced by GradientDialog::colorChanged(), GradientDialog::currentChanged(), Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.20 QSpinBox* Ui_GradientDialog::stopPos [inherited]

Definition at line 49 of file ui_gradientdialog.h.

Referenced by GradientDialog::currentChanged(), GradientDialog::on_stopPos_valueChanged(), Ui_GradientDialog::retranslateUi(), and Ui_GradientDialog::setupUi().

12.18.3.21 QSpacerItem* Ui_GradientDialog::verticalSpacer [inherited]

Definition at line 37 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.22 QSpacerItem* Ui_GradientDialog::verticalSpacer_2 [inherited]

Definition at line 38 of file ui_gradientdialog.h.

Referenced by Ui_GradientDialog::setupUi().

12.18.3.23 GradientDialogWidget* Ui_GradientDialog::widget [inherited]

Definition at line 39 of file ui_gradientdialog.h.

Referenced by GradientDialog::colorChanged(), GradientDialog::currentChanged(), GradientDialog::getGradient(), GradientDialog::GradientDialog(), GradientDialog::on_presetSave_clicked(), GradientDialog::on_stopPos_valueChanged(), GradientDialog::setPreset(), and Ui_GradientDialog::setupUi().

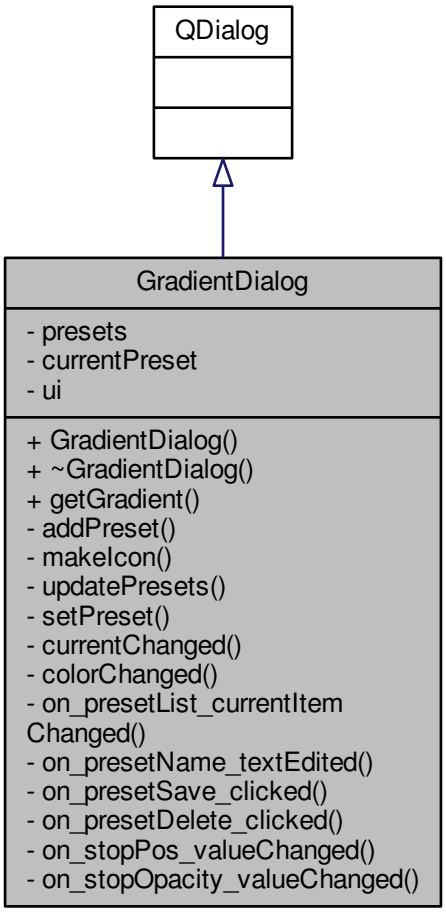
The documentation for this class was generated from the following file:

- src/ckb/ui_gradientdialog.h

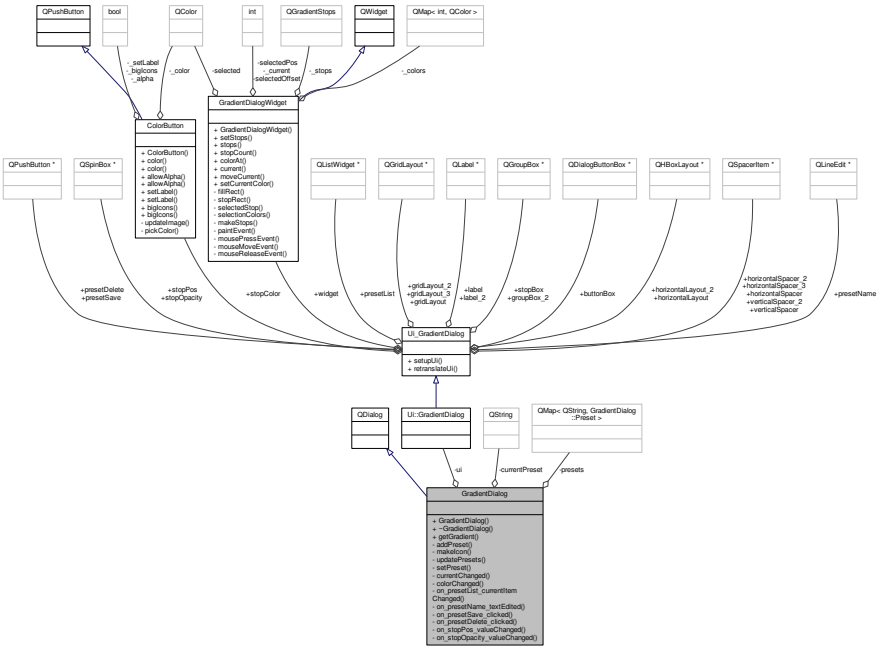
12.19 GradientDialog Class Reference

`#include <src/ckb/gradientdialog.h>`

Inheritance diagram for GradientDialog:



Collaboration diagram for GradientDialog:



Data Structures

- struct Preset

Public Member Functions

- GradientDialog (QWidget *parent=0)
- ~GradientDialog ()
- QGradientStops getGradient (const QGradientStops &prevGradient)

Private Slots

- void currentChanged (QColor color, bool spontaneous, int position)
- void colorChanged (QColor color)
- void on_presetList_currentItemChanged (QListWidgetItem *current, QListWidgetItem *previous)
- void on_presetName_textEdited (const QString &arg1)
- void on_presetSave_clicked ()
- void on_presetDelete_clicked ()
- void on_stopPos_valueChanged (int arg1)
- void on_stopOpacity_valueChanged (int arg1)

Private Member Functions

- void addPreset (const Preset &preset)
- QIcon makeIcon (const Preset &preset)
- void updatePresets ()
- void setPreset (const QString &newPreset)

Private Attributes

- QMap< QString, Preset > presets
- QString currentPreset
- Ui::GradientDialog * ui

12.19.1 Detailed Description

Definition at line 12 of file gradientdialog.h.

12.19.2 Constructor & Destructor Documentation

12.19.2.1 GradientDialog::GradientDialog (QWidget * *parent* = 0) [explicit]

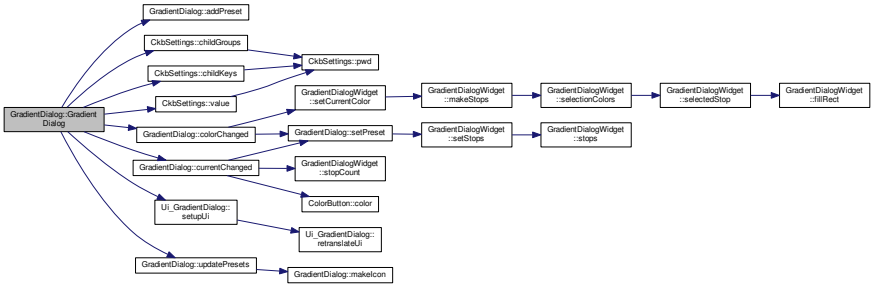
Definition at line 11 of file gradientdialog.cpp.

References addPreset(), CkbSettings::childGroups(), CkbSettings::childKeys(), colorChanged(), currentChanged(), GradientDialog::Preset::gradient, prefsPath, presets, Ui_GradientDialog::setupUi(), Ui_GradientDialog::stopColor, ui, updatePresets(), CkbSettings::value(), and Ui_GradientDialog::widget.

```
11                                                  :
12     QDialog(parent),
13     ui(new Ui::GradientDialog)
14 {
15     ui->setupUi(this);
16     connect(ui->widget, SIGNAL(currentChanged(QColor,bool,int)), this, SLOT(
       currentChanged(QColor,bool,int)));
17     connect(ui->stopColor, SIGNAL(colorChanged(QColor)), this, SLOT(
       colorChanged(QColor)));
18
19     // Add built-in presets
20     Preset blackAndWhite("B & W", true);
21     blackAndWhite.gradient.append(QGradientStop(0., QColor(0, 0, 0)));
22     blackAndWhite.gradient.append(QGradientStop(1., QColor(255, 255, 255)));
23     Preset fade("Fade", true);
24     fade.gradient.append(QGradientStop(0., QColor(255, 255, 255)));
25     fade.gradient.append(QGradientStop(1., QColor(255, 255, 255, 0)));
26     Preset rainbow("Rainbow", true);
27     rainbow.gradient.append(QGradientStop(0.00, QColor(255, 0, 0)));
28     rainbow.gradient.append(QGradientStop(0.17, QColor(255, 255, 0)));
29     rainbow.gradient.append(QGradientStop(0.33, QColor(0, 255, 0)));
30     rainbow.gradient.append(QGradientStop(0.50, QColor(0, 255, 255)));
31     rainbow.gradient.append(QGradientStop(0.67, QColor(0, 0, 255)));
32     rainbow.gradient.append(QGradientStop(0.83, QColor(255, 0, 255)));
33     rainbow.gradient.append(QGradientStop(1.00, QColor(255, 0, 0)));
34     addPreset(blackAndWhite);
35     addPreset(fade);
36     addPreset(rainbow);
37
38     // Load stored presets
39     CkbSettings settings(prefsPath);
40     foreach(const QString& name, settings.childGroups()){
41         QString pName = name.toLower();
42         if(presets.contains(pName))
43             continue;
44         SGroup group(settings, name);
45         // Create keys in a map to ensure correct order and no duplicates
46         QMap<int, QColor> colors;
47         foreach(const QString& position, settings.childKeys()){
48             int pos = position.toInt();
49             if(pos < 0 || pos > 100)
50                 continue;
51             colors[pos] = settings.value(position).value<QColor>();
52         }
53         if(colors.count() < 2)
54             continue;
55         Preset preset(name);
56         QMapIterator<int, QColor> i(colors);
57         while(i.hasNext()){
58             i.next();
59             preset.gradient.append(QGradientStop(i.key() / 100., i.value()));
60         }
61         presets[pName] = preset;
62     }
63
```

```
64     updatePresets();
65 }
```

Here is the call graph for this function:



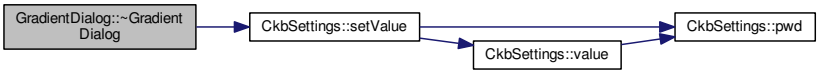
12.19.2.2 GradientDialog::~GradientDialog ( )

Definition at line 77 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, GradientDialog::Preset::gradient, GradientDialog::Preset::name, prefsPath, presets, CkbSettings::setValue(), and ui.

```
77                                {
78     // Save presets
79     CkbSettings settings(prefsPath, true);
80     QMapIterator<QString, Preset> i(presets);
81     while(i.hasNext()){
82         Preset preset = i.next().value();
83         if(!preset.builtIn){
84             SGroup group(settings, preset.name);
85             foreach(const QGradientStop& stop, preset.gradient)
86                 settings.setValue(QString::number((int)(stop.first * 100.)), stop.second);
87         }
88     }
89     delete ui;
90 }
```

Here is the call graph for this function:



12.19.3 Member Function Documentation

12.19.3.1 void GradientDialog::addPreset (const Preset & *preset*) [inline], [private]

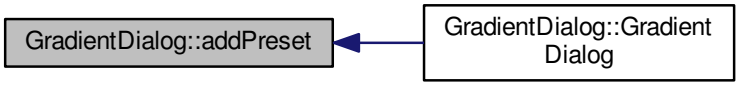
Definition at line 34 of file gradientdialog.h.

References GradientDialog::Preset::name, and presets.

Referenced by GradientDialog().

```
34 { presets[preset.name.toLower()] = preset; }
```

Here is the caller graph for this function:



12.19.3.2 void GradientDialog::colorChanged (QColor *color*) [private],[slot]

Definition at line 234 of file gradientdialog.cpp.

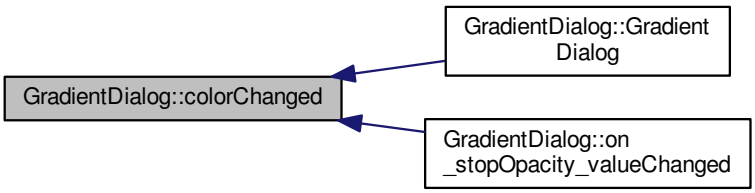
References GradientDialogWidget::setCurrentColor(), setPreset(), Ui_GradientDialog::stopOpacity, ui, and Ui_-GradientDialog::widget.

Referenced by GradientDialog(), and on_stopOpacity_valueChanged().

```
234                                          {
235     color.setAlphaF(ui->stopOpacity->value() / 100.f);
236     ui->widget->setCurrentColor(color);
237     setPreset("");
238 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.19.3.3 void GradientDialog::currentChanged (QColor *color,* bool *spontaneous,* int *position*) [private],[slot]

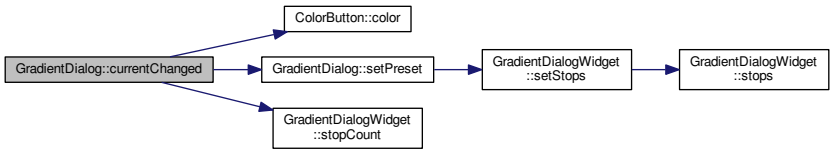
Definition at line 128 of file gradientdialog.cpp.

References ColorButton::color(), setPreset(), Ui_GradientDialog::stopBox, Ui_GradientDialog::stopColor, GradientDialogWidget::stopCount(), Ui_GradientDialog::stopOpacity, Ui_GradientDialog::stopPos, ui, and Ui_GradientDialog::widget.

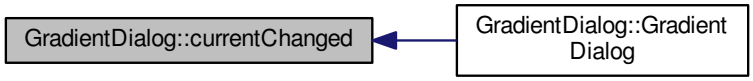
Referenced by GradientDialog().

```
128                                                                                 {
129     int pCount = ui->widget->stopCount();
130     if(pCount == 1)
131         ui->stopBox->setTitle("1 point");
132     else
133         ui->stopBox->setTitle(QString("%1 points").arg(pCount));
134     if(!color.isValid()){
135         ui->stopBox->setEnabled(false);
136         ui->stopPos->setValue(0);
137         ui->stopColor->color(QColor(0, 0, 0));
138         ui->stopOpacity->setValue(0);
139     } else {
140         ui->stopBox->setEnabled(true);
141         ui->stopPos->setValue(position);
142         ui->stopColor->color(color);
143         ui->stopOpacity->setValue(round(color.alphaF() * 100.f));
144     }
145     // Un-set current preset
146     if(spontaneous)
147         setPreset("");
148     setFocus();
149 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.19.3.4 QGradientStops GradientDialog::getGradient (const QGradientStops & *prevGradient*)

Definition at line 67 of file gradientdialog.cpp.

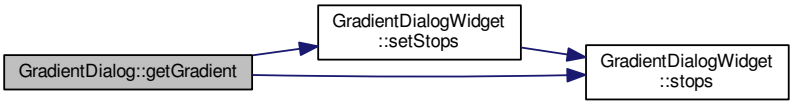
References Ui_GradientDialog::presetDelete, Ui_GradientDialog::presetSave, GradientDialogWidget::setStops(), GradientDialogWidget::stops(), ui, and Ui_GradientDialog::widget.

Referenced by GradientButton::pickGradient().

```
67                                                                                  {
68     ui->presetDelete->setEnabled(false);
69     ui->presetSave->setEnabled(true);
70     ui->widget->setStops(prevGradient);
```

```
71     exec();
72     if(result() != QDialog::Accepted)
73         return prevGradient;
74     return ui->widget->stops();
75 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.19.3.5 QIcon GradientDialog::makeIcon (const Preset & *preset*) [private]

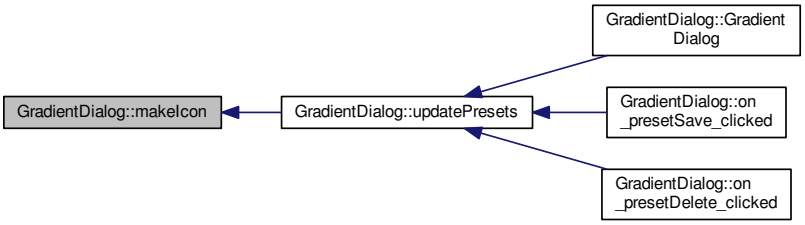
Definition at line 92 of file gradientdialog.cpp.

References GradientDialog::Preset::gradient, x, and y.

Referenced by updatePresets().

```
92                                                  {
93     // Paint gradient into a square
94     int w = 60, h = 60;
95     QImage image(w, h, QImage::Format_RGB888);
96     QPainter painter(&image);
97     // Draw background
98     for(int x = 0; x < w; x += 30){
99         for(int y = 0; y < h; y += 30){
100             painter.fillRect(x, y, 15, 15, QColor(255, 255, 255));
101             painter.fillRect(x + 15, y, 15, 15, QColor(192, 192, 192));
102             painter.fillRect(x, y + 15, 15, 15, QColor(192, 192, 192));
103             painter.fillRect(x + 15, y + 15, 15, 15, QColor(255, 255, 255));
104         }
105     }
106     // Draw gradient
107     painter.setPen(QColor(0, 0, 0));
108     QLinearGradient gradient(1., 1., w - 1., h - 1.);
109     gradient.setStops(preset.gradient);
110     painter.setBrush(gradient);
111     painter.drawRect(0, 0, w - 1, h - 1);
112     return QPixmap::fromImage(image);
113 }
```

Here is the caller graph for this function:



12.19.3.6 void GradientDialog::on_presetDelete_clicked ( ) [private], [slot]

Definition at line 211 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, current, currentPreset, GradientDialog::Preset::name, Ui_GradientDialog::presetDelete, presets, Ui_GradientDialog::presetSave, ui, and updatePresets().

```
211                                          {
212     Preset current = presets.value(currentPreset);
213     if(current.name == "" || current.builtIn)
214         return;
215     if(QMessageBox::warning(this, "Warning", QString("Delete preset \"%1\"?").arg(current.name),
      QMessageBox::StandardButtons(QMessageBox::Yes | QMessageBox::No)) != QMessageBox::Yes){
216         setFocus();
217         return;
218     }
219     presets.remove(currentPreset);
220     currentPreset = "";
221     ui->presetDelete->setEnabled(false);
222     ui->presetSave->setEnabled(true);
223     updatePresets();
224     setFocus();
225 }
```

Here is the call graph for this function:

12.19.3.7 void GradientDialog::on_presetList_currentItemChanged (QListWidgetItem * *current*, QListWidgetItem * *previous*) [private], [slot]

Definition at line 174 of file gradientdialog.cpp.

References setPreset().

```
174                                                                                                 {
175     setPreset(current ? current->text().toLower() : "");
176 }
```

Here is the call graph for this function:

12.19.3.8 void GradientDialog::on_presetName_textEdited (const QString & *arg1*) [private], [slot]

Definition at line 178 of file gradientdialog.cpp.

References currentPreset, presets, and setPreset().

```
178                                                                 {
179     if(currentPreset != "" && arg1.trimmed() != presets[
      currentPreset].name)
180         setPreset("");
181 }
```

Here is the call graph for this function:

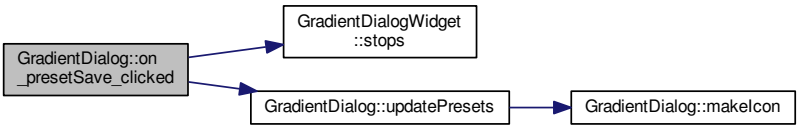
12.19.3.9 void GradientDialog::on_presetSave_clicked ( ) [private], [slot]

Definition at line 183 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, currentPreset, GradientDialog::Preset::name, Ui_GradientDialog::presetDelete, Ui_GradientDialog::presetName, presets, Ui_GradientDialog::presetSave, GradientDialogWidget::stops(), ui, updatePresets(), and Ui_GradientDialog::widget.

```
183                                         {
184     if(currentPreset != "")
185         return;
186     QString name = ui->presetName->text().trimmed();
187     if(name == "")
188         ui->presetName->setText(name = "Custom");
189     QString pName = name.toLower();
190     // Make sure not to overwrite a built-in preset
191     Preset previous = presets[pName];
192     if(previous.builtIn){
193         QMessageBox::warning(this, "Error", "Can't overwrite a built-in preset. Please choose a different
       name.");
194         setFocus();
195         return;
196     } else if(previous.name != ""){
197         // Warn when overwriting an existing preset
198         if(QMessageBox::warning(this, "Warning", QString("Preset \"%1\" already exists. Replace?").arg(name
      ), QMessageBox::StandardButtons(QMessageBox::Save | QMessageBox::Cancel)) != QMessageBox::Save){
199             setFocus();
200             return;
201         }
202     }
203     presets[pName] = Preset(name, ui->widget->stops());
204     currentPreset = pName;
205     ui->presetDelete->setEnabled(true);
206     ui->presetSave->setEnabled(false);
207     updatePresets();
208     setFocus();
209 }
```

Here is the call graph for this function:



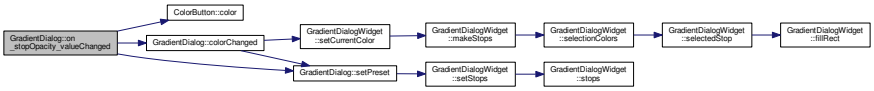
12.19.3.10 void GradientDialog::on_stopOpacity_valueChanged (int *arg1*) [private], [slot]

Definition at line 240 of file gradientdialog.cpp.

References ColorButton::color(), colorChanged(), setPreset(), Ui_GradientDialog::stopColor, and ui.

```
240                                                 {
241     colorChanged(ui->stopColor->color());
242     setPreset("");
243 }
```

Here is the call graph for this function:



12.19.3.11 void GradientDialog::on_stopPos_valueChanged (int *arg1*) [private], [slot]

Definition at line 227 of file gradientdialog.cpp.

References GradientDialogWidget::moveCurrent(), setPreset(), Ui_GradientDialog::stopPos, ui, and Ui_GradientDialog::widget.

```
227                                                 {
228     int res = ui->widget->moveCurrent(arg1);
229     if(res != arg1)
230         ui->stopPos->setValue(res);
231     setPreset("");
232 }
```

Here is the call graph for this function:

12.19.3.12 void GradientDialog::setPreset (const QString & *newPreset*) [private]

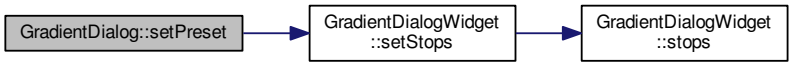
Definition at line 151 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, current, currentPreset, GradientDialog::Preset::gradient, GradientDialog::Preset::name, Ui_GradientDialog::presetDelete, Ui_GradientDialog::presetList, Ui_GradientDialog::presetName, presets, Ui_GradientDialog::presetSave, GradientDialogWidget::setStops(), ui, and Ui_GradientDialog::widget.

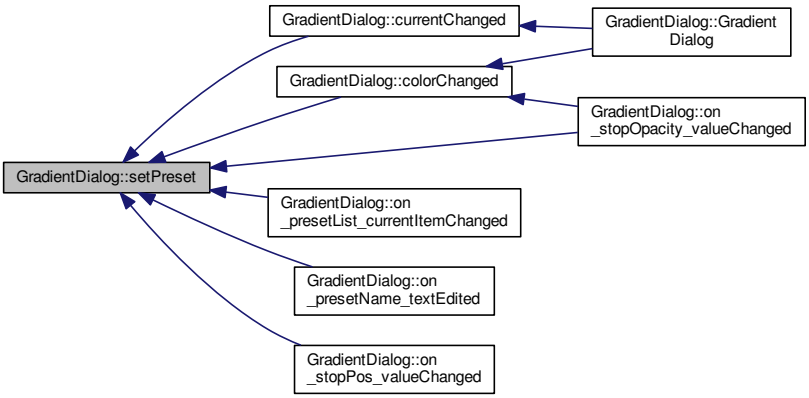
Referenced by colorChanged(), currentChanged(), on_presetList_currentItemChanged(), on_presetName_textEdited(), on_stopOpacity_valueChanged(), and on_stopPos_valueChanged().

```
151                                                 {
152     if(currentPreset == newPreset)
153         return;
154     if(newPreset == ""){
155         ui->presetList->clearSelection();
156         ui->presetList->setCurrentItem(0);
157         Preset current = presets.value(currentPreset);
158         QString curName = ui->presetName->text().trimmed();
159         if(curName == "" || (curName == current.name && current.builtIn))
160             ui->presetName->setText("Custom");
161         ui->presetDelete->setEnabled(false);
162         ui->presetSave->setEnabled(true);
163     } else {
164         Preset current = presets.value(newPreset);
165         ui->presetName->setText(current.name);
166         ui->presetDelete->setEnabled(!current.builtIn);
167         ui->presetSave->setEnabled(false);
168         ui->widget->setStops(current.gradient);
169     }
170     currentPreset = newPreset;
171     setFocus();
172 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.19.3.13 void GradientDialog::updatePresets ( ) [private]

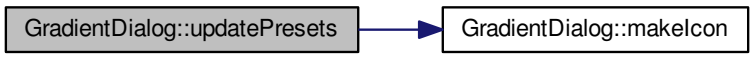
Definition at line 115 of file gradientdialog.cpp.

References currentPreset, makeIcon(), GradientDialog::Preset::name, Ui_GradientDialog::presetList, presets, and ui.

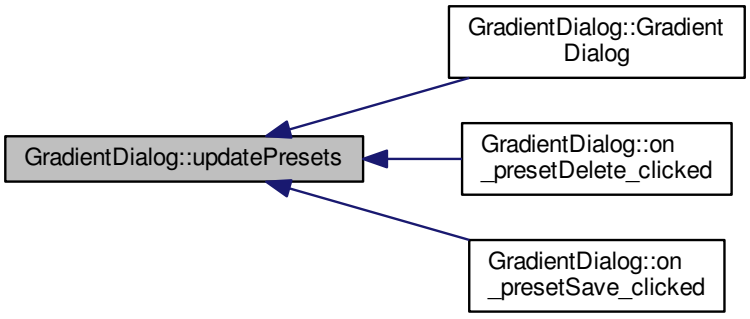
Referenced by GradientDialog(), on_presetDelete_clicked(), and on_presetSave_clicked().

```
115                                {
116     ui->presetList->clear();
117     QMapIterator<QString, Preset> i(presets);
118     while(i.hasNext()){
119         i.next();
120         const Preset& preset = i.value();
121         QListWidgetItem* item = new QListWidgetItem(makeIcon(preset), preset.name,
    ui->presetList);
122         if(currentPreset == i.key())
123             item->setSelected(true);
124         ui->presetList->addItem(item);
125     }
126 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.19.4 Field Documentation

12.19.4.1 QString GradientDialog::currentPreset [private]

Definition at line 32 of file gradientdialog.h.

Referenced by on_presetDelete_clicked(), on_presetName_textEdited(), on_presetSave_clicked(), setPreset(), and updatePresets().

12.19.4.2 QMap<QString, Preset> GradientDialog::presets [private]

Definition at line 31 of file gradientdialog.h.

Referenced by addPreset(), GradientDialog(), on_presetDelete_clicked(), on_presetName_textEdited(), on_presetSave_clicked(), setPreset(), updatePresets(), and ~GradientDialog().

12.19.4.3 Ui::GradientDialog* GradientDialog::ui [private]

Definition at line 39 of file gradientdialog.h.

Referenced by colorChanged(), currentChanged(), getGradient(), GradientDialog(), on_presetDelete_clicked(), on_presetSave_clicked(), on_stopOpacity_valueChanged(), on_stopPos_valueChanged(), setPreset(), updatePresets(), and ~GradientDialog().

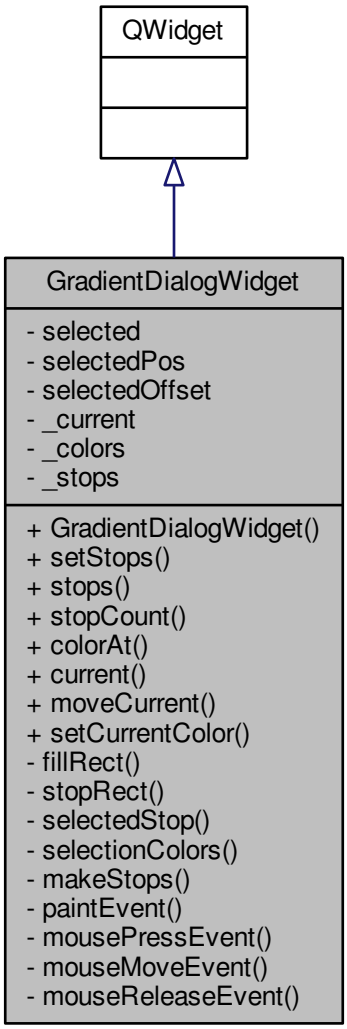
The documentation for this class was generated from the following files:

- src/ckb/gradientdialog.h
- src/ckb/gradientdialog.cpp

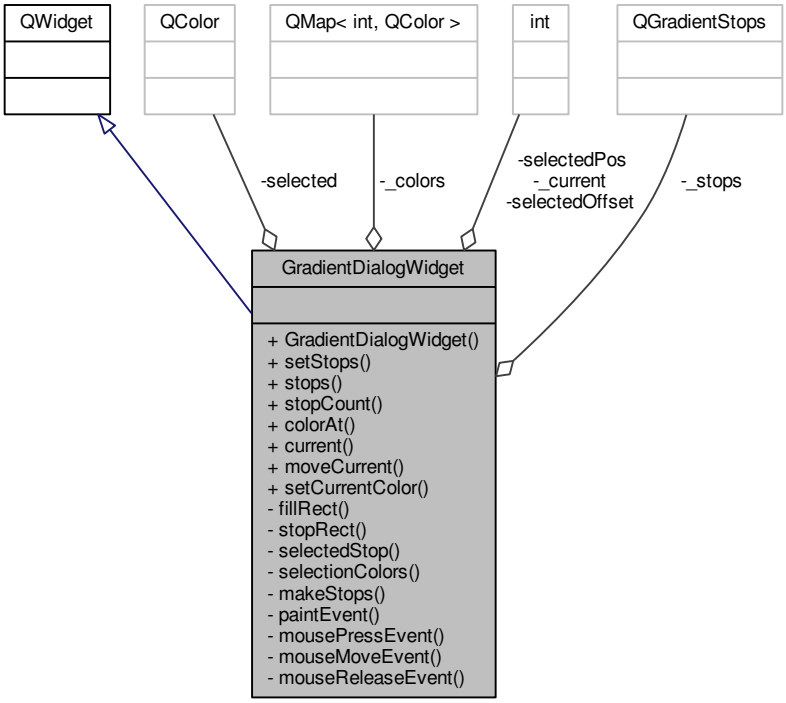
12.20 GradientDialogWidget Class Reference

`#include <src/ckb/gradientdialogwidget.h>`

Inheritance diagram for GradientDialogWidget:



Collaboration diagram for GradientDialogWidget:



Signals

- void currentChanged (QColor color, bool spontaneous, int position)

Public Member Functions

- GradientDialogWidget (QWidget ∗parent=0)
- void setStops (const QGradientStops &stops)
- QGradientStops stops ()
- int stopCount ()
- QColor colorAt (int position)
- int current ()
- int moveCurrent (int to)
- void setCurrentColor (const QColor &color)

Private Member Functions

- QRect fillRect ()
- QRect stopRect (int pos)
- int selectedStop ()
- QMap< int, QColor > selectionColors ()
- void makeStops ()
- void paintEvent (QPaintEvent ∗)
- void mousePressEvent (QMouseEvent ∗event)
- void mouseMoveEvent (QMouseEvent ∗event)
- void mouseReleaseEvent (QMouseEvent ∗event)

Private Attributes

- QColor selected
- int selectedPos
- int selectedOffset
- int _current
- QMap< int, QColor > _colors
- QGradientStops _stops

12.20.1 Detailed Description

Definition at line 8 of file gradientdialogwidget.h.

12.20.2 Constructor & Destructor Documentation

12.20.2.1 GradientDialogWidget::GradientDialogWidget (QWidget * *parent* = 0) [explicit]

Definition at line 6 of file gradientdialogwidget.cpp.

```
6                                                                   :
7     QWidget(parent), selectedPos(-1), _current(-1)
8 {
9     setMouseTracking(true);
10 }
```

12.20.3 Member Function Documentation

12.20.3.1 QColor GradientDialogWidget::colorAt (int *position*)

Definition at line 65 of file gradientdialogwidget.cpp.

References _stops.

Referenced by mousePressEvent().

```
65                                                {
66     // Find which stops the position is between
67     qreal pos = position / 100.;
68     QVectorIterator<QGradientStop> i(_stops);
69     QGradientStop previous = i.next();
70     while(i.hasNext()){
71         QGradientStop next = i.next();
72         if(next.first >= pos){
73             // Get color by linear interpolation. Premultiply the alpha value so that it returns the
       expected color
74             // (i.e. stops with zero opacity won't contribute to color)
75             QColor c1 = next.second, c2 = previous.second;
76             qreal distance = next.first - previous.first;
77             qreal dx = (pos - previous.first) / distance;
78             qreal a1 = c1.alphaF(), a2 = c2.alphaF();
79             qreal a3 = a1 * dx + a2 * (1.f - dx);
80             if(a3 == 0.){
81                 c2.setAlpha(0);
82                 return c2;
83             }
84             qreal r = (c1.redF() * a1 * dx + c2.redF() * a2 * (1.f - dx)) / a3;
85             qreal g = (c1.greenF() * a1 * dx + c2.greenF() * a2 * (1.f - dx)) / a3;
86             qreal b = (c1.blueF() * a1 * dx + c2.blueF() * a2 * (1.f - dx)) / a3;
87             return QColor::fromRgbF(r, g, b, a3);
88         }
89         previous = next;
90     }
91     return QColor();
92 }
```

Here is the caller graph for this function:

12.20.3.2 int GradientDialogWidget::current ( ) [inline]

Definition at line 20 of file gradientdialogwidget.h.

References _current.

```
20 { return _current; }
```

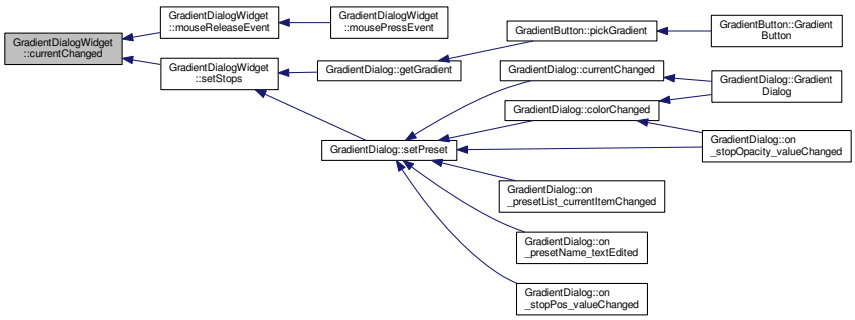
12.20.3.3 void GradientDialogWidget::currentChanged (QColor *color,* bool *spontaneous,* int *position*) [signal]

Definition at line 126 of file moc_gradientdialogwidget.cpp.

Referenced by mouseReleaseEvent(), and setStops().

```
127 {
128     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)), const_cast<void*>(
      reinterpret_cast<const void*>(&_t2)), const_cast<void*>(reinterpret_cast<const void*>(&_t3)) };
129     QMetaObject::activate(this, &staticMetaObject, 0, _a);
130 }
```

Here is the caller graph for this function:



12.20.3.4 QRect GradientDialogWidget::fillRect ( ) [private]

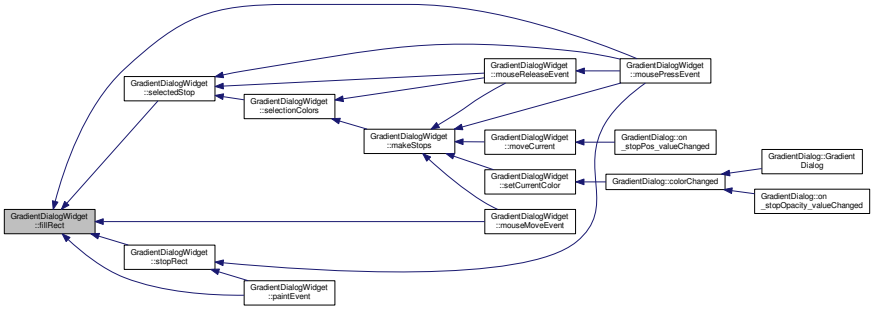
Definition at line 138 of file gradientdialogwidget.cpp.

References width, x, and y.

Referenced by mouseMoveEvent(), mousePressEvent(), paintEvent(), selectedStop(), and stopRect().

```
138                                     {
139     int x = 8, y = 1;
140     int w = (width() - x * 2) / 16 * 16, h = (height() - 24) / 16 * 16;
141     return QRect(x, y, w, h);
142 }
```

Here is the caller graph for this function:



12.20.3.5 void GradientDialogWidget::makeStops ( ) [private]

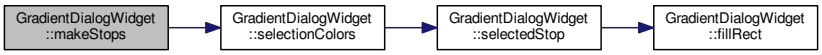
Definition at line 30 of file gradientdialogwidget.cpp.

References _stops, and selectionColors().

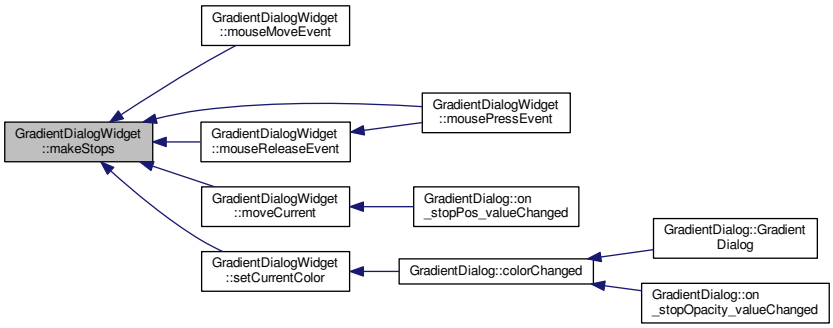
Referenced by mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), moveCurrent(), and setCurrentColor().

```
30                                     {
31     _stops.clear();
32     QMap<int, QColor> colors = selectionColors();
33     // If there's only a single color, put it at the beginning and end
34     if(colors.count() == 1){
35 #if QT_VERSION >= QT_VERSION_CHECK(5, 2, 0)
36         QColor color = colors.first();
37 #else
38         QColor color = colors.value(colors.keys().first());
39 #endif
40         _stops.append(QGradientStop(0., color));
41         _stops.append(QGradientStop(1., color));
42         update();
43         return;
44     }
45     // Add colors at 0 and 100 if needed
46 #if QT_VERSION >= QT_VERSION_CHECK(5, 2, 0)
47     if(!colors.contains(0))
48         colors[0] = colors.first();
49     if(!colors.contains(100))
50         colors[100] = colors.last();
51 #else
52     if(!colors.contains(0))
53         colors[0] = colors.value(colors.keys().first());
54     if(!colors.contains(100))
55         colors[100] = colors.value(colors.keys().last());
56 #endif
57     QMapIterator<int, QColor> i(colors);
58     while(i.hasNext()){
59         i.next();
60         _stops.append(QGradientStop(i.key() / 100., i.value()));
61     }
62     update();
63 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



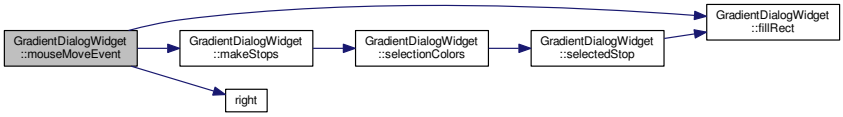
12.20.3.6 void GradientDialogWidget::mouseMoveEvent (QMouseEvent * *event*) [private]

Definition at line 255 of file gradientdialogwidget.cpp.

References _colors, fillRect(), left, makeStops(), right(), selected, selectedOffset, selectedPos, and top.

```
255                                                     {
256     if(!selected.isValid()){
257         if(event->y() >= height() - 24)
258             setCursor(QCursor(Qt::PointingHandCursor));
259         else
260             setCursor(QCursor(Qt::ArrowCursor));
261         return;
262     }
263     // Move selected point (if any)
264     setCursor(QCursor(Qt::ClosedHandCursor));
265     selectedPos = event->x() + selectedOffset;
266     // Allow the point to be deleted as long as there is at least 1 other point
267     bool last = (_colors.count() == 0);
268     // Hug the left/right edges
269     QRect rect = fillRect();
270     int left = rect.left(), right = rect.right();
271     if(selectedPos < left && (last || selectedPos >= left - 30))
272         selectedPos = left;
273     else if(selectedPos > right && (last || selectedPos <=
    right + 30))
274         selectedPos = right;
275     // Remove point if it's too far above/below the widget
276     int top = -30, bottom = height() + 30;
277     if(!last && (event->y() < top || event->y() > bottom))
278         selectedPos = -1;
279     makeStops();
280 }
```

Here is the call graph for this function:



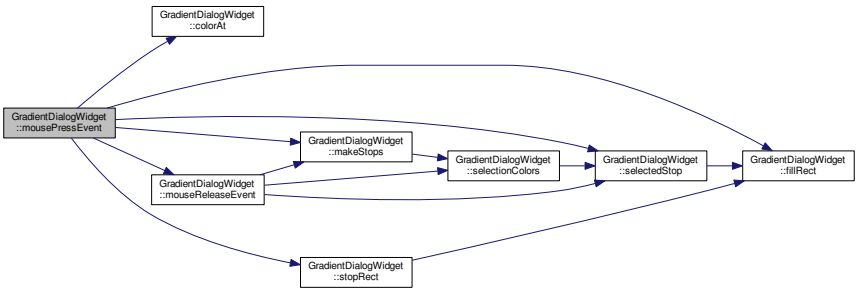
12.20.3.7 void GradientDialogWidget::mousePressEvent (QMouseEvent * *event*) [private]

Definition at line 222 of file gradientdialogwidget.cpp.

References _colors, colorAt(), fillRect(), makeStops(), mouseReleaseEvent(), selected, selectedOffset, selectedPos, selectedStop(), stopRect(), x, and y.

```
222                                                     {
223     int x = event->x(), y = event->y();
224     if(y < height() - 24)
225         return;
226     QMapIterator<int, QColor> i(_colors);
227     while(i.hasNext()){
228         i.next();
229         // If a point was selected, remove it from the color map and set it as the selected point
230         QRect stop = stopRect(i.key());
231         if(x >= stop.left() - 1 && x <= stop.right() + 1){
232             selected = i.value();
233             selectedPos = stop.x() + stop.width() / 2;
234             selectedOffset = selectedPos - x;
235             _colors.remove(i.key());
236             // On right click, delete the point
237             if(event->button() == Qt::RightButton && _colors.count() > 1){
238                 selectedPos = -1;
239                 mouseReleaseEvent(event);
240             } else
241                 update();
242             return;
243         }
244     }
245     // If nothing was selected, create a new point
246     QRect fill = fillRect();
247     if(x > fill.left() && x < fill.right() && event->button() != Qt::RightButton){
248         selectedPos = x;
249         selectedOffset = 0;
250         selected = colorAt(selectedStop());
251         makeStops();
252     }
253 }
```

Here is the call graph for this function:



12.20.3.8 void GradientDialogWidget::mouseReleaseEvent (QMouseEvent * *event*) [private]

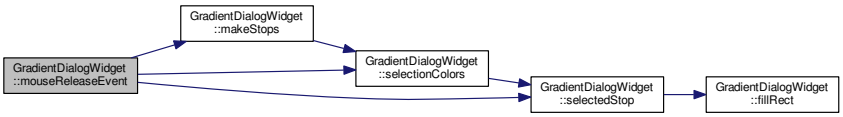
Definition at line 282 of file gradientdialogwidget.cpp.

References _colors, _current, currentChanged(), makeStops(), selected, selectedPos, selectedStop(), and selectionColors().

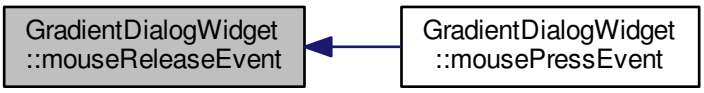
Referenced by mousePressEvent().

```
282                                                         {
283     if(selected.isValid()){
284         // Rejoin selection with gradient
285         _current = selectedStop();
286         _colors = selectionColors();
287         emit currentChanged(_colors.value(_current), true,
      _current);
288         selectedPos = -1;
289         selected = QColor();
290         makeStops();
291     }
292     if(event->y() >= height() - 24)
293         setCursor(QCursor(Qt::PointingHandCursor));
294     else
295         setCursor(QCursor(Qt::ArrowCursor));
296 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.20.3.9 int GradientDialogWidget::moveCurrent (int *to*)

Definition at line 94 of file gradientdialogwidget.cpp.

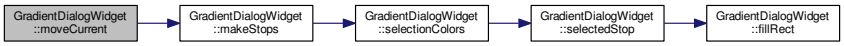
References _colors, _current, makeStops(), and selected.

Referenced by GradientDialog::on_stopPos_valueChanged().

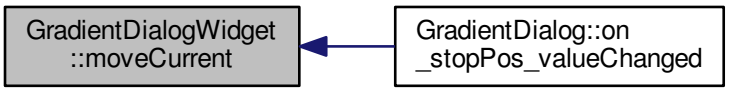
```
94                                                  {
95     if(_current < 0 || to == _current || selected.isValid())
96         return _current;
97     // If there's a point in the way, skip past it by continuing in the current direction
98     if(_current < to){
99         while(_colors.contains(to)){
100             to++;
101             if(to > 100)
```

```
102                 return _current;
103             }
104         } else {
105             while(_colors.contains(to)){
106                 to--;
107                 if(to < 0)
108                     return _current;
109             }
110         }
111         _colors[to] = _colors.take(_current);
112         _current = to;
113         makeStops();
114         return to;
115 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.20.3.10 void GradientDialogWidget::paintEvent (QPaintEvent *) [private]

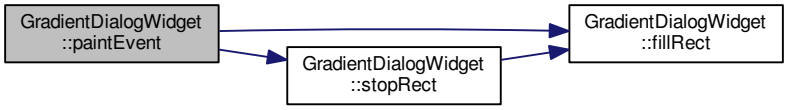
Definition at line 172 of file gradientdialogwidget.cpp.

References _colors, _current, _stops, fillRect(), selected, selectedPos, stopRect(), x, and y.

```
172                                                 {
173     QPainter painter(this);
174     QRect fill = fillRect();
175     int x = fill.x(), y = fill.y(), w = fill.width(), h = fill.height();
176 
177     // Draw border around gradient
178     painter.setPen(palette().color(QPalette::Dark));
179     painter.drawLine(x - 1, y - 1, x + w, y - 1);
180     painter.drawLine(x - 1, y - 1, x - 1, y + h);
181     painter.setPen(palette().color(QPalette::Light));
182     painter.drawLine(x - 1, y + h, x + w, y + h);
183     painter.drawLine(x + w, y - 1, x + w, y + h);
184     // Draw background grid
185     for(int i = 0; i < w; i += 16){
186         for(int iy = 0; iy < h; iy += 16){
187             if(i % 32 != iy % 32)
188                 painter.fillRect(x + i, y + iy, 16, 16, QColor(192, 192, 192));
189             else
190                 painter.fillRect(x + i, y + iy, 16, 16, QColor(255, 255, 255));
191         }
192     }
193     // Draw gradient
194     painter.setRenderHint(QPainter::Antialiasing);
195     QLinearGradient gradient(x, 0, x + w, 0);
196     gradient.setStops(_stops);
197     painter.fillRect(fill, QBrush(gradient));
198 
199     // Draw stops
```

```
200     QMapIterator<int, QColor> i(_colors);
201     painter.setPen(palette().color(QPalette::Shadow));
202     while(i.hasNext()){
203         i.next();
204         painter.setBrush(QBrush(i.value()));
205         int index = i.key();
206         if(index == _current){
207             // Highlight currently-edited stop
208             painter.setPen(QPen(palette().color(QPalette::Highlight), 2));
209             painter.drawRoundedRect(stopRect(index), 4, 4);
210             painter.setPen(palette().color(QPalette::Shadow));
211         } else
212             painter.drawRoundedRect(stopRect(index), 4, 4);
213     }
214     // Draw selection (if any)
215     if(selected.isValid() && selectedPos >= x && selectedPos <= x + w){
216         painter.setPen(QPen(palette().color(QPalette::Highlight), 2.5));
217         painter.setBrush(selected);
218         painter.drawRoundedRect(selectedPos - 6, height() - 18 - 2, 12, 18, 4, 4);
219     }
220 }
```

Here is the call graph for this function:



12.20.3.11 int GradientDialogWidget::selectedStop ( ) [private]

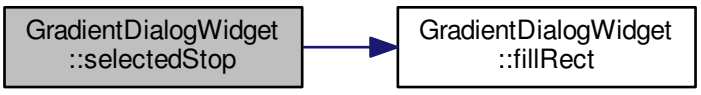
Definition at line 152 of file gradientdialogwidget.cpp.

References fillRect(), and selectedPos.

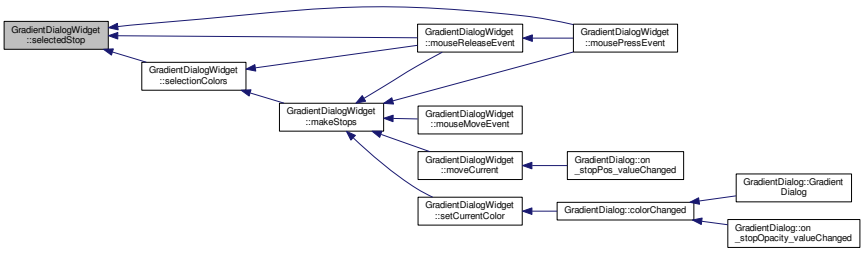
Referenced by mousePressEvent(), mouseReleaseEvent(), and selectionColors().

```
152                                     {
153     QRect rect = fillRect();
154     int res = round((selectedPos - rect.x()) * 100. / rect.width());
155     if(res < 0 || res > 100)
156         return -1;
157     return res;
158 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.20.3.12 QMap< int, QColor > GradientDialogWidget::selectionColors ( ) [private]

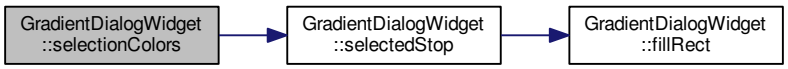
Definition at line 160 of file gradientdialogwidget.cpp.

References _colors, selected, and selectedStop().

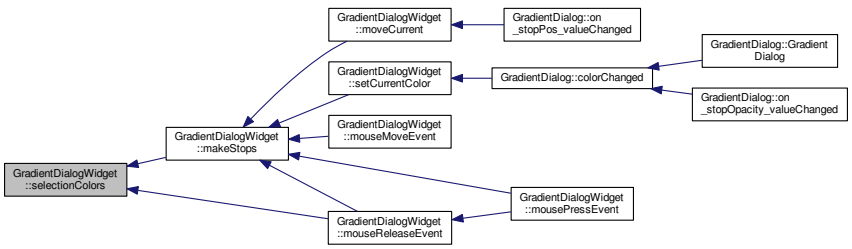
Referenced by makeStops(), and mouseReleaseEvent().

```
160                                                     {
161     if(!selected.isValid())
162         return _colors;
163     // Determine selected stop position
164     int selPos = selectedStop();
165     // Re-add if in range
166     QMap<int, QColor> res = _colors;
167     if(selPos >= 0 && selPos <= 100)
168         res[selPos] = selected;
169     return res;
170 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.20.3.13 void GradientDialogWidget::setCurrentColor (const QColor & *color*)

Definition at line 117 of file gradientdialogwidget.cpp.

References _colors, _current, makeStops(), and selected.

Referenced by GradientDialog::colorChanged().

```
117                                                          {
118     if(_current < 0 || selected.isValid())
119         return;
120     QRgb rgb = _colors.value(_current).rgb();
121     _colors[_current] = color;
122     // If any points follow with the same color but different opacity, change them to match
123     QMutableMapIterator<int, QColor> i(_colors);
124     while(i.hasNext()){
125         i.next();
126         if(i.key() <= _current)
127             continue;
128         QColor& value = i.value();
129         if(value.rgb() != rgb)
130             break;
131         int alpha = value.alpha();
132         value = color;
133         value.setAlpha(alpha);
134     }
135     makeStops();
136 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.20.3.14 void GradientDialogWidget::setStops (const QGradientStops & *stops*)

Definition at line 12 of file gradientdialogwidget.cpp.

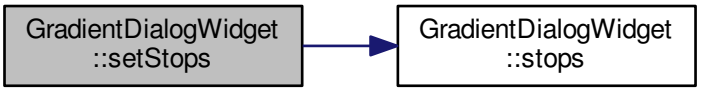
References _colors, _current, _stops, currentChanged(), and stops().

Referenced by GradientDialog::getGradient(), and GradientDialog::setPreset().

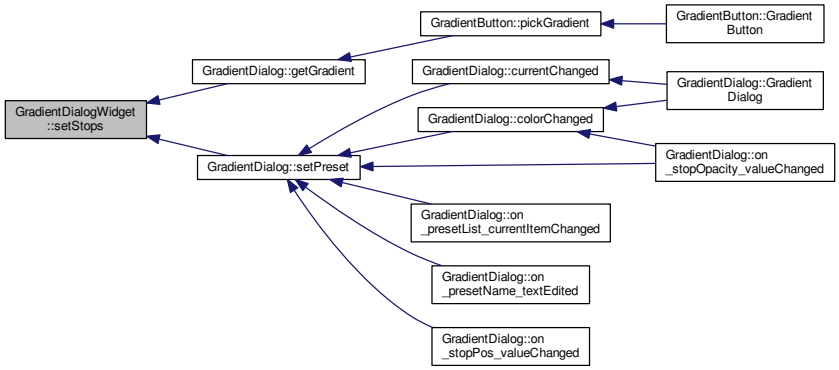
```
12                                                          {
13     _stops = stops;
14     _colors.clear();
15     foreach(const QGradientStop& stop, stops){
16         double pos = stop.first;
17         if(pos < 0. || pos > 1.)
18             continue;
19         _colors[round(pos * 100.)] = stop.second;
20     }
21     _current = 0;
```

```
22 #if QT_VERSION >= QT_VERSION_CHECK(5, 2, 0)
23     emit currentChanged(_colors.first(), false, 0);
24 #else
25     emit currentChanged(_colors.value(_colors.keys().first()), false, 0);
26 #endif
27     update();
28 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.20.3.15 int GradientDialogWidget::stopCount ( ) [inline]

Definition at line 16 of file gradientdialogwidget.h.

References _stops.

Referenced by GradientDialog::currentChanged().

```
16 { return _stops.count(); }
```

Here is the caller graph for this function:

12.20.3.16 QRect GradientDialogWidget::stopRect (int *pos*) [private]

Definition at line 144 of file gradientdialogwidget.cpp.

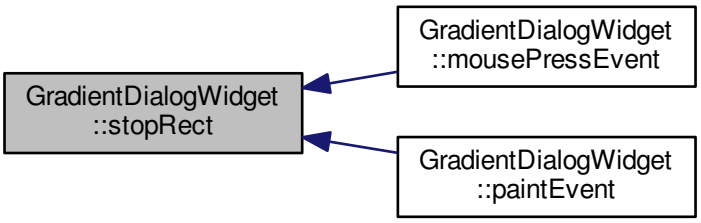
References fillRect().

Referenced by mousePressEvent(), and paintEvent().

```
144                                             {
145     QRect fill = fillRect();
146     int sx = round(pos / 100. * fill.width()) + fill.x() - 6;
147     int sw = 12, sh = 18;
148     int sy = height() - sh - 2;
149     return QRect(sx, sy, sw, sh);
150 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.20.3.17 QGradientStops GradientDialogWidget::stops ( ) [inline]

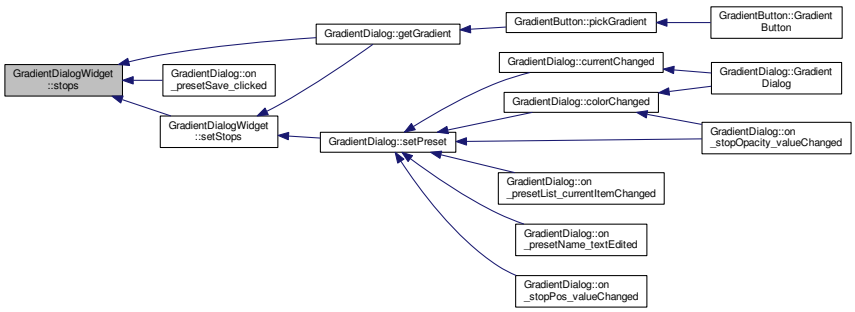
Definition at line 15 of file gradientdialogwidget.h.

References _stops.

Referenced by GradientDialog::getGradient(), GradientDialog::on_presetSave_clicked(), and setStops().

```
15 { return _stops; }
```

Here is the caller graph for this function:



12.20.4 Field Documentation

12.20.4.1 QMap<int, QColor> GradientDialogWidget::_colors [private]

Definition at line 31 of file gradientdialogwidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), moveCurrent(), paintEvent(), selectionColors(), setCurrentColor(), and setStops().

12.20.4.2 int GradientDialogWidget::_current [private]

Definition at line 30 of file gradientdialogwidget.h.

Referenced by current(), mouseReleaseEvent(), moveCurrent(), paintEvent(), setCurrentColor(), and setStops().

12.20.4.3 QGradientStops GradientDialogWidget::_stops [private]

Definition at line 32 of file gradientdialogwidget.h.

Referenced by colorAt(), makeStops(), paintEvent(), setStops(), stopCount(), and stops().

12.20.4.4 QColor GradientDialogWidget::selected [private]

Definition at line 28 of file gradientdialogwidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), moveCurrent(), paintEvent(), selectionColors(), and setCurrentColor().

12.20.4.5 int GradientDialogWidget::selectedOffset [private]

Definition at line 29 of file gradientdialogwidget.h.

Referenced by mouseMoveEvent(), and mousePressEvent().

12.20.4.6 int GradientDialogWidget::selectedPos [private]

Definition at line 29 of file gradientdialogwidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), and selectedStop().

The documentation for this class was generated from the following files:

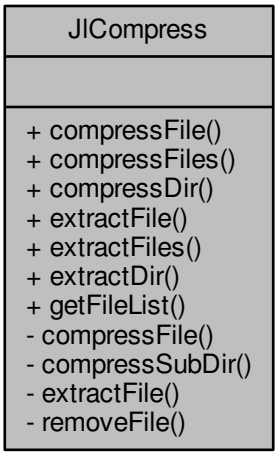
- src/ckb/gradientdialogwidget.h
- src/ckb/gradientdialogwidget.cpp
- src/ckb/release/moc_gradientdialogwidget.cpp

12.21 JICompress Class Reference

Utility class for typical operations.

`#include <src/ckb/quazip/JlCompress.h>`

Collaboration diagram for JICompress:



Static Public Member Functions

- static bool compressFile (QString fileCompressed, QString file)

  Compress a single file.
- static bool compressFiles (QString fileCompressed, QStringList files)

  Compress a list of files.
- static bool compressDir (QString fileCompressed, QString dir=QString(), bool recursive=true)

  Compress a whole directory.
- static QString extractFile (QString fileCompressed, QString fileName, QString fileDest=QString())

  Extract a single file.
- static QStringList extractFiles (QString fileCompressed, QStringList files, QString dir=QString())

  Extract a list of files.
- static QStringList extractDir (QString fileCompressed, QString dir=QString())

  Extract a whole archive.
- static QStringList getFileList (QString fileCompressed)

  Get the file list.

Static Private Member Functions

- static bool compressFile (QuaZip *zip, QString fileName, QString fileDest)

  Compress a single file.
- static bool compressSubDir (QuaZip *parentZip, QString dir, QString parentDir, bool recursive=true)

  Compress a subdirectory.
- static bool extractFile (QuaZip *zip, QString fileName, QString fileDest)

  Extract a single file.
- static bool removeFile (QStringList listFile)

  Remove some files.

12.21.1 Detailed Description

This class contains a number of useful static functions to perform simple operations, such as mass ZIP packing or extraction.

Definition at line 42 of file JlCompress.h.

12.21.2 Member Function Documentation

12.21.2.1 bool JlCompress::compressDir (QString *fileCompressed*, QString *dir* = `QString()`, bool *recursive* = `true`) [static]

OK Comprime la cartella dir nel file fileCompressed, se recursive e true allora comprime anche le sotto cartelle.

Parameters

fileCompressed	The name of the archive.
dir	The directory to compress.
recursive	Whether to pack the subdirectories as well, or just regular files.

Returns

true if success, false otherwise.

Se la funzione fallisce restituisce false e cancella il file che si e tentato di creare.

La funzione fallisce se:

- non si riesce ad aprire l'oggetto zip;
- la compressione di un file fallisce;
- non si riesce a chiudere l'oggetto zip;

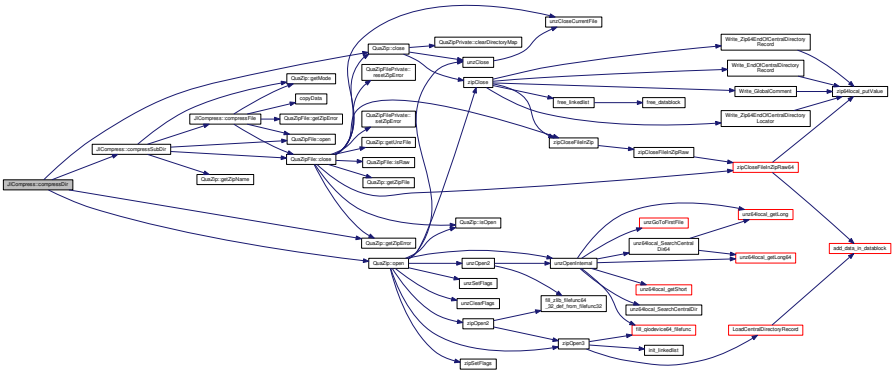
Definition at line 338 of file JlCompress.cpp.

References QuaZip::close(), compressSubDir(), QuaZip::getZipError(), QuaZip::mdCreate, and QuaZip::open().

```
338                                                                                 {
339     // Creo lo zip
340     QuaZip zip(fileCompressed);
341     QDir().mkpath(QFileInfo(fileCompressed).absolutePath());
342     if(!zip.open(QuaZip::mdCreate)) {
343         QFile::remove(fileCompressed);
344         return false;
345     }
346
347     // Aggiungo i file e le sotto cartelle
348     if (!compressSubDir(&zip,dir,dir,recursive)) {
349         QFile::remove(fileCompressed);
350         return false;
```

```
351     }
352
353     // Chiudo il file zip
354     zip.close();
355     if(zip.getZipError()!=0) {
356         QFile::remove(fileCompressed);
357         return false;
358     }
359
360     return true;
361 }
```

Here is the call graph for this function:



12.21.2.2 bool JICompress::compressFile (QuaZip * *zip,* QString *fileName,* QString *fileDest*) [static], [private]

OK Comprime il file fileName, nell'oggetto zip, con il nome fileDest.

Parameters

zip	Opened zip to compress the file to.
fileName	The full path to the source file.
fileDest	The full name of the file inside the archive.

Returns

true if success, false otherwise.

La funzione fallisce se:

- zip==NULL;
- l'oggetto zip e stato aperto in una modalita non compatibile con l'aggiunta di file;
- non e possibile aprire il file d'origine;
- non e possibile creare il file all'interno dell'oggetto zip;
- si e rilevato un errore nella copia dei dati;
- non e stato possibile chiudere il file all'interno dell'oggetto zip;

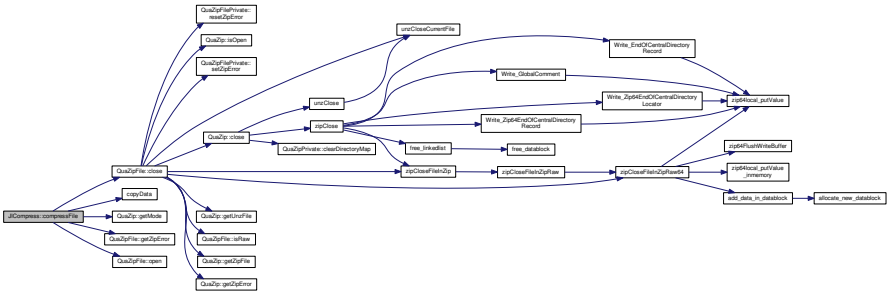
Definition at line 53 of file JICompress.cpp.

References QuaZipFile::close(), copyData(), QuaZip::getMode(), QuaZipFile::getZipError(), QuaZip::mdAdd, QuaZip::mdAppend, QuaZip::mdCreate, QuaZipFile::open(), and UNZ_OK.

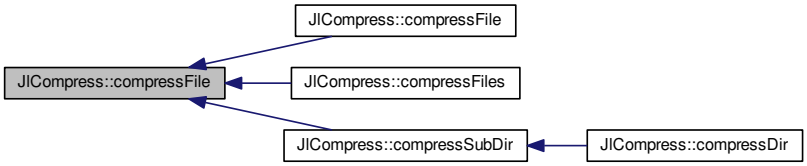
Referenced by compressFile(), compressFiles(), and compressSubDir().

```
53                                                                                  {
54     // zip: oggetto dove aggiungere il file
55     // fileName: nome del file reale
56     // fileDest: nome del file all'interno del file compresso
57
58     // Controllo l'apertura dello zip
59     if (!zip) return false;
60     if (zip->getMode()!=QuaZip::mdCreate &&
61         zip->getMode()!=QuaZip::mdAppend &&
62         zip->getMode()!=QuaZip::mdAdd) return false;
63
64     // Apro il file originale
65     QFile inFile;
66     inFile.setFileName(fileName);
67     if(!inFile.open(QIODevice::ReadOnly)) return false;
68
69     // Apro il file risulato
70     QuaZipFile outFile(zip);
71     if(!outFile.open(QIODevice::WriteOnly, QuaZipNewInfo(fileDest, inFile.fileName()))) return
       false;
72
73     // Copio i dati
74     if (!copyData(inFile, outFile) || outFile.getZipError()!=UNZ_OK) {
75         return false;
76     }
77
78     // Chiudo i file
79     outFile.close();
80     if (outFile.getZipError()!=UNZ_OK) return false;
81     inFile.close();
82
83     return true;
84 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.21.2.3 bool JICompress::compressFile (QString *fileCompressed,* QString *file*) [static]

OK Comprime il file fileName nel file fileCompressed.

Parameters

fileCompressed	The name of the archive.
file	The file to compress.

Returns

true if success, false otherwise.

Se la funzione fallisce restituisce false e cancella il file che si e tentato di creare.

La funzione fallisce se:

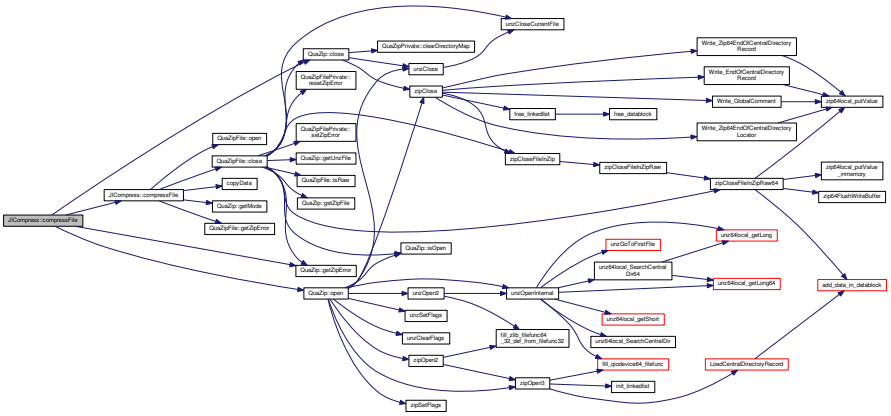
- non si riesce ad aprire l'oggetto zip;
- la compressione del file fallisce;
- non si riesce a chiudere l'oggetto zip;

Definition at line 263 of file JICompress.cpp.

References QuaZip::close(), compressFile(), QuaZip::getZipError(), QuaZip::mdCreate, and QuaZip::open().

```
263                                                                 {
264     // Creo lo zip
265     QuaZip zip(fileCompressed);
266     QDir().mkpath(QFileInfo(fileCompressed).absolutePath());
267     if(!zip.open(QuaZip::mdCreate)) {
268         QFile::remove(fileCompressed);
269         return false;
270     }
271
272     // Aggiungo il file
273     if (!compressFile(&zip,file,QFileInfo(file).fileName())) {
274         QFile::remove(fileCompressed);
275         return false;
276     }
277
278     // Chiudo il file zip
279     zip.close();
280     if(zip.getZipError()!=0) {
281         QFile::remove(fileCompressed);
282         return false;
283     }
284
285     return true;
286 }
```

Here is the call graph for this function:



12.21.2.4 bool JlCompress::compressFiles (QString *fileCompressed*, QStringList *files*) [static]

OK Comprime i file specificati in files nel file fileCompressed.

Parameters

fileCompressed	The name of the archive.
files	The file list to compress.

Returns

true if success, false otherwise.

Se la funzione fallisce restituisce false e cancella il file che si e tentato di creare.

La funzione fallisce se:

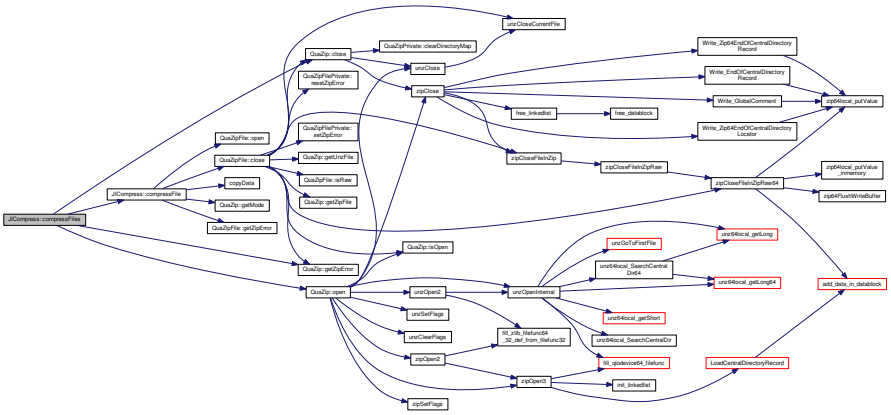
- non si riesce ad aprire l'oggetto zip;
- la compressione di un file fallisce;
- non si riesce a chiudere l'oggetto zip;

Definition at line 298 of file JICompress.cpp.

References QuaZip::close(), compressFile(), QuaZip::getZipError(), QuaZip::mdCreate, and QuaZip::open().

```
298                                                                    {
299     // Creo lo zip
300     QuaZip zip(fileCompressed);
301     QDir().mkpath(QFileInfo(fileCompressed).absolutePath());
302     if(!zip.open(QuaZip::mdCreate)) {
303         QFile::remove(fileCompressed);
304         return false;
305     }
306
307     // Comprimo i file
308     QFileInfo info;
309     Q_FOREACH (QString file, files) {
310         info.setFile(file);
311         if (!info.exists() || !compressFile(&zip,file,info.fileName())) {
312             QFile::remove(fileCompressed);
313             return false;
314         }
315     }
316
317     // Chiudo il file zip
318     zip.close();
319     if(zip.getZipError()!=0) {
320         QFile::remove(fileCompressed);
321         return false;
322     }
323
324     return true;
325 }
```

Here is the call graph for this function:



12.21.2.5 bool JICompress::compressSubDir (QuaZip * *zip,* QString *dir,* QString *origDir,* bool *recursive* = `true`) [static], [private]

OK Comprime la cartella dir nel file fileCompressed, se recursive e true allora comprime anche le sotto cartelle.

Parameters

parentZip	Opened zip containing the parent directory.
dir	The full path to the directory to pack.
parentDir	The full path to the directory corresponding to the root of the ZIP.
recursive	Whether to pack sub-directories as well or only files.

Returns

true if success, false otherwise.

I nomi dei file preceduti dal path creato togliendo il pat della cartella origDir al path della cartella dir. Se la funzione fallisce restituisce false e cancella il file che si e tentato di creare.

La funzione fallisce se:

- zip==NULL;
- l'oggetto zip e stato aperto in una modalita non compatibile con l'aggiunta di file;
- la cartella dir non esiste;
- la compressione di una sotto cartella fallisce (1);
- la compressione di un file fallisce; (1) La funzione si richiama in maniera ricorsiva per comprimere le sotto cartelle dunque gli errori di compressione di una sotto cartella sono gli stessi di questa funzione.

Definition at line 103 of file JICompress.cpp.

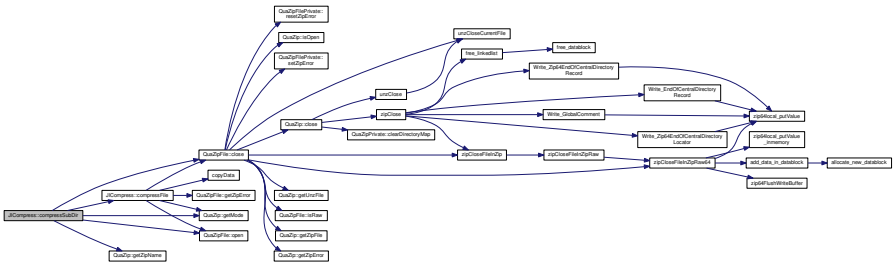
References QuaZipFile::close(), compressFile(), QuaZip::getMode(), QuaZip::getZipName(), QuaZip::mdAdd, QuaZip::mdAppend, QuaZip::mdCreate, and QuaZipFile::open().

Referenced by compressDir().

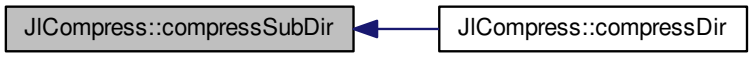
```
103                                                                                                 {
104     // zip: oggetto dove aggiungere il file
105     // dir: cartella reale corrente
106     // origDir: cartella reale originale
107     // (path(dir)-path(origDir)) = path interno all'oggetto zip
108
109     // Controllo l'apertura dello zip
110     if (!zip) return false;
111     if (zip->getMode()!=QuaZip::mdCreate &&
112         zip->getMode()!=QuaZip::mdAppend &&
113         zip->getMode()!=QuaZip::mdAdd) return false;
114
115     // Controllo la cartella
116     QDir directory(dir);
117     if (!directory.exists()) return false;
118
119     QDir origDirectory(origDir);
120     if (dir != origDir) {
121         QuaZipFile dirZipFile(zip);
122         if (!dirZipFile.open(QIODevice::WriteOnly,
123             QuaZipNewInfo(origDirectory.relativeFilePath(dir) + "/", dir), 0, 0, 0)) {
124                 return false;
125         }
126         dirZipFile.close();
127     }
128
129
130     // Se comprimo anche le sotto cartelle
131     if (recursive) {
132         // Per ogni sotto cartella
133         QFileInfoList files = directory.entryInfoList(QDir::AllDirs|QDir::NoDotAndDotDot);
134         Q_FOREACH (QFileInfo file, files) {
135             // Comprimo la sotto cartella
136             if(!compressSubDir(zip,file.absoluteFilePath(),origDir,recursive)) return false;
137         }
138     }
139
140     // Per ogni file nella cartella
141     QFileInfoList files = directory.entryInfoList(QDir::Files);
142     Q_FOREACH (QFileInfo file, files) {
143         // Se non e un file o e il file compresso che sto creando
```

```
144         if(!file.isFile()||file.absoluteFilePath()==zip->getZipName()) continue;
145
146         // Creo il nome relativo da usare all'interno del file compresso
147         QString filename = origDirectory.relativeFilePath(file.absoluteFilePath());
148
149         // Comprimo il file
150         if (!compressFile(zip,file.absoluteFilePath(),filename)) return false;
151     }
152
153     return true;
154 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.21.2.6 QStringList JICompress::extractDir (QString *fileCompressed*, QString *dir* = QString()) [static]

OK Estrae il file fileCompressed nella cartella dir.

Parameters

fileCompressed	The name of the archive.
dir	The directory to extract to, the current directory if left empty.

Returns

The list of the full paths of the files extracted, empty on failure.

Se dir = "" allora il file viene estratto nella cartella corrente. Se la funzione fallisce cancella i file che si e tentato di estrarre. Restituisce i nomi assoluti dei file estratti.

La funzione fallisce se:

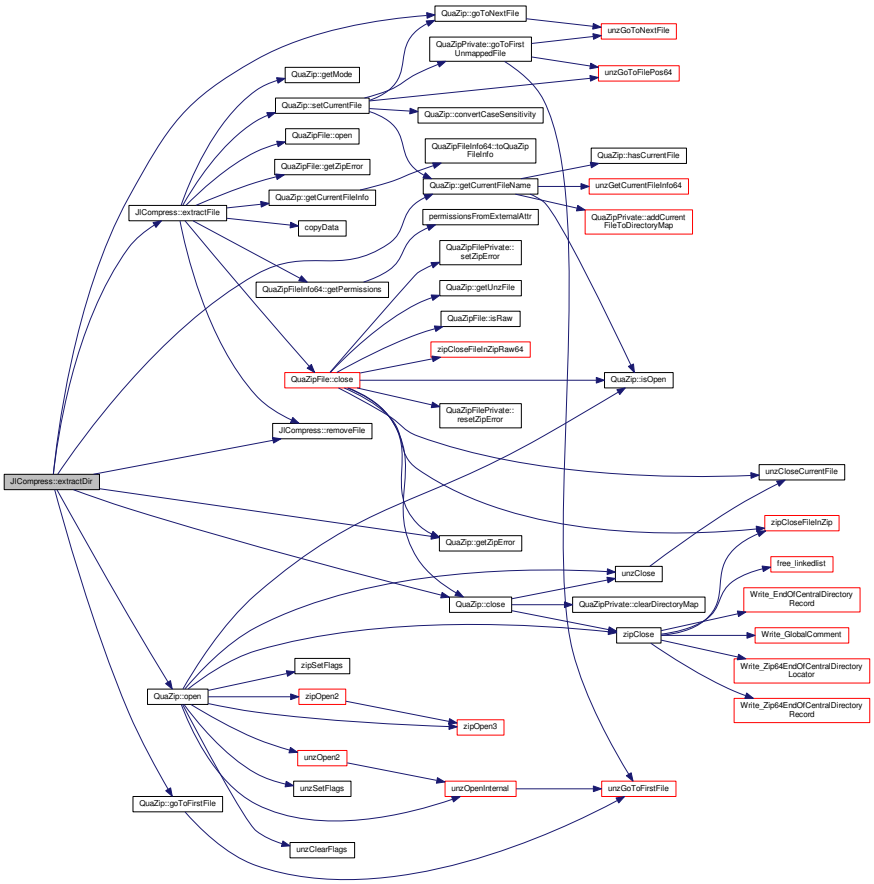
- non si riesce ad aprire l'oggetto zip;
- la compressione di un file fallisce;
- non si riesce a chiudere l'oggetto zip;

Definition at line 451 of file JICompress.cpp.

References QuaZip::close(), extractFile(), QuaZip::getCurrentFileName(), QuaZip::getZipError(), QuaZip::goToFirstFile(), QuaZip::goToNextFile(), QuaZip::mdUnzip, QuaZip::open(), and removeFile().

```
451                                                                 {
452     // Apro lo zip
453     QuaZip zip(fileCompressed);
454     if(!zip.open(QuaZip::mdUnzip)) {
455         return QStringList();
456     }
457
458     QDir directory(dir);
459     QStringList extracted;
460     if (!zip.goToFirstFile()) {
461         return QStringList();
462     }
463     do {
464         QString name = zip.getCurrentFileName();
465         QString absFilePath = directory.absoluteFilePath(name);
466         if (!extractFile(&zip, "", absFilePath)) {
467             removeFile(extracted);
468             return QStringList();
469         }
470         extracted.append(absFilePath);
471     } while (zip.goToNextFile());
472
473     // Chiudo il file zip
474     zip.close();
475     if(zip.getZipError()!=0) {
476         removeFile(extracted);
477         return QStringList();
478     }
479
480     return extracted;
481 }
```

Here is the call graph for this function:



12.21.2.7 bool JICompress::extractFile (QuaZip * *zip*, QString *fileName*, QString *fileDest*) [static],[private]

OK Estrae il file fileName, contenuto nell'oggetto zip, con il nome fileDest.

Parameters

zip	The opened zip archive to extract from.
fileName	The full name of the file to extract.
fileDest	The full path to the destination file.

Returns

true if success, false otherwise.

Se la funzione fallisce restituisce false e cancella il file che si e tentato di estrarre.

La funzione fallisce se:

- zip==NULL;
- l'oggetto zip e stato aperto in una modalita non compatibile con l'estrazione di file;
- non e possibile aprire il file all'interno dell'oggetto zip;
- non e possibile creare il file estratto;
- si e rilevato un errore nella copia dei dati (1);
- non e stato possibile chiudere il file all'interno dell'oggetto zip (1);

(1): prima di uscire dalla funzione cancella il file estratto.

Definition at line 170 of file JICompress.cpp.

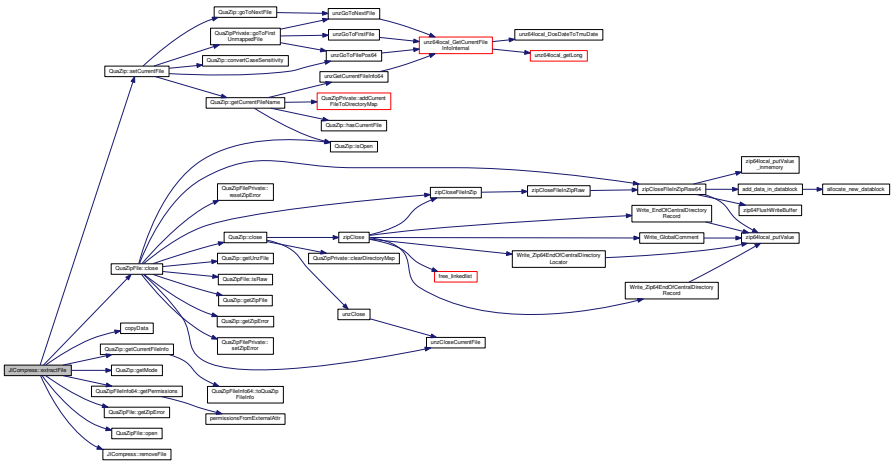
References QuaZipFile::close(), copyData(), QuaZip::getCurrentFileInfo(), QuaZip::getMode(), QuaZipFileInfo64::getPermissions(), QuaZipFile::getZipError(), QuaZip::mdUnzip, QuaZipFile::open(), removeFile(), QuaZip::setCurrentFile(), and UNZ_OK.

Referenced by extractDir(), extractFile(), and extractFiles().

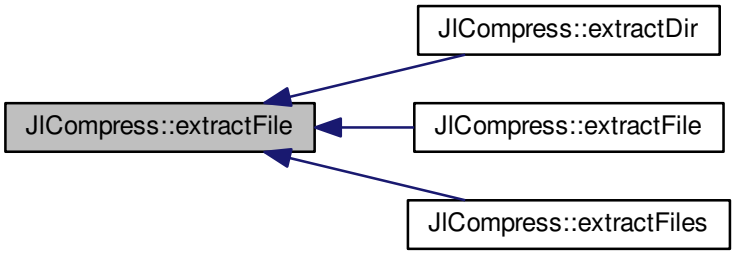
```
170                                                                                 {
171     // zip: oggetto dove aggiungere il file
172     // filename: nome del file reale
173     // fileincompress: nome del file all'interno del file compresso
174
175     // Controllo l'apertura dello zip
176     if (!zip) return false;
177     if (zip->getMode()!=QuaZip::mdUnzip) return false;
178
179     // Apro il file compresso
180     if (!fileName.isEmpty())
181         zip->setCurrentFile(fileName);
182     QuaZipFile inFile(zip);
183     if(!inFile.open(QIODevice::ReadOnly) || inFile.getZipError()!=UNZ_OK) return false;
184
185     // Controllo esistenza cartella file risultato
186     QDir curDir;
187     if (fileDest.endsWith('/')) {
188         if (!curDir.mkpath(fileDest)) {
189             return false;
190         }
191     } else {
192         if (!curDir.mkpath(QFileInfo(fileDest).absolutePath())) {
193             return false;
194         }
195     }
196
197     QuaZipFileInfo64 info;
198     if (!zip->getCurrentFileInfo(&info))
199         return false;
200
201     QFile::Permissions srcPerm = info.getPermissions();
202     if (fileDest.endsWith('/') && QFileInfo(fileDest).isDir()) {
203         if (srcPerm != 0) {
```

```
204             QFile(fileDest).setPermissions(srcPerm);
205         }
206         return true;
207     }
208
209     // Apro il file risultato
210     QFile outFile;
211     outFile.setFileName(fileDest);
212     if(!outFile.open(QIODevice::WriteOnly)) return false;
213
214     // Copio i dati
215     if (!copyData(inFile, outFile) || inFile.getZipError()!=UNZ_OK) {
216         outFile.close();
217         removeFile(QStringList(fileDest));
218         return false;
219     }
220     outFile.close();
221
222     // Chiudo i file
223     inFile.close();
224     if (inFile.getZipError()!=UNZ_OK) {
225         removeFile(QStringList(fileDest));
226         return false;
227     }
228
229     if (srcPerm != 0) {
230         outFile.setPermissions(srcPerm);
231     }
232     return true;
233 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.21.2.8 QString JICompress::extractFile (QString *fileCompressed,* QString *fileName,* QString *fileDest* = `QString()`) [static]

OK Estrae il file fileName, contenuto nel file fileCompressed, con il nome fileDest.

Parameters

fileCompressed	The name of the archive.
fileName	The file to extract.
fileDest	The destination file, assumed to be identical to *file* if left empty.

Returns

The list of the full paths of the files extracted, empty on failure.

Se fileDest = "" allora il file viene estratto con lo stesso nome con cui e stato compresso. Se la funzione fallisce cancella il file che si e tentato di estrarre. Restituisce il nome assoluto del file estratto.

La funzione fallisce se:

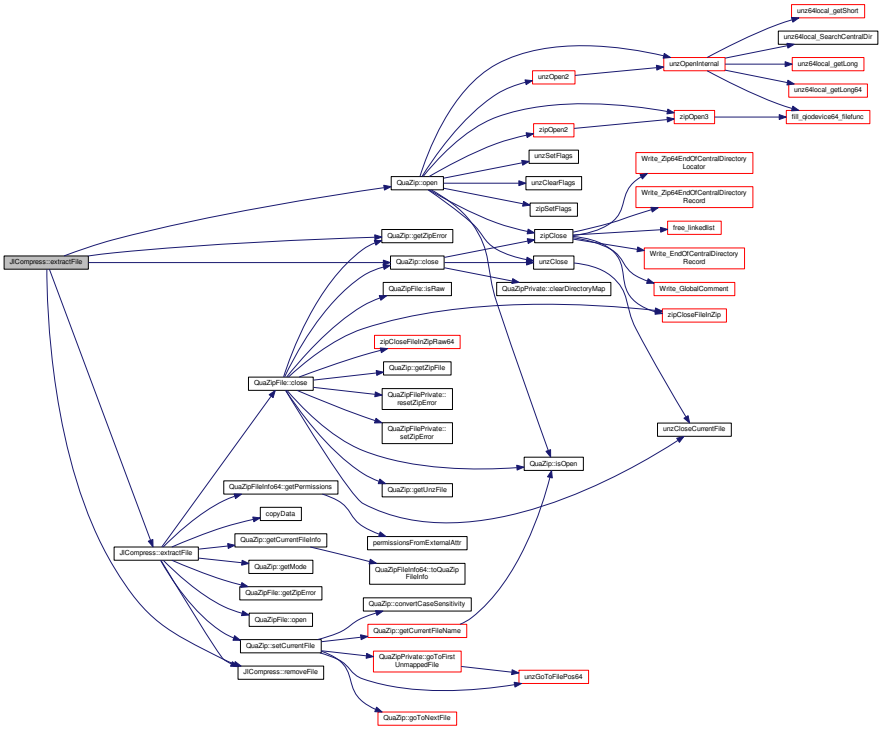
- non si riesce ad aprire l'oggetto zip;
- l'estrazione del file fallisce;
- non si riesce a chiudere l'oggetto zip;

Definition at line 377 of file JICompress.cpp.

References QuaZip::close(), extractFile(), QuaZip::getZipError(), QuaZip::mdUnzip, QuaZip::open(), and removeFile().

```
377                                                                                                 {
378     // Apro lo zip
379     QuaZip zip(fileCompressed);
380     if(!zip.open(QuaZip::mdUnzip)) {
381         return QString();
382     }
383
384     // Estraggo il file
385     if (fileDest.isEmpty())
386         fileDest = fileName;
387     if (!extractFile(&zip,fileName,fileDest)) {
388         return QString();
389     }
390
391     // Chiudo il file zip
392     zip.close();
393     if(zip.getZipError()!=0) {
394         removeFile(QStringList(fileDest));
395         return QString();
396     }
397     return QFileInfo(fileDest).absoluteFilePath();
398 }
```

Here is the call graph for this function:



12.21.2.9 QStringList JICompress::extractFiles (QString *fileCompressed,* QStringList *files,* QString *dir* = `QString()`) `[static]`

OK Estrae i file specificati in files, contenuti nel file fileCompressed, nella cartella dir.

Parameters

fileCompressed	The name of the archive.
files	The file list to extract.
dir	The directory to put the files to, the current directory if left empty.

Returns

The list of the full paths of the files extracted, empty on failure.

La struttura a cartelle del file compresso viene rispettata. Se dir = "" allora il file viene estratto nella cartella corrente. Se la funzione fallisce cancella i file che si e tentato di estrarre. Restituisce i nomi assoluti dei file estratti.

La funzione fallisce se:

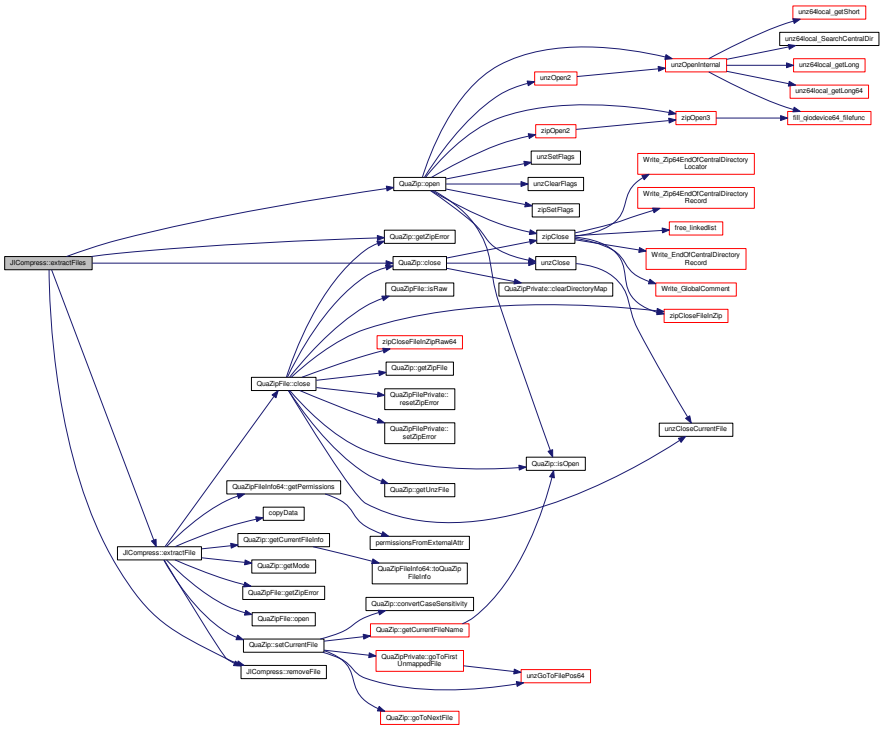
- non si riesce ad aprire l'oggetto zip;
- l'estrazione di un file fallisce;
- non si riesce a chiudere l'oggetto zip;

Definition at line 412 of file JICompress.cpp.

References QuaZip::close(), extractFile(), QuaZip::getZipError(), QuaZip::mdUnzip, QuaZip::open(), and removeFile().

```
412                                                                                                 {
413     // Creo lo zip
414     QuaZip zip(fileCompressed);
415     if(!zip.open(QuaZip::mdUnzip)) {
416         return QStringList();
417     }
418
419     // Estraggo i file
420     QStringList extracted;
421     for (int i=0; i<files.count(); i++) {
422         QString absPath = QDir(dir).absoluteFilePath(files.at(i));
423         if (!extractFile(&zip, files.at(i), absPath)) {
424             removeFile(extracted);
425             return QStringList();
426         }
427         extracted.append(absPath);
428     }
429
430     // Chiudo il file zip
431     zip.close();
432     if(zip.getZipError()!=0) {
433         removeFile(extracted);
434         return QStringList();
435     }
436
437     return extracted;
438 }
```

Here is the call graph for this function:



12.21.2.10 QStringList JlCompress::getFileList (QString *fileCompressed*) [static]

OK Restituisce la lista dei file resenti nel file compresso fileCompressed.

Returns

The list of the files in the archive, or, more precisely, the list of the entries, including both files and directories if they are present separately.

Se la funzione fallisce, restituisce un elenco vuoto.

La funzione fallisce se:

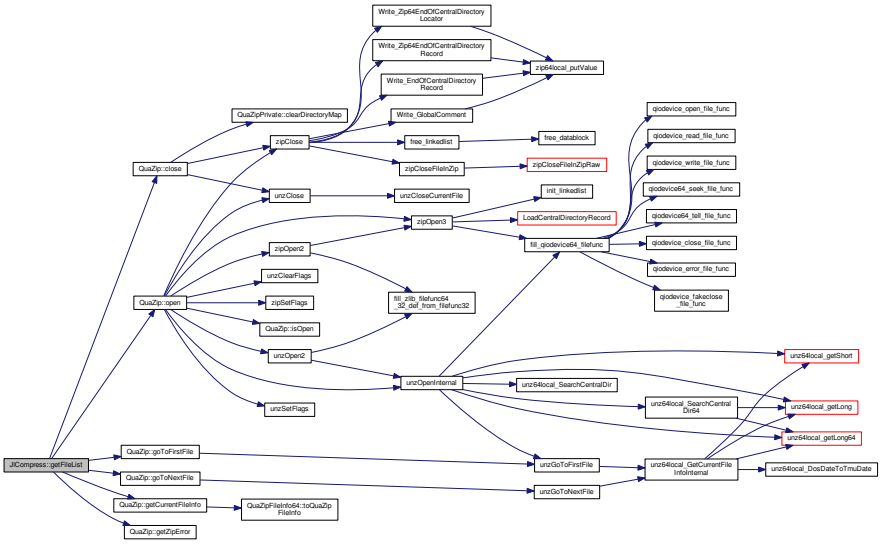
- non si riesce ad aprire l'oggetto zip;
- la richiesta di informazioni di un file fallisce;
- non si riesce a chiudere l'oggetto zip;

Definition at line 492 of file JICompress.cpp.

References QuaZip::close(), QuaZip::getCurrentFileInfo(), QuaZip::getZipError(), QuaZip::goToFirstFile(), QuaZip::goToNextFile(), QuaZip::mdUnzip, QuaZipFileInfo64::name, and QuaZip::open().

```
492                                                         {
493     // Apro lo zip
494     QuaZip* zip = new QuaZip(QFileInfo(fileCompressed).absoluteFilePath());
495     if(!zip->open(QuaZip::mdUnzip)) {
496         delete zip;
497         return QStringList();
498     }
499
500     // Estraggo i nomi dei file
501     QStringList lst;
502     QuaZipFileInfo64 info;
503     for(bool more=zip->goToFirstFile(); more; more=zip->
      goToNextFile()) {
504         if(!zip->getCurrentFileInfo(&info)) {
505             delete zip;
506             return QStringList();
507         }
508         lst << info.name;
509         //info.name.toLocal8Bit().constData()
510     }
511
512     // Chiudo il file zip
513     zip->close();
514     if(zip->getZipError()!=0) {
515         delete zip;
516         return QStringList();
517     }
518     delete zip;
519
520     return lst;
521 }
```

Here is the call graph for this function:



12.21.2.11 bool JlCompress::removeFile (QStringList *listFile*) [static],[private]

Rimuove i file il cui nome e specificato all'interno di listFile.

Parameters

listFile	The list of files to remove.

Returns

true if success, false otherwise.

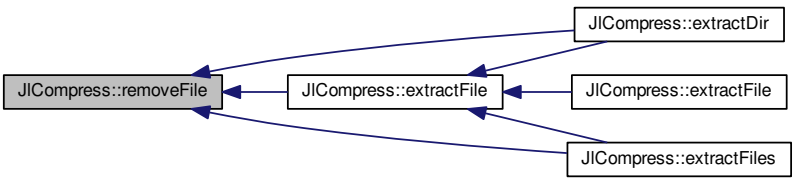
Restituisce true se tutti i file sono stati cancellati correttamente, attenzione perche puo restituire false anche se alcuni file non esistevano e si e tentato di cancellarli.

Definition at line 241 of file JlCompress.cpp.

Referenced by extractDir(), extractFile(), and extractFiles().

```
241                                                 {
242     bool ret = true;
243     // Per ogni file
244     for (int i=0; i<listFile.count(); i++) {
245         // Lo elimino
246         ret = ret && QFile::remove(listFile.at(i));
247     }
248     return ret;
249 }
```

Here is the caller graph for this function:



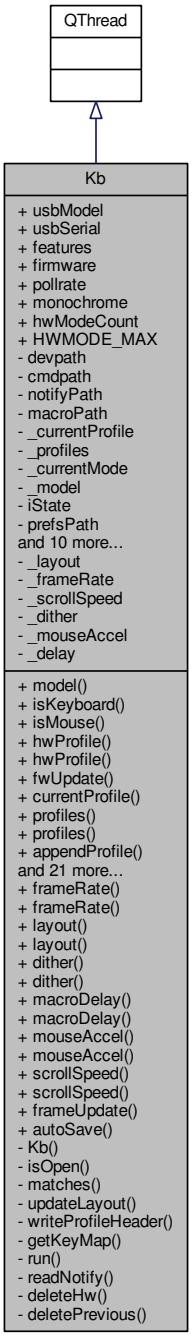
The documentation for this class was generated from the following files:

- src/ckb/quazip/JlCompress.h
- src/ckb/quazip/JlCompress.cpp

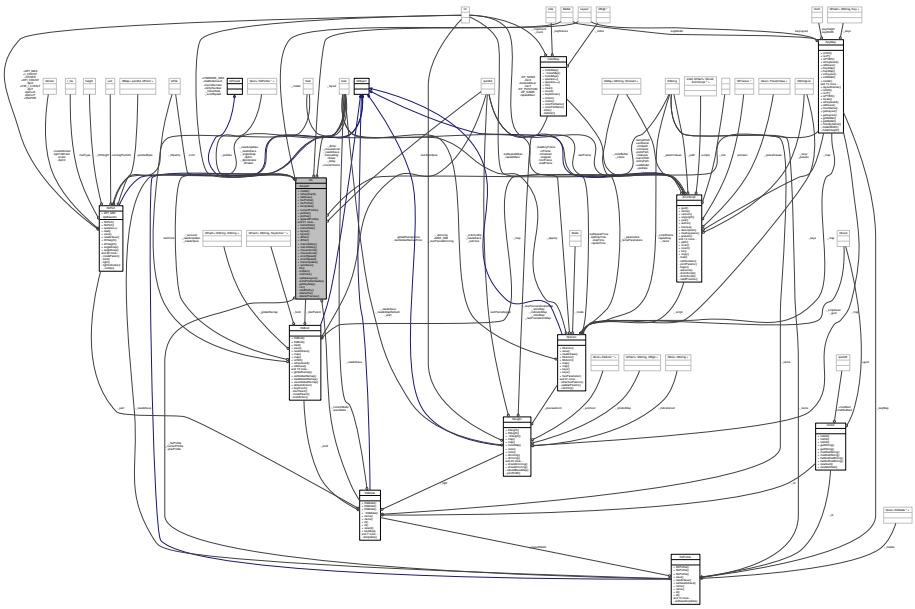
12.22 Kb Class Reference

`#include <src/ckb/kb.h>`

Inheritance diagram for Kb:



Collaboration diagram for Kb:



Public Slots

- void frameUpdate ()
- void autoSave ()

Signals

- void infoUpdated ()
- void profileAdded ()
- void profileRenamed ()
- void modeRenamed ()
- void profileChanged ()
- void modeChanged (bool spontaneous)
- void fwUpdateProgress (int current, int total)
- void fwUpdateFinished (bool succeeded)

Public Member Functions

- KeyMap::Model model () const
- bool isKeyboard () const
- bool isMouse () const
- KbProfile * hwProfile ()
- void hwProfile (KbProfile *newHwProfile)
- void fwUpdate (const QString &path)
- KbProfile * currentProfile ()
- const QList< KbProfile * > & profiles () const
- void profiles (const QList< KbProfile * > &newProfiles)
- void appendProfile (KbProfile *newProfile)
- int indexOf (KbProfile *profile)
- KbProfile * find (const QUuid &id)
- KbMode * currentMode ()

- KbLight * currentLight ()
- KbBind * currentBind ()
- KbPerf * currentPerf ()
- void setCurrentProfile (KbProfile *profile, bool spontaneous=true)
- void setCurrentMode (KbProfile *profile, KbMode *mode, bool spontaneous=true)
- void setCurrentMode (KbProfile *profile, int index, bool spontaneous=true)
- void setCurrentMode (KbMode *mode, bool spontaneous=true)
- KbProfile * newProfile ()
- KbProfile * newProfile (KbProfile *other)
- KbMode * newMode ()
- KbMode * newMode (KbMode *other)
- void load ()
- void save ()
- bool needsSave () const
- void hwSave ()
- int getMacroNumber ()

  For usage with macro definions, these two params must only be readable. So there are no setters.
- QString getMacroPath ()

  getMacroPath returns the macroPath (e.g. /dev/input/ckb1/notify), which we have saved in the constructor. For usage with macro definions, this param must only be readable. So there is no setter.
- ~Kb ()

Static Public Member Functions

- static int frameRate ()
- static void frameRate (int newFrameRate)
- static KeyMap::Layout layout ()
- static void layout (KeyMap::Layout newLayout)
- static bool dither ()
- static void dither (bool newDither)
- static bool macroDelay ()
- static void macroDelay (bool flag)

  Kb::macroDelay handles the UI-Element macroBox. Sends a command to the keyboard to switch on or off the delay function on very large macros.
- static bool mouseAccel ()
- static void mouseAccel (bool newAccel)
- static int scrollSpeed ()
- static void scrollSpeed (int newSpeed)

Data Fields

- QString usbModel
- QString usbSerial
- QString features
- QString firmware
- QString pollrate
- bool monochrome
- int hwModeCount

Static Public Attributes

- static const int HWMODE_MAX = 3

Private Slots

- void readNotify (QString line)
- void deleteHw ()
- void deletePrevious ()

Private Member Functions

- Kb (QObject *parent, const QString &path)
- bool isOpen () const
- bool matches (const QString &path, const QString &serial)
- void updateLayout ()
- void writeProfileHeader ()
- KeyMap getKeyMap ()
- void run ()

Private Attributes

- QString devpath

  pathVars devpath is the device root path (e.g. /dev/device/ckb1), cmdpath leads to the daemon input pipe for daemon commands, notifyPath is the standard input monitor for general purpose, macroPath added for a second thread to read macro input.
- QString cmdpath
- QString notifyPath
- QString macroPath
- KbProfile * _currentProfile
- QList< KbProfile * > _profiles
- KbMode * _currentMode
- KeyMap::Model _model
- bool iState [KbPerf::HW_I_COUNT]
- QString prefsPath
- quint64 lastAutoSave
- QString fwUpdPath
- KbProfile * _hwProfile
- KbProfile * prevProfile
- KbMode * prevMode
- QFile cmd
- int notifyNumber

  notifyNumber is the trailing number in the device path.
- int macroNumber
- bool _needsSave
- bool hwLoading [HWMODE_MAX+1]

Static Private Attributes

- static KeyMap::Layout _layout = KeyMap::NO_LAYOUT
- static int _frameRate = 30
- static int _scrollSpeed = 0
- static bool _dither = false
- static bool _mouseAccel = true
- static bool _delay = false

Friends

- class KbManager

12.22.1 Detailed Description

Definition at line 11 of file kb.h.

12.22.2 Constructor & Destructor Documentation

12.22.2.1 Kb::~Kb ( )

<

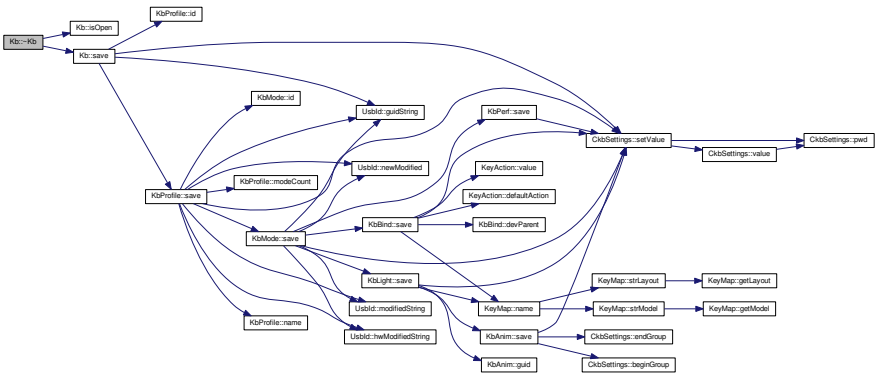
Todo I don't think, that notifypaths is used somewhere. So why do we have it? If we do not need it, searching for an ununsed notify channel can easy be refactored to a private member function.

Definition at line 141 of file kb.cpp.

References activeDevices, isOpen(), macroPath, notifyNumber, notifyPaths, and save().

```
141         {
142     // Save settings first
143     save();
144
145     // remove the notify channel from the list of notifyPaths.
148     notifyPaths.remove(macroPath);
149
150     // Kill notification thread and remove node
151     activeDevices.remove(this);
152     if(!isOpen()){
153         terminate();
154         wait(1000);
155         return;
156     }
157     if(notifyNumber > 0)
158         cmd.write(QString("idle\nnotifyoff %1\n").arg(notifyNumber).toLatin1());
159     cmd.flush();
160     terminate();
161     wait(1000);
162     cmd.close();
163 }
```

Here is the call graph for this function:



12.22.2.2 Kb::Kb (QObject * *parent*, const QString & *path*) [private]

<

Todo Is adding notify2 to the notifypaths neccessary?

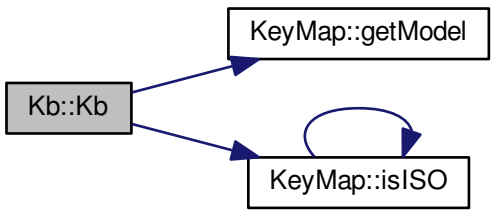
Definition at line 19 of file kb.cpp.

References _delay, _dither, _frameRate, _layout, _mouseAccel, _scrollSpeed, activeDevices, cmdpath, features, firmware, KeyMap::getModel(), hwLoading, hwModeCount, infoUpdated(), KeyMap::isISO(), iState, KeyMap::K95, macroNumber, macroPath, monochrome, KeyMap::NO_MODEL, notifyNumber, notifyPath, notifyPathMutex, notifyPaths, pollrate, prefsPath, usbModel, and usbSerial.

```
19                                      :
20     QThread(parent), features("N/A"), firmware("N/A"),
      pollrate("N/A"), monochrome(false),
21     devpath(path), cmdpath(path + "/cmd"), notifyPath(path + "/notify1"),
      macroPath(path + "/notify2"),
22     _currentProfile(0), _currentMode(0), _model(
      KeyMap::NO_MODEL),
23     lastAutoSave(QDateTime::currentMSecsSinceEpoch()),
24     _hwProfile(0), prevProfile(0), prevMode(0),
25     cmd(cmdpath), notifyNumber(1), macroNumber(2),
      _needsSave(false)
26 {
27     memset(iState, 0, sizeof(iState));
28     memset(hwLoading, 0, sizeof(hwLoading));
29
30     // Get the features, model, serial number, FW version (if available), and poll rate (if available) from
       /dev nodes
31     QFile ftpath(path + "/features"), mpath(path + "/model"), spath(path + "/serial"), fwpath(path + "
      /fwversion"), ppath(path + "/pollrate");
32     if(ftpath.open(QIODevice::ReadOnly)){
33         features = ftpath.read(1000);
34         features = features.trimmed();
35         ftpath.close();
36         // Read model from features (first word: vendor, second word: product)
37         QStringList list = features.split(" ");
38         if(list.length() < 2)
39             return;
40         _model = KeyMap::getModel(list[1]);
41         if(_model == KeyMap::NO_MODEL)
42             return;
43     } else
44         // Bail if features aren't readable
45         return;
46     if(features.contains("monochrome"))
47         monochrome = true;
48     if(mpath.open(QIODevice::ReadOnly)){
49         usbModel = mpath.read(100);
50         usbModel = usbModel.remove("Corsair").remove("Gaming").remove("Keyboard").remove("
      Mouse").remove("Bootloader").trimmed();
51         mpath.close();
52     }
53     if(usbModel == "")
54         usbModel = "Keyboard";
55     if(spath.open(QIODevice::ReadOnly)){
56         usbSerial = spath.read(100);
57         usbSerial = usbSerial.trimmed().toUpper();
58         spath.close();
59     }
60     if(usbSerial == "")
61         usbSerial = "Unknown-" + usbModel;
62     if(features.contains("fwversion") && fwpath.open(QIODevice::ReadOnly)){
63         firmware = fwpath.read(100);
64         firmware = QString::number(firmware.trimmed().toInt() / 100., 'f', 2);
65         fwpath.close();
66     }
67     if(features.contains("pollrate") && ppath.open(QIODevice::ReadOnly)){
68         pollrate = ppath.read(100);
69         pollrate = pollrate.trimmed();
70         ppath.close();
71     }
72
73     prefsPath = "Devices/" + usbSerial;
74
75     hwModeCount = (_model == KeyMap::K95) ? 3 : 1;
76     // Open cmd in non-blocking mode so that it doesn't lock up if nothing is reading
77     // (e.g. if the daemon crashed and didn't clean up the node)
78     int fd = open(cmdpath.toLatin1().constData(), O_WRONLY | O_NONBLOCK);
79     if(!cmd.open(fd, QIODevice::WriteOnly, QFileDevice::AutoCloseHandle))
80         return;
```

```
81
82      // Find an available notification node (if none is found, take notify1)
83      {
84          QMutexLocker locker(&notifyPathMutex);
85          for(int i = 1; i < 10; i++){
86              QString notify = QString(path + "/notify%1").arg(i);
87              if(!QFile::exists(notify) && !notifyPaths.contains(notify)){
88                  notifyNumber = i;
89                  notifyPath = notify;
90                  break;
91              }
92          }
93          notifyPaths.insert(notifyPath);
94      }
95      cmd.write(QString("notifyon %1\n").arg(notifyNumber).toLatin1());
96      cmd.flush();
97
98      // Again, find an available notification node for macro definition
99      // (if none is found, take notify2)
100      {
101          QMutexLocker locker(&notifyPathMutex);
102          for(int i = 1; i < 10; i++){
103              QString notify = QString(path + "/notify%1").arg(i);
104              if(!QFile::exists(notify) && !notifyPaths.contains(notify)){
105                  macroNumber = i;
106                  macroPath = notify;
107                  break;
108              }
109          }
110          notifyPaths.insert(notifyPath);
111      }
112      // Activate device, apply settings, and ask for hardware profile
113      cmd.write(QString("fps %1\n").arg(_frameRate).toLatin1());
114      cmd.write(QString("dither %1\n").arg(static_cast<int>(_dither)).toLatin1());
115      cmd.write(QString("\ndelay %1\n").arg(_delay? "on" : "off").toLatin1());
116  #ifdef Q_OS_MACX
117      // Write ANSI/ISO flag to daemon (OSX only)
118      cmd.write("layout ");
119      cmd.write(KeyMap::isISO(_layout) ? "iso" : "ansi");
120      // Also OSX only: scroll speed and mouse acceleration
121      cmd.write(QString("accel %1\n").arg(QString(_mouseAccel ? "on" : "off")).toLatin1());
122      cmd.write(QString("scrollspeed %1\n").arg(_scrollSpeed).toLatin1());
123  #endif
124      cmd.write(QString("\nactive\n@%1 get :hwprofileid").arg(notifyNumber).toLatin1());
125      hwLoading[0] = true;
126      for(int i = 0; i < hwModeCount; i++){
127          cmd.write(QString(" mode %1 get :hwid").arg(i + 1).toLatin1());
128          hwLoading[i + 1] = true;
129      }
130      // Ask for current indicator and key state
131      cmd.write(" get :i :keys\n");
132      cmd.flush();
133
134      emit infoUpdated();
135      activeDevices.insert(this);
136
137      // Start a separate thread to read from the notification node
138      start();
139  }
```

Here is the call graph for this function:



12.22.3 Member Function Documentation

12.22.3.1 void Kb::appendProfile (KbProfile * *newProfile*) [inline]

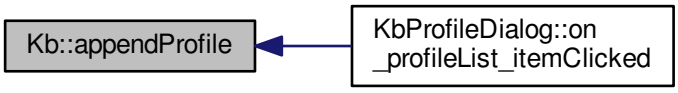
Definition at line 60 of file kb.h.

References _needsSave, and _profiles.

Referenced by KbProfileDialog::on_profileList_itemClicked().

```
60 { _needsSave = true; _profiles.append(newProfile); }
```

Here is the caller graph for this function:



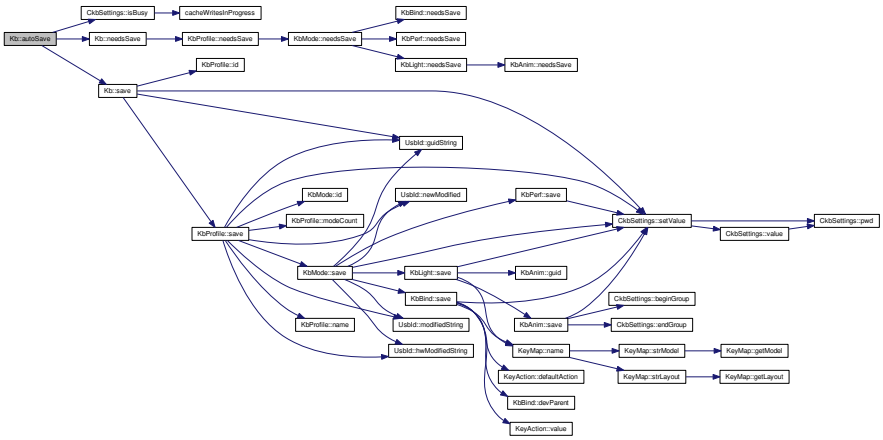
12.22.3.2 void Kb::autoSave ( ) [slot]

Definition at line 289 of file kb.cpp.

References CkbSettings::isBusy(), lastAutoSave, needsSave(), and save().

```
289                 {
290     quint64 now = QDateTime::currentMSecsSinceEpoch();
291     if(needsSave() && now >= lastAutoSave + 15 * 1000 && !
      CkbSettings::isBusy()){
292         save();
293         lastAutoSave = now;
294     }
295 }
```

Here is the call graph for this function:



12.22.3.3 KbBind* Kb::currentBind ( ) [inline]

Definition at line 67 of file kb.h.

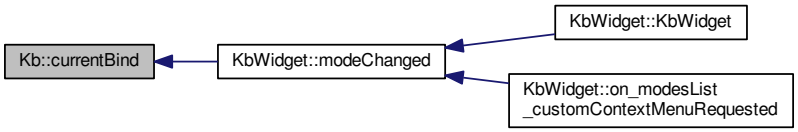
References _currentMode, and KbMode::bind().

Referenced by KbWidget::modeChanged().

```
67 { return !_currentMode ? 0 : _currentMode->bind(); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.22.3.4 KbLight* Kb::currentLight ( ) [inline]

Definition at line 66 of file kb.h.

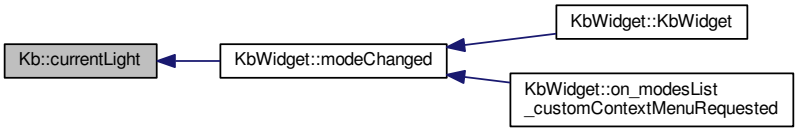
References _currentMode, and KbMode::light().

Referenced by KbWidget::modeChanged().

```
66 { return !_currentMode ? 0 : _currentMode->light(); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.22.3.5 KbMode* Kb::currentMode ( ) [inline]

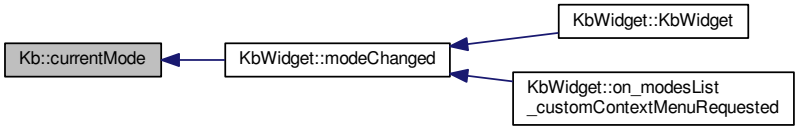
Definition at line 65 of file kb.h.

References _currentMode.

Referenced by KbWidget::modeChanged().

```
65 { return _currentMode; }
```

Here is the caller graph for this function:



12.22.3.6 KbPerf* Kb::currentPerf ( ) [inline]

Definition at line 68 of file kb.h.

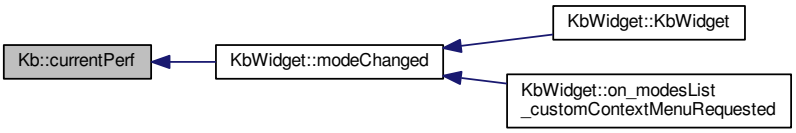
References _currentMode, and KbMode::perf().

Referenced by KbWidget::modeChanged().

```
68 { return !_currentMode ? 0 : _currentMode->perf(); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.22.3.7 KbProfile* Kb::currentProfile ( ) [inline]

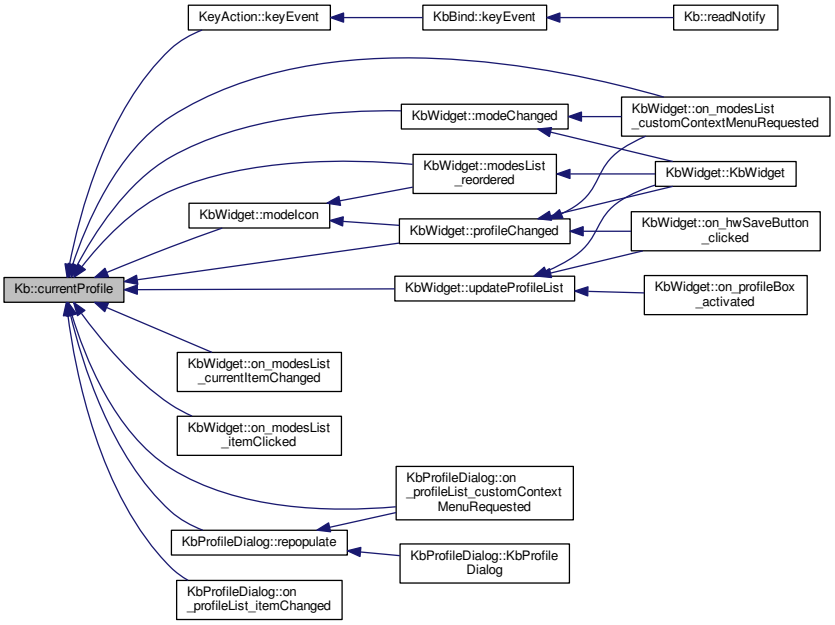
Definition at line 56 of file kb.h.

References _currentProfile.

Referenced by KeyAction::keyEvent(), KbWidget::modeChanged(), KbWidget::modeIcon(), KbWidget::modesList_reordered(), KbWidget::on_modesList_currentItemChanged(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::on_modesList_itemClicked(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_itemChanged(), KbWidget::profileChanged(), KbProfileDialog::repopulate(), and KbWidget::updateProfileList().

```
56 { return _currentProfile; }
```

Here is the caller graph for this function:



12.22.3.8 void Kb::deleteHw ( ) [private], [slot]

Definition at line 441 of file kb.cpp.

References _hwProfile.

Referenced by hwProfile().

```
441                 {
442     disconnect(_hwProfile, SIGNAL(destroyed()), this, SLOT(deleteHw()));
443     _hwProfile = 0;
444 }
```

Here is the caller graph for this function:

12.22.3.9 void Kb::deletePrevious ( ) [private], [slot]

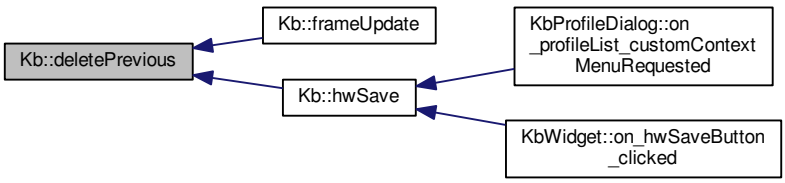
Definition at line 426 of file kb.cpp.

References prevMode.

Referenced by frameUpdate(), and hwSave().

```
426                     {
427     disconnect(prevMode, SIGNAL(destroyed()), this, SLOT(deletePrevious()));
428     prevMode = 0;
429 }
```

Here is the caller graph for this function:



12.22.3.10 static bool Kb::dither ( ) [inline], [static]

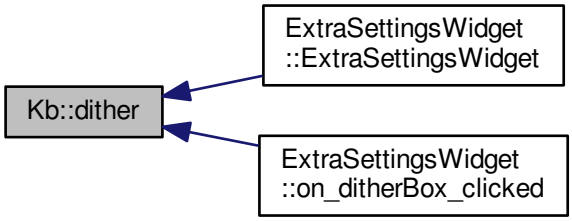
Definition at line 33 of file kb.h.

References _dither.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and ExtraSettingsWidget::on_ditherBox_clicked().

```
33 { return _dither; }
```

Here is the caller graph for this function:



12.22.3.11 void Kb::dither (bool *newDither*) [static]

Definition at line 204 of file kb.cpp.

References _dither, activeDevices, and cmd.

```
204                                {
205     if(newDither == _dither)
206         return;
207     _dither = newDither;
208     // Update all devices
209     foreach(Kb* kb, activeDevices){
210         kb->cmd.write(QString("dither %1\n").arg(static_cast<int>(newDither)).toLatin1());
211         kb->cmd.flush();
212     }
213 }
```

12.22.3.12 KbProfile* Kb::find (const QUuid & *id*) [inline]

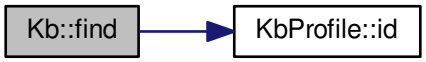
Definition at line 62 of file kb.h.

References _profiles, UsbId::guid, and KbProfile::id().

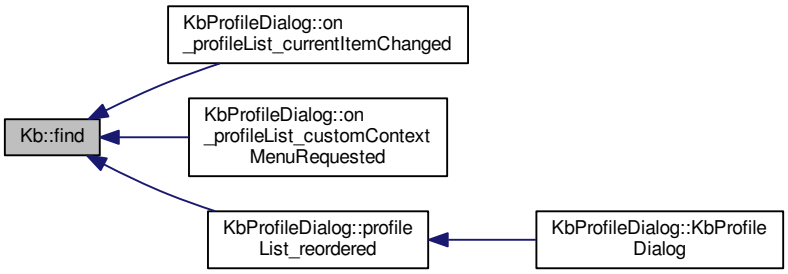
Referenced by KbProfileDialog::on_profileList_currentItemChanged(), KbProfileDialog::on_profileList_customContextMenuRequested(), and KbProfileDialog::profileList_reordered().

```
62 { foreach(KbProfile* profile, _profiles) { if(profile->id().
      guid == id) return profile; } return 0; }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.13 static int Kb::frameRate ( ) [inline], [static]

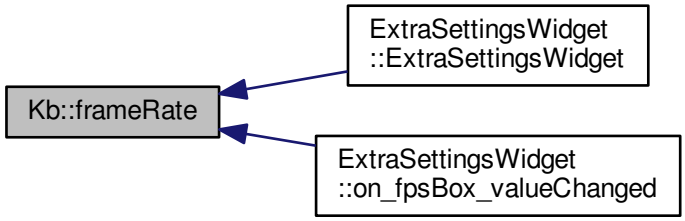
Definition at line 27 of file kb.h.

References _frameRate.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and ExtraSettingsWidget::on_fpsBox_valueChanged().

```
27 { return _frameRate; }
```

Here is the caller graph for this function:



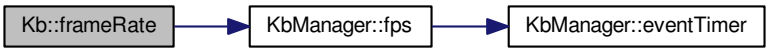
12.22.3.14 void Kb::frameRate (int *newFrameRate*) [static]

Definition at line 165 of file kb.cpp.

References _frameRate, activeDevices, cmd, and KbManager::fps().

```
165                                {
166     KbManager::fps(newFrameRate);
167     // If the rate has changed, send to all devices
168     if(newFrameRate == _frameRate)
169         return;
170     _frameRate = newFrameRate;
171     foreach(Kb* kb, activeDevices){
172         kb->cmd.write(QString("fps %1\n").arg(newFrameRate).toLatin1());
173         kb->cmd.flush();
174     }
175 }
```

Here is the call graph for this function:



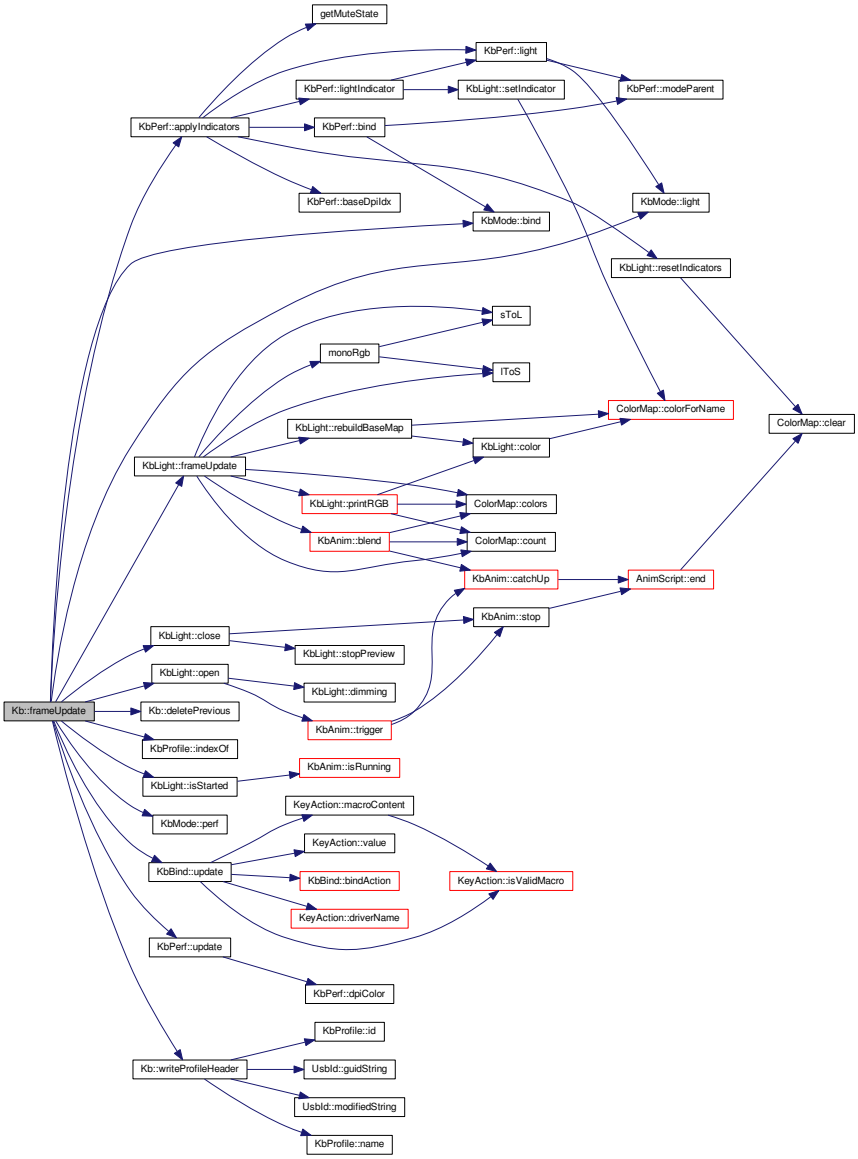
12.22.3.15 void Kb::frameUpdate ( ) [slot]

Definition at line 375 of file kb.cpp.

References _currentMode, _currentProfile, KbPerf::applyIndicators(), KbMode::bind(), KbLight::close(), deletePrevious(), KbLight::frameUpdate(), KbProfile::indexOf(), KbLight::isStarted(), iState, KbMode::light(), monochrome, notifyNumber, KbLight::open(), KbMode::perf(), prevMode, prevProfile, KbBind::update(), KbPerf::update(), and writeProfileHeader().

```
375                     {
376     // Advance animation frame
377     if(!_currentMode)
378         return;
379     KbLight* light = _currentMode->light();
380     KbBind* bind = _currentMode->bind();
381     KbPerf* perf = _currentMode->perf();
382     if(!light->isStarted()){
383         // Don't do anything until the animations are started
384         light->open();
385         return;
386     }
387
388     // Stop animations on the previously active mode (if any)
389     bool changed = false;
390     if(prevMode != _currentMode){
391         if(prevMode){
392             prevMode->light()->close();
393             disconnect(prevMode, SIGNAL(destroyed()), this, SLOT(
    deletePrevious()));
394         }
395         prevMode = _currentMode;
396         connect(prevMode, SIGNAL(destroyed()), this, SLOT(deletePrevious()));
397         changed = true;
398     }
399
400     // If the profile has changed, update it
401     if(prevProfile != _currentProfile){
402         writeProfileHeader();
403         cmd.write(" ");
404         prevProfile = _currentProfile;
405     }
406
407     // Update current mode
408     int index = _currentProfile->indexOf(_currentMode);
409     // ckb-daemon only has 6 modes: 3 hardware, 3 non-hardware. Beyond mode six, switch back to four.
410     // e.g. 1, 2, 3, 4, 5, 6, 4, 5, 6, 4, 5, 6 ...
411     if(index >= 6)
412         index = 3 + index % 3;
413
414     // Send lighting/binding to driver
415     cmd.write(QString("mode %1 switch ").arg(index + 1).toLatin1());
416     perf->applyIndicators(index, iState);
417     light->frameUpdate(cmd, monochrome);
418     cmd.write(QString("\n@%1 ").arg(notifyNumber).toLatin1());
419     bind->update(cmd, changed);
420     cmd.write(" ");
421     perf->update(cmd, notifyNumber, changed, true);
422     cmd.write("\n");
423     cmd.flush();
424 }
```

Here is the call graph for this function:



12.22.3.16 void Kb::fwUpdate (const QString & *path*)

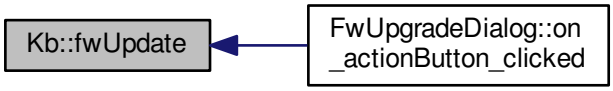
Definition at line 365 of file kb.cpp.

References fwUpdPath, and notifyNumber.

Referenced by FwUpgradeDialog::on_actionButton_clicked().

```
365                                     {
366     fwUpdPath = path;
367     // Write the active command to ensure it's not ignored
368     cmd.write("active");
369     cmd.write(QString(" @%1 ").arg(notifyNumber).toLatin1());
370     cmd.write("fwupdate ");
371     cmd.write(path.toLatin1());
372     cmd.write("\n");
373 }
```

Here is the caller graph for this function:



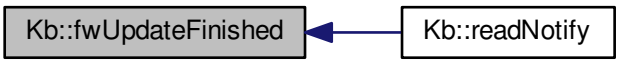
12.22.3.17 void Kb::fwUpdateFinished (bool *succeeded*) [signal]

Definition at line 277 of file moc_kb.cpp.

Referenced by readNotify().

```
278 {
279     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
280     QMetaObject::activate(this, &staticMetaObject, 7, _a);
281 }
```

Here is the caller graph for this function:



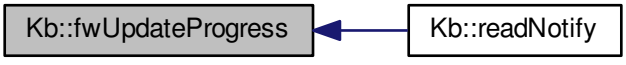
12.22.3.18 void Kb::fwUpdateProgress (int *current*, int *total*) [signal]

Definition at line 270 of file moc_kb.cpp.

Referenced by readNotify().

```
271 {
272     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)), const_cast<void*>(
      reinterpret_cast<const void*>(&_t2)) };
273     QMetaObject::activate(this, &staticMetaObject, 6, _a);
274 }
```

Here is the caller graph for this function:



12.22.3.19 KeyMap Kb::getKeyMap ( ) [private]

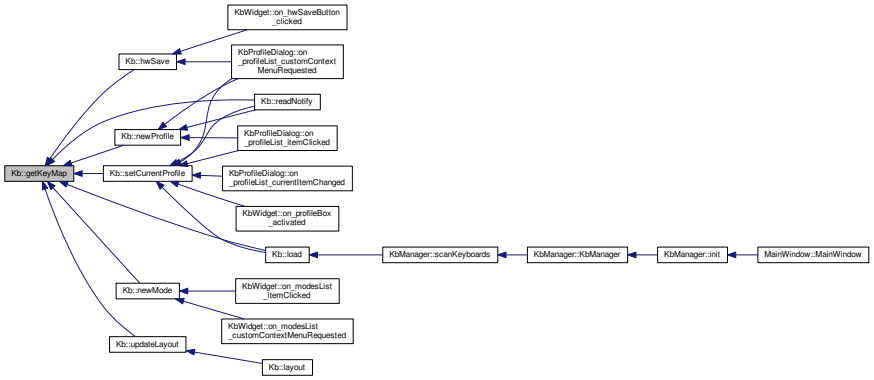
Definition at line 756 of file kb.cpp.

References _layout.

Referenced by hwSave(), load(), newMode(), newProfile(), readNotify(), setCurrentProfile(), and updateLayout().

```
756                         {
757     return KeyMap(_model, _layout);
758 }
```

Here is the caller graph for this function:



12.22.3.20 int Kb::getMacroNumber ( ) [inline]

getMacroNumber returns the macroNumber, which we have saved in the constructor. For usage with macro definions, this param must only be readable. So there is no setter.

Returns

The Number is returned as int.

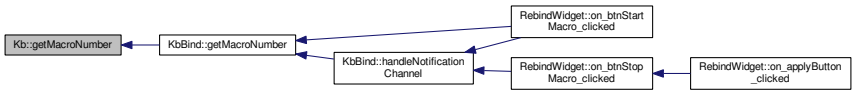
Definition at line 97 of file kb.h.

References macroNumber.

Referenced by KbBind::getMacroNumber().

```
97 { return macroNumber; }
```

Here is the caller graph for this function:



12.22.3.21 QString Kb::getMacroPath ( ) [inline]

Returns

The absolute path as String

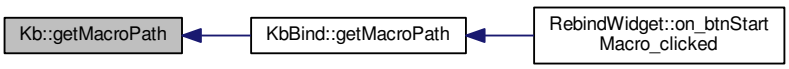
Definition at line 106 of file kb.h.

References macroPath.

Referenced by KbBind::getMacroPath().

```
106 { return macroPath; }
```

Here is the caller graph for this function:



12.22.3.22 KbProfile* Kb::hwProfile ( ) [inline]

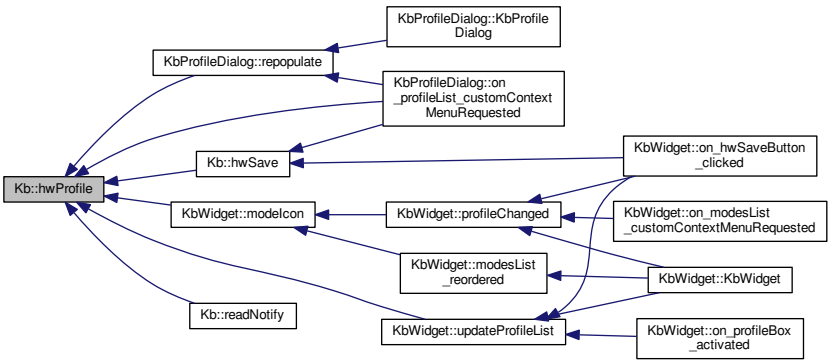
Definition at line 46 of file kb.h.

References _hwProfile.

Referenced by hwSave(), KbWidget::modeIcon(), KbProfileDialog::on_profileList_customContextMenuRequested(), readNotify(), KbProfileDialog::repopulate(), and KbWidget::updateProfileList().

```
46 { return _hwProfile; }
```

Here is the caller graph for this function:



12.22.3.23 void Kb::hwProfile (KbProfile * *newHwProfile*)

Definition at line 431 of file kb.cpp.

References _hwProfile, and deleteHw().

```
431                                         {
432     if(_hwProfile == newHwProfile)
433         return;
434     if(_hwProfile)
435         disconnect(_hwProfile, SIGNAL(destroyed()), this, SLOT(
      deleteHw()));
436     _hwProfile = newHwProfile;
437     if(_hwProfile)
438         connect(_hwProfile, SIGNAL(destroyed()), this, SLOT(deleteHw()));
439 }
```

Here is the call graph for this function:

12.22.3.24 void Kb::hwSave ( )

Definition at line 297 of file kb.cpp.

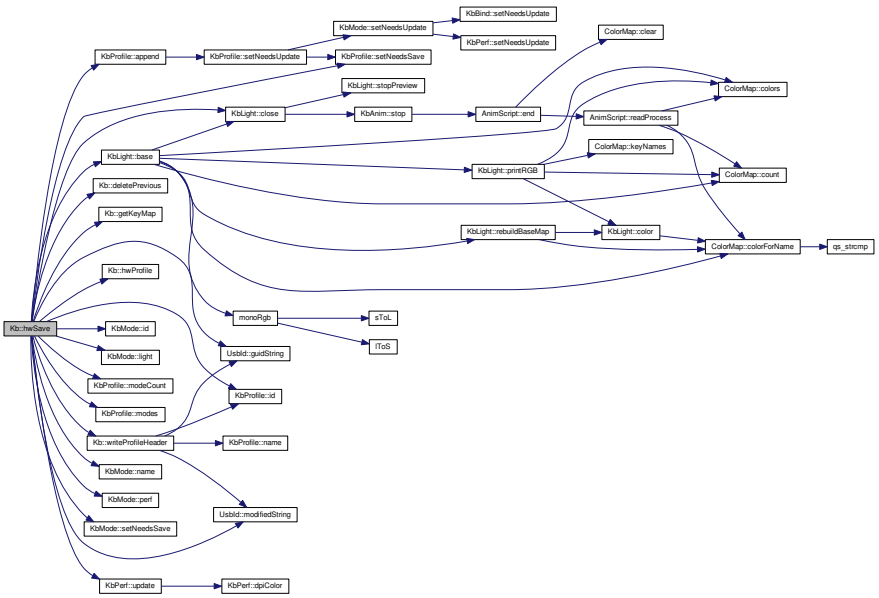
References _currentMode, _currentProfile, _hwProfile, KbProfile::append(), KbLight::base(), KbLight::close(), deletePrevious(), getKeyMap(), UsbId::guidString(), hwModeCount, UsbId::hwModified, hwProfile(), KbProfile::id(), KbMode::id(), KbMode::light(), KbProfile::modeCount(), KbProfile::modes(), UsbId::modified, UsbId::modifiedString(), monochrome, KbMode::name(), notifyNumber, KbMode::perf(), prevMode, KbProfile::setNeedsSave(), KbMode::setNeedsSave(), KbPerf::update(), and writeProfileHeader().

Referenced by KbWidget::on_hwSaveButton_clicked(), and KbProfileDialog::on_profileList_customContextMenuRequested().

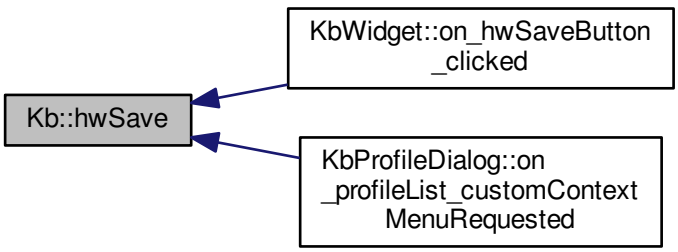
```
297                 {
298     if(!_currentProfile)
299         return;
300     // Close active lighting (if any)
301     if(prevMode){
302         prevMode->light()->close();
303         deletePrevious();
304     }
305     hwProfile(_currentProfile);
306     _hwProfile->id().hwModified = _hwProfile->id().
      modified;
307     _hwProfile->setNeedsSave();
308     // Re-send the current profile from scratch to ensure consistency
309     writeProfileHeader();
310     // Make sure there are enough modes
311     while(_currentProfile->modeCount() < hwModeCount)
312         _currentProfile->append(new KbMode(this,
      getKeyMap()));
313     // Write only the base colors of each mode, no animations
314     for(int i = 0; i < hwModeCount; i++){
315         KbMode* mode = _currentProfile->modes()[i];
316         cmd.write(QString("\nmode %1").arg(i + 1).toLatin1());
317         KbLight* light = mode->light();
318         KbPerf* perf = mode->perf();
319         if(mode == _currentMode)
320             cmd.write(" switch");
321         // Write the mode name and ID
322         cmd.write(" name ");
323         cmd.write(QUrl::toPercentEncoding(mode->name()));
324         cmd.write(" id ");
325         cmd.write(mode->id().guidString().toLatin1());
326         cmd.write(" ");
327         cmd.write(mode->id().modifiedString().toLatin1());
328         cmd.write(" ");
329         // Write lighting and performance
330         light->base(cmd, true, monochrome);
```

```
331         cmd.write(" ");
332         perf->update(cmd, notifyNumber, true, false);
333         // Update mode ID
334         mode->id().hwModified = mode->id().modified;
335         mode->setNeedsSave();
336     }
337     cmd.write("\n");
338
339     // Save the profile to memory
340     cmd.write("hwsave\n");
341     cmd.flush();
342 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.25 int Kb::indexOf (KbProfile * *profile*) [inline]

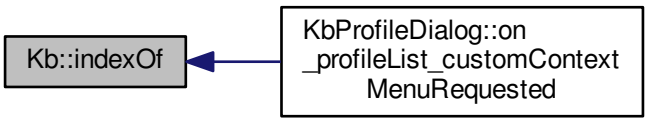
Definition at line 61 of file kb.h.

References _profiles.

Referenced by KbProfileDialog::on_profileList_customContextMenuRequested().

```
61 { return _profiles.indexOf(profile); }
```

Here is the caller graph for this function:



12.22.3.26 void Kb::infoUpdated ( ) [signal]

Definition at line 233 of file moc_kb.cpp.

Referenced by Kb(), load(), and updateLayout().

```
234 {
235     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
236 }
```

Here is the caller graph for this function:

12.22.3.27 bool Kb::isKeyboard ( ) const [inline]

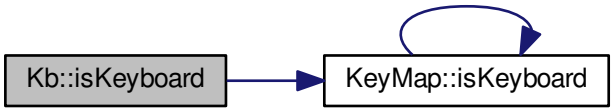
Definition at line 23 of file kb.h.

References KeyMap::isKeyboard().

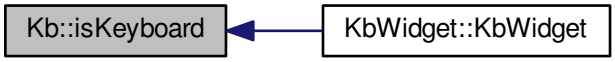
Referenced by KbWidget::KbWidget().

```
23 { return KeyMap::isKeyboard(_model); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.28 bool Kb::isMouse ( ) const [inline]

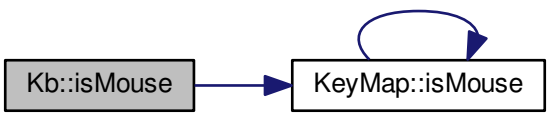
Definition at line 24 of file kb.h.

References KeyMap::isMouse().

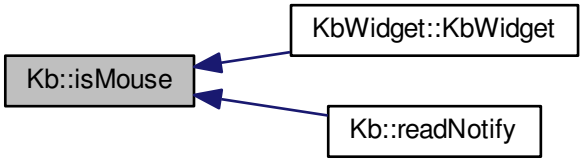
Referenced by KbWidget::KbWidget(), and readNotify().

```
24 { return KeyMap::isMouse(_model); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



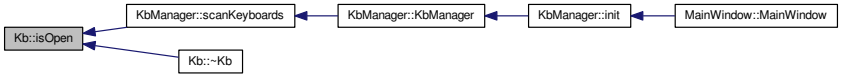
12.22.3.29 bool Kb::isOpen ( ) const [inline], [private]

Definition at line 147 of file kb.h.

Referenced by KbManager::scanKeyboards(), and ~Kb().

```
147 { return cmd.isOpen(); }
```

Here is the caller graph for this function:



12.22.3.30 static KeyMap::Layout Kb::layout ( ) [inline], [static]

Definition at line 30 of file kb.h.

References _layout.

Referenced by LayoutDialog::LayoutDialog(), SettingsWidget::on_layoutBox_activated(), and SettingsWidget::~SettingsWidget().

```
30 { return _layout; }
```

Here is the caller graph for this function:

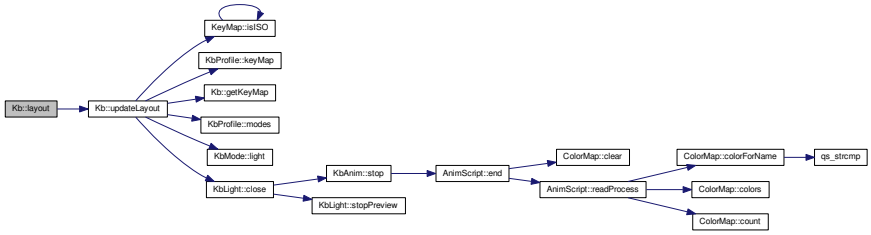
12.22.3.31 void Kb::layout (KeyMap::Layout *newLayout*) [static]

Definition at line 177 of file kb.cpp.

References _layout, activeDevices, KeyMap::NO_LAYOUT, and updateLayout().

```
177                                     {
178     if(newLayout == KeyMap::NO_LAYOUT || newLayout == _layout)
179         return;
180     _layout = newLayout;
181     // Update all devices
182     foreach(Kb* kb, activeDevices)
183         kb->updateLayout();
184 }
```

Here is the call graph for this function:



12.22.3.32 void Kb::load ( )

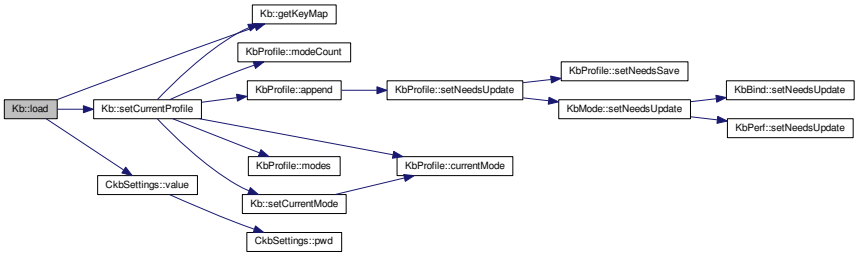
Definition at line 241 of file kb.cpp.

References _needsSave, _profiles, current, getKeyMap(), infoUpdated(), prefsPath, profileAdded(), setCurrentProfile(), and CkbSettings::value().

Referenced by KbManager::scanKeyboards().

```
241             {
242     if(prefsPath.isEmpty())
243         return;
244     _needsSave = false;
245     CkbSettings settings(prefsPath);
246     // Read profiles
247     KbProfile* newCurrentProfile = 0;
248     QString current = settings.value("CurrentProfile").toString().trimmed().toUpper();
249     foreach(QString guid, settings.value("Profiles").toString().split(" ")){
250         guid = guid.trimmed().toUpper();
251         if(guid != ""){
252             KbProfile* profile = new KbProfile(this, getKeyMap(), settings, guid
    );
253             _profiles.append(profile);
254             if(guid == current || !newCurrentProfile)
255                 newCurrentProfile = profile;
256         }
257     }
258     if(newCurrentProfile)
259         setCurrentProfile(newCurrentProfile);
260     else {
261         // If nothing was loaded, load the demo profile
262         QSettings demoSettings(":/txt/demoprofile.conf", QSettings::IniFormat, this);
263         CkbSettings cSettings(demoSettings);
264         KbProfile* demo = new KbProfile(this, getKeyMap(), cSettings, "
    {BA7FC152-2D51-4C26-A7A6-A036CC93D924}");
265         _profiles.append(demo);
266         setCurrentProfile(demo);
267     }
268
269     emit infoUpdated();
270     emit profileAdded();
271 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.22.3.33 static bool Kb::macroDelay ( ) [inline], [static]

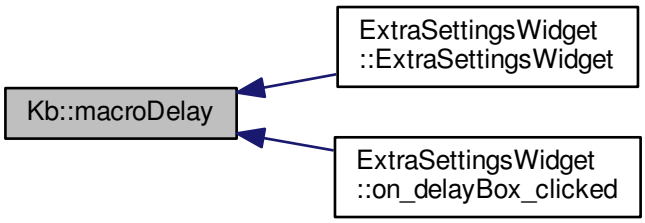
Definition at line 36 of file kb.h.

References _delay.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and ExtraSettingsWidget::on_delayBox_clicked().

```
36 { return _delay; }
```

Here is the caller graph for this function:



12.22.3.34 void Kb::macroDelay (bool *flag*) [static]

Parameters

flag	true: Switch on delay function, else switch off

Definition at line 787 of file kb.cpp.

References _delay, activeDevices, and cmd.

```
787                             {
788     _delay = flag;
789
790     foreach(Kb* kb, activeDevices){
791         kb->cmd.write(QString("\ndelay %1\n").arg(flag? "on" : "off").toLatin1());
792     }
793 }
```

12.22.3.35 bool Kb::matches (const QString & *path,* const QString & *serial*) [inline], [private]

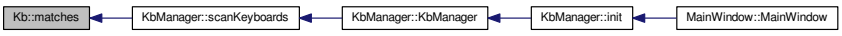
Definition at line 157 of file kb.h.

References devpath, and usbSerial.

Referenced by KbManager::scanKeyboards().

```
157 { return path.trimmed() == devpath.trimmed() && usbSerial == serial.trimmed().toUpper(); }
```

Here is the caller graph for this function:



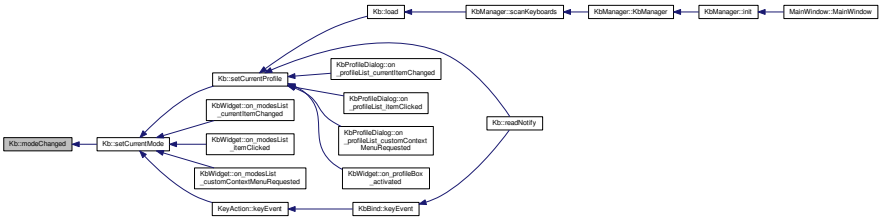
12.22.3.36 void Kb::modeChanged (bool *spontaneous*) [signal]

Definition at line 263 of file moc_kb.cpp.

Referenced by setCurrentMode().

```
264 {
265     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
266     QMetaObject::activate(this, &staticMetaObject, 5, _a);
267 }
```

Here is the caller graph for this function:



12.22.3.37 KeyMap::Model Kb::model ( ) const [inline]

Definition at line 22 of file kb.h.

References _model.

```
22 { return _model; }
```

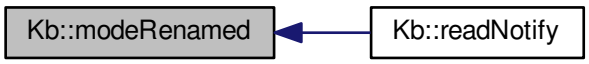
12.22.3.38 void Kb::modeRenamed ( ) [signal]

Definition at line 251 of file moc_kb.cpp.

Referenced by readNotify().

```
252 {
253     QMetaObject::activate(this, &staticMetaObject, 3, Q_NULLPTR);
254 }
```

Here is the caller graph for this function:



12.22.3.39 static bool Kb::mouseAccel ( ) [inline], [static]

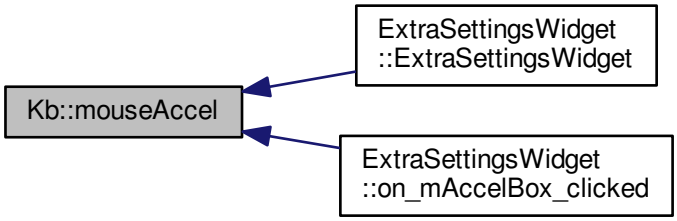
Definition at line 39 of file kb.h.

References _mouseAccel.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and ExtraSettingsWidget::on_mAccelBox_clicked().

```
39 { return _mouseAccel; }
```

Here is the caller graph for this function:



12.22.3.40 void Kb::mouseAccel (bool *newAccel*) [static]

Definition at line 215 of file kb.cpp.

References _mouseAccel, activeDevices, and cmd.

```
215                                 {
216     if(newAccel == _mouseAccel)
217         return;
218     _mouseAccel = newAccel;
219 #ifdef Q_OS_MACX
220     // Update all devices
221     foreach(Kb* kb, activeDevices){
222         kb->cmd.write(QString("accel %1\n").arg(QString(newAccel ? "on" : "off")).toLatin1());
223         kb->cmd.flush();
224     }
225 #endif
226 }
```

12.22.3.41 bool Kb::needsSave ( ) const

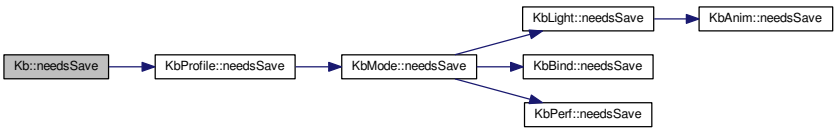
Definition at line 344 of file kb.cpp.

References _needsSave, _profiles, and KbProfile::needsSave().

Referenced by autoSave().

```
344                           {
345     if(_needsSave)
346         return true;
347     foreach(const KbProfile* profile, _profiles){
348         if(profile->needsSave())
349             return true;
350     }
351     return false;
352 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.22.3.42 KbMode* Kb::newMode ( ) [inline]

Definition at line 79 of file kb.h.

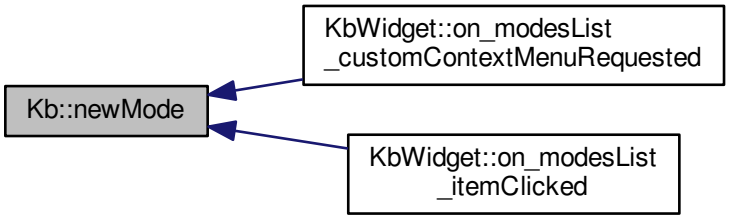
References getKeyMap().

Referenced by KbWidget::on_modesList_customContextMenuRequested(), and KbWidget::on_modesList_itemClicked().

```
79 { return new KbMode(this, getKeyMap()); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.22.3.43 KbMode* Kb::newMode (KbMode * *other*) [inline]

Definition at line 80 of file kb.h.

References getKeyMap().

```
80 { return new KbMode(this, getKeyMap(), *other); }
```

Here is the call graph for this function:

12.22.3.44 KbProfile* Kb::newProfile ( ) [inline]

Definition at line 77 of file kb.h.

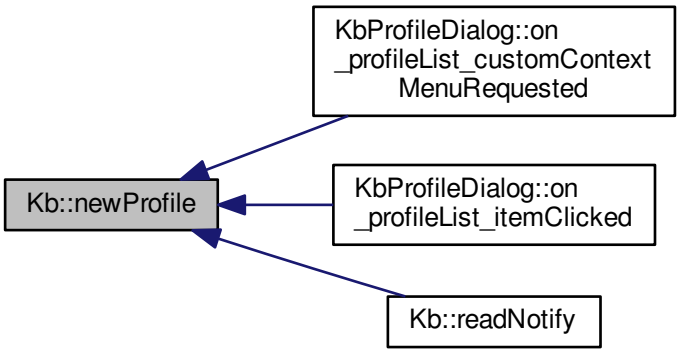
References getKeyMap().

Referenced by KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_-itemClicked(), and readNotify().

```
77 { return new KbProfile(this, getKeyMap()); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.22.3.45 KbProfile* Kb::newProfile (KbProfile * *other*) [inline]

Definition at line 78 of file kb.h.

References getKeyMap().

```
78 { return new KbProfile(this, getKeyMap(), *other); }
```

Here is the call graph for this function:

12.22.3.46 void Kb::profileAdded ( ) [signal]

Definition at line 239 of file moc_kb.cpp.

Referenced by load(), and readNotify().

```
240 {
241     QMetaObject::activate(this, &staticMetaObject, 1, Q_NULLPTR);
242 }
```

Here is the caller graph for this function:

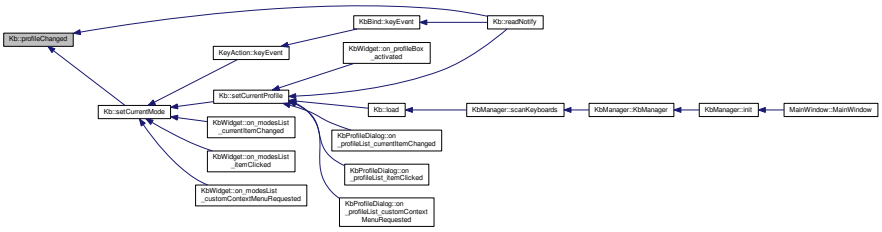
12.22.3.47 void Kb::profileChanged ( ) [signal]

Definition at line 257 of file moc_kb.cpp.

Referenced by readNotify(), and setCurrentMode().

```
258 {
259     QMetaObject::activate(this, &staticMetaObject, 4, Q_NULLPTR);
260 }
```

Here is the caller graph for this function:



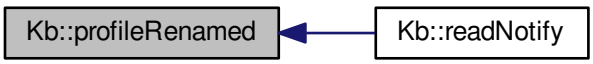
12.22.3.48 void Kb::profileRenamed ( ) [signal]

Definition at line 245 of file moc_kb.cpp.

Referenced by readNotify().

```
246 {
247     QMetaObject::activate(this, &staticMetaObject, 2, Q_NULLPTR);
248 }
```

Here is the caller graph for this function:



12.22.3.49 const QList<KbProfile*>& Kb::profiles ( ) const [inline]

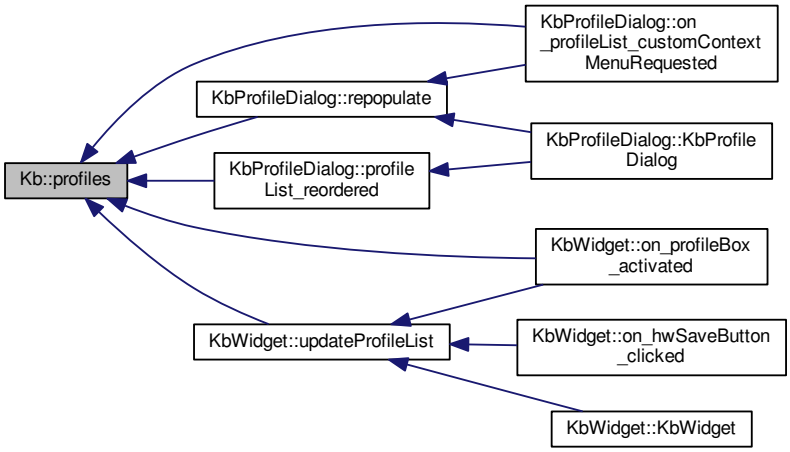
Definition at line 58 of file kb.h.

References _profiles.

Referenced by KbWidget::on_profileBox_activated(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::profileList_reordered(), KbProfileDialog::repopulate(), and KbWidget::updateProfileList().

```
58 { return _profiles; }
```

Here is the caller graph for this function:



12.22.3.50 void Kb::profiles (const QList< KbProfile * > & *newProfiles*) [inline]

Definition at line 59 of file kb.h.

References _needsSave, and _profiles.

```
59 { _needsSave = true; _profiles = newProfiles; }
```

12.22.3.51 void Kb::readNotify (QString *line*) [private], [slot]

Definition at line 471 of file kb.cpp.

References _currentMode, _currentProfile, _hwProfile, _needsSave, _profiles, KbPerf::angleSnap(), KbLight::animKeypress(), KbProfile::append(), KbPerf::baseDpiIdx(), KbMode::bind(), KbLight::color(), KbPerf::dpi(), KbPerf::DPI_COUNT, KbPerf::dpiColor(), KbPerf::dpiEnabled(), fwUpdateFinished(), fwUpdateProgress(), fwUpdPath, getKeyMap(), UsbId::guid, hwLoading, HWMODE_MAX, hwModeCount, UsbId::hwModifiedString(), hwProfile(), KbProfile::id(), KbMode::id(), KbProfile::indexOf(), isMouse(), iState, KbBind::keyEvent(), KbPerf::liftHeight(), KbMode::light(), KbProfile::modeCount(), modeRenamed(), KbProfile::modes(), UsbId::modifiedString(), KbProfile::move(), KbProfile::name(), KbMode::name(), newProfile(), notifyNumber, KbMode::perf(), profileAdded(), profileChanged(), profileRenamed(), setCurrentProfile(), KbProfile::setNeedsSave(), KbMode::setNeedsSave(), x, and y.

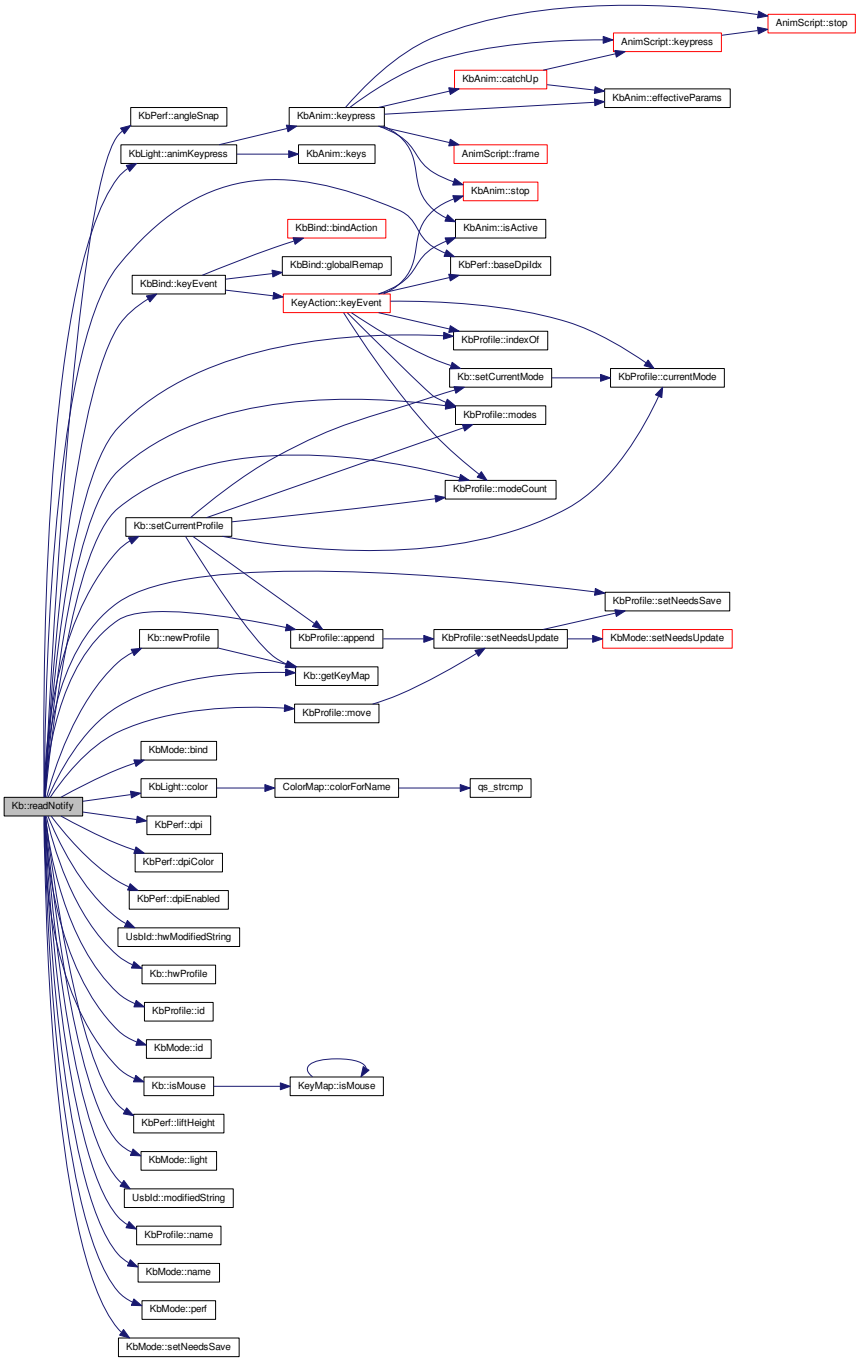
```
471                                {
472     QStringList components = line.trimmed().split(" ");
473     if(components.count() < 2)
474         return;
475     if(components[0] == "key"){
476         // Key event
477         QString key = components[1];
478         if(key.length() < 2)
479             return;
480         QString keyName = key.mid(1);
481         bool keyPressed = (key[0] == '+');
482         KbMode* mode = _currentMode;
483         if(mode){
484             mode->light()->animKeypress(keyName, keyPressed);
485             mode->bind()->keyEvent(keyName, keyPressed);
486         }
487     } else if(components[0] == "i"){
488         // Indicator event
489         QString i = components[1];
490         if(i.length() < 2)
491             return;
492         QString iName = i.mid(1);
493         bool on = (i[0] == '+');
494         if(iName == "num")
495             iState[0] = on;
496         else if(iName == "caps")
497             iState[1] = on;
498         else if(iName == "scroll")
499             iState[2] = on;
500     } else if(components[0] == "hwprofileid"){
501         // Hardware profile ID
502         if(components.count() < 3)
503             return;
504         // Find the hardware profile in the list of profiles
505         QString guid = components[1];
506         QString modified = components[2];
507         KbProfile* newProfile = 0;
508         foreach(KbProfile* profile, _profiles){
509             if(profile->id().guid == guid){
510                 newProfile = profile;
511                 break;
512             }
513         }
514         // If it wasn't found, create it
515         if(!newProfile){
516             newProfile = new KbProfile(this, getKeyMap(), guid, modified);
517             hwLoading[0] = true;
518             cmd.write(QString("@%1 get :hwprofilename\n").arg(notifyNumber).toLatin1());
519             cmd.flush();
520         } else {
521             // If it's been updated, fetch its name
522             if(newProfile->id().hwModifiedString() != modified){
523                 newProfile->id().modifiedString(modified);
524                 newProfile->id().hwModifiedString(modified);
525                 newProfile->setNeedsSave();
```

```
526                 if(hwLoading[0]){
527                     cmd.write(QString("@%1 get :hwprofilename\n").arg(
    notifyNumber).toLatin1());
528                     cmd.flush();
529                 }
530             } else {
531                 hwLoading[0] = false;
532             }
533         }
534         hwProfile(newProfile);
535         emit profileAdded();
536         if(_hwProfile == _currentProfile)
537             emit profileChanged();
538     } else if(components[0] == "hwprofilename"){
539         // Hardware profile name
540         QString name = QUrl::fromPercentEncoding(components[1].toUtf8());
541         if(!_hwProfile || !hwLoading[0])
542             return;
543         QString oldName = _hwProfile->name();
544         if(!(oldName.length() >= name.length() && oldName.left(name.length()) == name)){
545             // Don't change the name if it's a truncated version of what we already have
546             _hwProfile->name(name);
547             emit profileRenamed();
548         }
549     } else if(components[0] == "mode"){
550         // Mode-specific data
551         if(components.count() < 4)
552             return;
553         int mode = components[1].toInt() - 1;
554         if(components[2] == "hwid"){
555             if(components.count() < 5 || mode >= HWMODE_MAX || !
    _hwProfile)
556                 return;
557             // Hardware mode ID
558             QString guid = components[3];
559             QString modified = components[4];
560             // Look for this mode in the hardware profile
561             KbMode* hwMode = 0;
562             bool isUpdated = false;
563             foreach(KbMode* kbMode, _hwProfile->modes()){
564                 if(kbMode->id().guid == guid){
565                     hwMode = kbMode;
566                     if(kbMode->id().hwModifiedString() != modified){
567                         // Update modification time
568                         hwMode->id().modifiedString(modified);
569                         hwMode->id().hwModifiedString(modified);
570                         hwMode->setNeedsSave();
571                         isUpdated = true;
572                     } else {
573                         hwLoading[mode + 1] = false;
574                     }
575                     break;
576                 }
577             }
578             // If it wasn't found, add it
579             if(!hwMode){
580                 isUpdated = true;
581                 hwMode = new KbMode(this, getKeyMap(), guid, modified);
582                 _hwProfile->append(hwMode);
583                 // If the hardware profile now contains enough modes to be added to the list, do so
584                 if(!_profiles.contains(_hwProfile) &&
    _hwProfile->modeCount() >= hwModeCount){
585                     _profiles.append(_hwProfile);
586                     _needsSave = true;
587                     emit profileAdded();
588                     if(!_currentProfile)
589                         setCurrentProfile(_hwProfile);
590                 }
591             }
592             if(hwLoading[mode + 1] && isUpdated){
593                 // If the mode isn't in the right place, move it
594                 int index = _hwProfile->indexOf(hwMode);
595                 if(mode < _hwProfile->modeCount() && index != mode)
596                     _hwProfile->move(index, mode);
597                 // Fetch the updated data
598                 cmd.write(QString("@%1 mode %2 get :hwname :hwrgb").arg(
    notifyNumber).arg(mode + 1).toLatin1());
599                 if(isMouse())
600                     cmd.write(" :hwdpi :hwdpisel :hwlift :hwsnap");
601                 cmd.write("\n");
602                 cmd.flush();
603             }
604         } else if(components[2] == "hwname"){
605             // Mode name - update list
606             if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >=
    HWMODE_MAX || !hwLoading[mode + 1])
607                 return;
```

```
608             KbMode* hwMode = _hwProfile->modes()[mode];
609             QString name = QUrl::fromPercentEncoding(components[3].toUtf8());
610             QString oldName = hwMode->name();
611             if(!(oldName.length() >= name.length() && oldName.left(name.length()) == name)){
612                 // Don't change the name if it's a truncated version of what we already have
613                 hwMode->name(name);
614                 if(_hwProfile == _currentProfile)
615                     emit modeRenamed();
616             }
617         } else if(components[2] == "hwrgb"){
618             // RGB - set mode lighting
619             if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >=
    HWMODE_MAX || !hwLoading[mode + 1])
620                 return;
621             KbMode* kbmode = _hwProfile->modes()[mode];
622             KbLight* light = kbmode->light();
623             // Scan the input for colors
624             QColor lightColor = QColor();
625             for(int i = 3; i < components.count(); i++){
626                 QString comp = components[i];
627                 if(comp.indexOf(":") < 0){
628                     // No ":" - single hex constant
629                     bool ok;
630                     int rgb = comp.toInt(&ok, 16);
631                     if(ok)
632                         light->color(QColor::fromRgb((QRgb)rgb));
633                 } else {
634                     // List of keys ("a,b:xxxxxx"). Parse color first
635                     QStringList set = comp.split(":");
636                     bool ok;
637                     int rgb = set[1].toInt(&ok, 16);
638                     if(ok){
639                         QColor color = QColor::fromRgb((QRgb)rgb);
640                         // Parse keys
641                         QStringList keys = set[0].split(",");
642                         foreach(QString key, keys){
643                             if(key == "light")
644                                 // Extrapolate the Light key to the M-keys and Lock key, since those will
    be set to black on hwsave
645                                 lightColor = color;
646                             if(key.startsWith("dpi") && key.length() > 3){
647                                 // DPI levels go to the KbPerf object instead of KbLight
648                                 bool ok = false;
649                                 int index = key.mid(3).toInt(&ok);
650                                 if(ok)
651                                     kbmode->perf()->dpiColor(index, color);
652                                 continue;
653                             }
654                             light->color(key, color);
655                         }
656                     }
657                 }
658             }
659             if(lightColor.isValid()){
660                 light->color("mr", lightColor);
661                 light->color("m1", lightColor);
662                 light->color("m2", lightColor);
663                 light->color("m3", lightColor);
664                 light->color("lock", lightColor);
665             }
666         } else if(components[2] == "hwdpi"){
667             // DPI settings
668             if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >=
    HWMODE_MAX || !hwLoading[mode + 1])
669                 return;
670             KbPerf* perf = _hwProfile->modes()[mode]->perf();
671             // Read the rest of the line as stage:x,y
672             foreach(QString comp, components.mid(3)){
673                 QStringList dpi = comp.split(':');
674                 if(dpi.length() != 2)
675                     continue;
676                 QStringList xy = dpi[1].split(',');
677                 int x, y;
678                 bool off = false;
679                 if(xy.length() < 2){
680                     // If the right side only has one parameter, set both X and Y
681                     if(xy[0] == "off")
682                         off = true;
683                     else
684                         x = y = xy[0].toInt();
685                 } else {
686                     x = xy[0].toInt();
687                     y = xy[1].toInt();
688                 }
689                 // Set DPI for this stage
690                 int index = dpi[0].toInt();
691                 if(off){
```

```
692                     perf->dpiEnabled(index, false);
693                     // If all DPIs have been disabled, turn them back on
694                     bool allOff = true;
695                     for(int i = 1; i < KbPerf::DPI_COUNT; i++){
696                         if(perf->dpiEnabled(i)){
697                             allOff = false;
698                             break;
699                         }
700                     }
701                     if(allOff){
702                         for(int i = 1; i < KbPerf::DPI_COUNT; i++)
703                             perf->dpiEnabled(i, true);
704                     }
705                 } else {
706                     perf->dpiEnabled(index, true);
707                     perf->dpi(index, QPoint(x, y));
708                 }
709             }
710         } else if(components[2] == "hwdpisel"){
711             // Hardware DPI selection (0...5)
712             if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >=
    HWMODE_MAX || !hwLoading[mode + 1])
713                 return;
714             KbPerf* perf = _hwProfile->modes()[mode]->perf();
715             int idx = components[3].toInt();
716             if(idx < 1)
717                 idx = 1;
718             if(idx >= KbPerf::DPI_COUNT)
719                 idx = KbPerf::DPI_COUNT - 1;
720             perf->baseDpiIdx(idx);
721         } else if(components[2] == "hwlift"){
722             // Mouse lift height (1...5)
723             if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >=
    HWMODE_MAX || !hwLoading[mode + 1])
724                 return;
725             KbPerf* perf = _hwProfile->modes()[mode]->perf();
726             perf->liftHeight((KbPerf::height)components[3].toInt());
727         } else if(components[3] == "hwsnap"){
728             // Mouse angle snapping ("on" or "off")
729             if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >=
    HWMODE_MAX || !hwLoading[mode + 1])
730                 return;
731             KbPerf* perf = _hwProfile->modes()[mode]->perf();
732             perf->angleSnap(components[3] == "on");
733         }
734     } else if(components[0] == "fwupdate"){
735         // Firmware update progress
736         if(components.count() < 3)
737             return;
738         // Make sure path is the same
739         if(components[1] != fwUpdPath)
740             return;
741         QString res = components[2];
742         if(res == "invalid" || res == "fail")
743             emit fwUpdateFinished(false);
744         else if(res == "ok")
745             emit fwUpdateFinished(true);
746         else {
747             // "xx/yy" indicates progress
748             if(!res.contains("/"))
749                 return;
750             QStringList numbers = res.split("/");
751             emit fwUpdateProgress(numbers[0].toInt(), numbers[1].toInt());
752         }
753     }
754 }
```

Here is the call graph for this function:



12.22.3.52 void Kb::run ( ) [private]

Definition at line 446 of file kb.cpp.

References notifyPath, notifyPathMutex, and notifyPaths.

```
446             {
447     QFile notify(notifyPath);
448     // Wait a small amount of time for the node to open (100ms)
```

```
449     QThread::usleep(100000);
450     if(!notify.open(QIODevice::ReadOnly)){
451         // If it's still not open, try again before giving up (1s at a time, 10s total)
452         QThread::usleep(900000);
453         for(int i = 1; i < 10; i++){
454             if(notify.open(QIODevice::ReadOnly))
455                 break;
456             QThread::sleep(1);
457         }
458         if(!notify.isOpen())
459             return;
460     }
461     // Read data from notification node
462     QByteArray line;
463     while(notify.isOpen() && (line = notify.readLine()).length() > 0){
464         QString text = QString::fromUtf8(line);
465         metaObject()->invokeMethod(this, "readNotify", Qt::QueuedConnection, Q_ARG(QString, text));
466     }
467     QMutexLocker locker(&notifyPathMutex);
468     notifyPaths.remove(notifyPath);
469 }
```

12.22.3.53 void Kb::save ( )

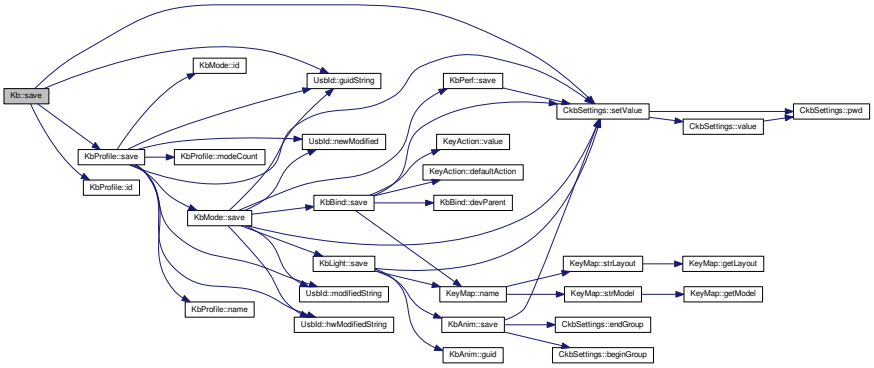
Definition at line 273 of file kb.cpp.

References _currentProfile, _needsSave, _profiles, UsbId::guidString(), KbProfile::id(), prefsPath, KbProfile::save(), and CkbSettings::setValue().

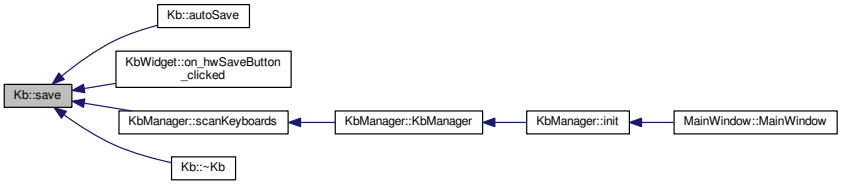
Referenced by autoSave(), KbWidget::on_hwSaveButton_clicked(), KbManager::scanKeyboards(), and ~Kb().

```
273             {
274     if(prefsPath.isEmpty())
275         return;
276     _needsSave = false;
277     CkbSettings settings(prefsPath, true);
278     QString guids, currentGuid;
279     foreach(KbProfile* profile, _profiles){
280         guids.append(" " + profile->id().guidString());
281         if(profile == _currentProfile)
282             currentGuid = profile->id().guidString();
283         profile->save(settings);
284     }
285     settings.setValue("CurrentProfile", currentGuid);
286     settings.setValue("Profiles", guids.trimmed());
287 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.54 static int Kb::scrollSpeed ( ) [inline],[static]

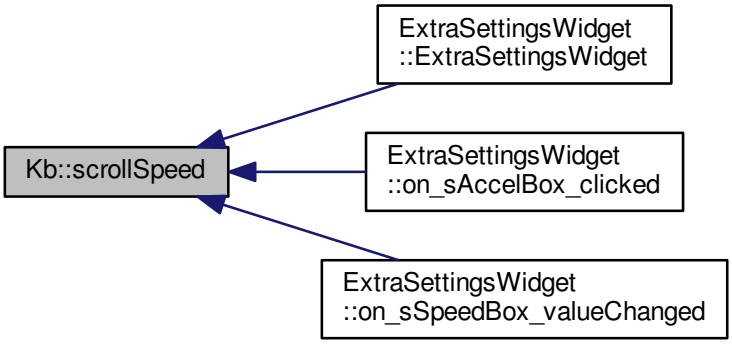
Definition at line 42 of file kb.h.

References _scrollSpeed.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sAccelBox_clicked(), and ExtraSettingsWidget::on_sSpeedBox_valueChanged().

```
42 { return _scrollSpeed; }
```

Here is the caller graph for this function:



12.22.3.55 void Kb::scrollSpeed (int *newSpeed*) [static]

Definition at line 228 of file kb.cpp.

References _scrollSpeed, activeDevices, and cmd.

```
228                                {
229     if(newSpeed == _scrollSpeed)
230         return;
231     _scrollSpeed = newSpeed;
232 #ifdef Q_OS_MACX
233     // Update all devices
234     foreach(Kb* kb, activeDevices){
235         kb->cmd.write(QString("scrollspeed %1\n").arg(newSpeed).toLatin1());
```

```
236         kb->cmd.flush();
237     }
238 #endif
239 }
```

12.22.3.56 void Kb::setCurrentMode (KbProfile * *profile*, KbMode * *mode*, bool *spontaneous* = true)

Definition at line 769 of file kb.cpp.

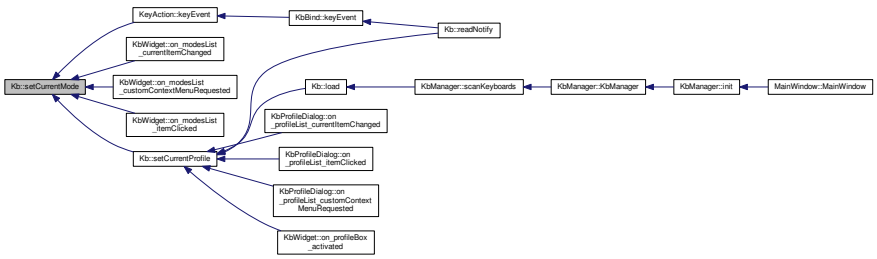
References _currentMode, _currentProfile, _needsSave, KbProfile::currentMode(), modeChanged(), and profileChanged().

Referenced by KeyAction::keyEvent(), KbWidget::on_modesList_currentItemChanged(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::on_modesList_itemClicked(), and setCurrentProfile().

```
769                                                                 {
770     if(_currentProfile != profile){
771         _currentProfile = profile;
772         _needsSave = true;
773         emit profileChanged();
774     }
775     if(_currentMode != mode || _currentProfile->
    currentMode() != mode){
776         _currentProfile->currentMode(_currentMode = mode);
777         _needsSave = true;
778         emit modeChanged(spontaneous);
779     }
780 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.22.3.57 void Kb::setCurrentMode (KbProfile * *profile*, int *index*, bool *spontaneous* = true) [inline]

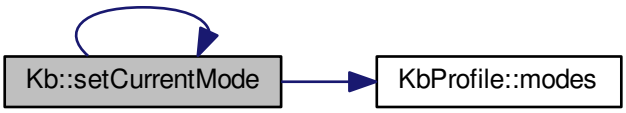
Definition at line 73 of file kb.h.

References KbProfile::modes(), and setCurrentMode().

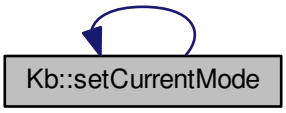
Referenced by setCurrentMode().

```
73 { setCurrentMode(profile, profile->modes()[index], spontaneous); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.58 void Kb::setCurrentMode (KbMode * *mode*, bool *spontaneous* = true) [inline]

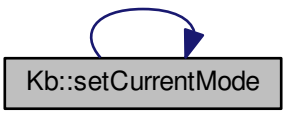
Definition at line 74 of file kb.h.

References _currentProfile, and setCurrentMode().

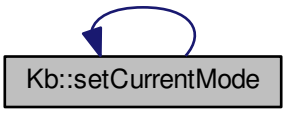
Referenced by setCurrentMode().

```
74 { setCurrentMode(_currentProfile, mode, spontaneous); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.59 void Kb::setCurrentProfile (KbProfile * *profile*, bool *spontaneous* = true)

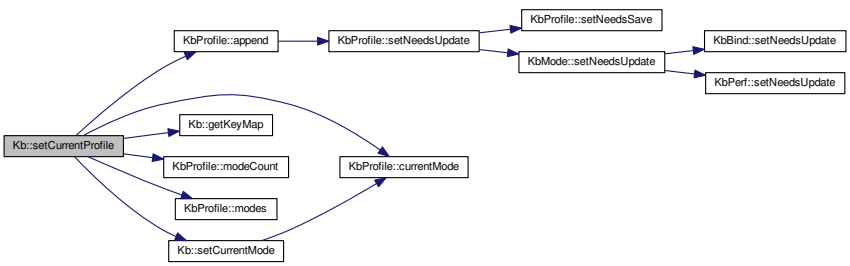
Definition at line 760 of file kb.cpp.

References KbProfile::append(), KbProfile::currentMode(), getKeyMap(), hwModeCount, KbProfile::modeCount(), KbProfile::modes(), and setCurrentMode().

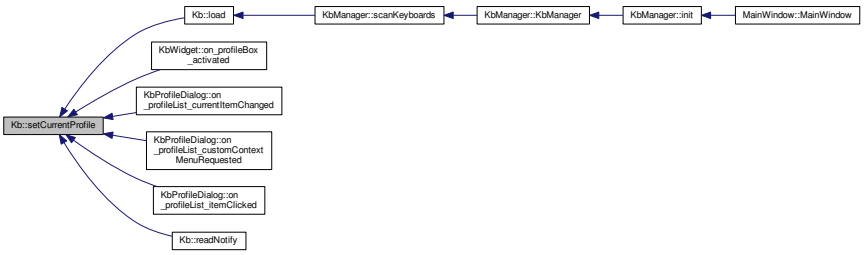
Referenced by load(), KbWidget::on_profileBox_activated(), KbProfileDialog::on_profileList_currentItemChanged(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_itemClicked(), and readNotify().

```
760                                                                 {
761     while(profile->modeCount() < hwModeCount)
762         profile->append(new KbMode(this, getKeyMap()));
763     KbMode* mode = profile->currentMode();
764     if(!mode)
765         profile->currentMode(mode = profile->modes().first());
766     setCurrentMode(profile, mode, spontaneous);
767 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.60 void Kb::updateLayout ( ) [private]

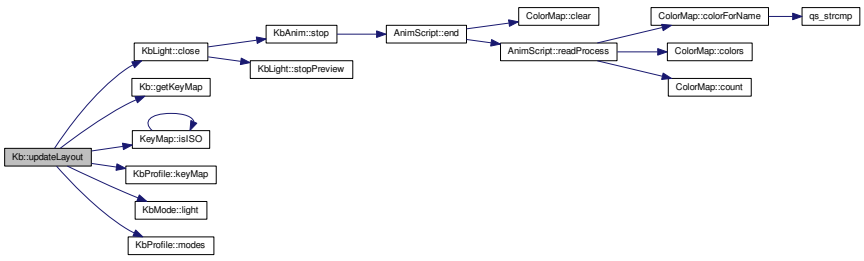
Definition at line 186 of file kb.cpp.

References _currentProfile, _hwProfile, _layout, _profiles, KbLight::close(), getKeyMap(), infoUpdated(), KeyMap::isISO(), KbProfile::keyMap(), KbMode::light(), and KbProfile::modes().

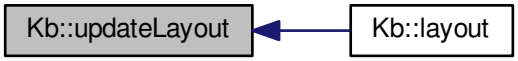
Referenced by layout().

```
186                     {
187 #ifdef Q_OS_MACX
188     // Write ANSI/ISO flag to daemon (OSX only)
189     cmd.write("layout ");
190     cmd.write(KeyMap::isISO(_layout) ? "iso" : "ansi");
191     cmd.write("\n");
192     cmd.flush();
193 #endif
194     foreach(KbProfile* profile, _profiles)
195         profile->keyMap(getKeyMap());
196     if(_hwProfile && !_profiles.contains(_hwProfile))
197         _hwProfile->keyMap(getKeyMap());
198     // Stop all animations as they'll need to be restarted
199     foreach(KbMode* mode, _currentProfile->modes())
200         mode->light()->close();
201     emit infoUpdated();
202 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.3.61 void Kb::writeProfileHeader ( ) [private]

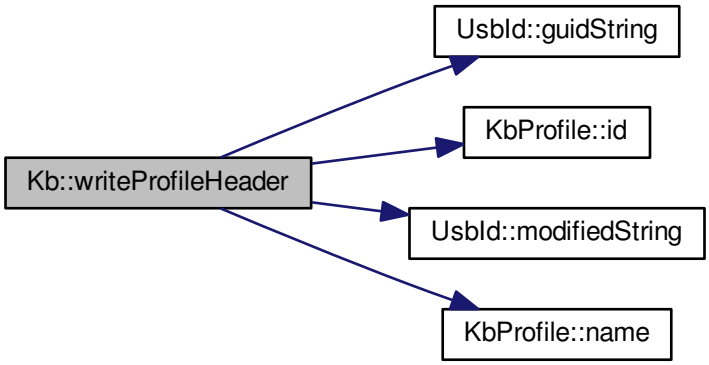
Definition at line 354 of file kb.cpp.

References _currentProfile, UsbId::guidString(), KbProfile::id(), UsbId::modifiedString(), and KbProfile::name().

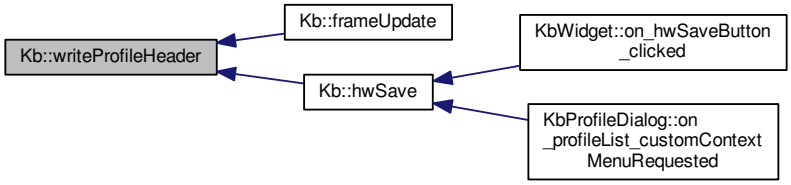
Referenced by frameUpdate(), and hwSave().

```
354                             {
355     cmd.write("eraseprofile");
356     // Write the profile name and ID
357     cmd.write(" profilename ");
358     cmd.write(QUrl::toPercentEncoding(_currentProfile->name()));
359     cmd.write(" profileid ");
360     cmd.write(_currentProfile->id().guidString().toLatin1());
361     cmd.write(" ");
362     cmd.write(_currentProfile->id().modifiedString().toLatin1());
363 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.22.4 Friends And Related Function Documentation

12.22.4.1 friend class KbManager [friend]

Definition at line 142 of file kb.h.

12.22.5 Field Documentation

12.22.5.1 KbMode* Kb::_currentMode [private]

Definition at line 169 of file kb.h.

Referenced by currentBind(), currentLight(), currentMode(), currentPerf(), frameUpdate(), hwSave(), readNotify(), and setCurrentMode().

12.22.5.2 KbProfile* Kb::_currentProfile [private]

Definition at line 167 of file kb.h.

Referenced by currentProfile(), frameUpdate(), hwSave(), readNotify(), save(), setCurrentMode(), updateLayout(), and writeProfileHeader().

12.22.5.3 bool Kb::_delay = false [static], [private]

Definition at line 198 of file kb.h.

Referenced by Kb(), and macroDelay().

12.22.5.4 bool Kb::_dither = false [static], [private]

Definition at line 165 of file kb.h.

Referenced by dither(), and Kb().

12.22.5.5 int Kb::_frameRate = 30 [static], [private]

Definition at line 164 of file kb.h.

Referenced by frameRate(), and Kb().

12.22.5.6 KbProfile* Kb::_hwProfile [private]

Definition at line 183 of file kb.h.

Referenced by deleteHw(), hwProfile(), hwSave(), readNotify(), and updateLayout().

12.22.5.7 KeyMap::Layout Kb::_layout = KeyMap::NO_LAYOUT [static], [private]

Definition at line 161 of file kb.h.

Referenced by getKeyMap(), Kb(), layout(), and updateLayout().

12.22.5.8 KeyMap::Model Kb::_model [private]

Definition at line 171 of file kb.h.

Referenced by model().

12.22.5.9 bool Kb::_mouseAccel = true [static], [private]

Definition at line 165 of file kb.h.

Referenced by Kb(), and mouseAccel().

12.22.5.10 bool Kb::_needsSave [private]

Definition at line 201 of file kb.h.

Referenced by appendProfile(), load(), needsSave(), profiles(), readNotify(), save(), and setCurrentMode().

12.22.5.11 QList<KbProfile*> Kb::_profiles [private]

Definition at line 168 of file kb.h.

Referenced by appendProfile(), find(), indexOf(), load(), needsSave(), profiles(), readNotify(), save(), and updateLayout().

12.22.5.12 int Kb::_scrollSpeed = 0 [static], [private]

Definition at line 164 of file kb.h.

Referenced by Kb(), and scrollSpeed().

12.22.5.13 QFile Kb::cmd [private]

Definition at line 191 of file kb.h.

Referenced by dither(), frameRate(), macroDelay(), mouseAccel(), and scrollSpeed().

12.22.5.14 QString Kb::cmdpath [private]

Definition at line 155 of file kb.h.

Referenced by Kb().

12.22.5.15 QString Kb::devpath [private]

Definition at line 155 of file kb.h.

Referenced by matches().

12.22.5.16 QString Kb::features

Definition at line 18 of file kb.h.

Referenced by MainWindow::checkFwUpdates(), FwUpgradeDialog::exec(), Kb(), KbWidget::KbWidget(), KbWidget::on_fwUpdButton_clicked(), and KbWidget::updateFwButton().

12.22.5.17 QString Kb::firmware

Definition at line 18 of file kb.h.

Referenced by MainWindow::checkFwUpdates(), KbWidget::devUpdate(), FwUpgradeDialog::FwUpgradeDialog(), Kb(), KbWidget::on_fwUpdButton_clicked(), and KbWidget::updateFwButton().

12.22.5.18 QString Kb::fwUpdPath [private]

Definition at line 181 of file kb.h.

Referenced by fwUpdate(), and readNotify().

12.22.5.19 bool Kb::hwLoading[HWMODE_MAX+1] [private]

Definition at line 205 of file kb.h.

Referenced by Kb(), and readNotify().

12.22.5.20 const int Kb::HWMODE_MAX = 3 [static]

Definition at line 50 of file kb.h.

Referenced by KbPerf::applyIndicators(), and readNotify().

12.22.5.21 int Kb::hwModeCount

Definition at line 49 of file kb.h.

Referenced by hwSave(), Kb(), KbWidget::modeIcon(), KbWidget::on_modesList_customContextMenuRequested(), readNotify(), and setCurrentProfile().

12.22.5.22 bool Kb::iState[KbPerf::HW_I_COUNT] [private]

Definition at line 174 of file kb.h.

Referenced by frameUpdate(), Kb(), and readNotify().

12.22.5.23 quint64 Kb::lastAutoSave [private]

Definition at line 178 of file kb.h.

Referenced by autoSave().

12.22.5.24 int Kb::macroNumber [private]

Definition at line 196 of file kb.h.

Referenced by getMacroNumber(), and Kb().

12.22.5.25 QString Kb::macroPath [private]

Definition at line 155 of file kb.h.

Referenced by getMacroPath(), Kb(), and ~Kb().

12.22.5.26 bool Kb::monochrome

Definition at line 19 of file kb.h.

Referenced by frameUpdate(), hwSave(), Kb(), and KbWidget::KbWidget().

12.22.5.27 int Kb::notifyNumber [private]

Definition at line 194 of file kb.h.

Referenced by frameUpdate(), fwUpdate(), hwSave(), Kb(), readNotify(), and ~Kb().

12.22.5.28 QString Kb::notifyPath [private]

Definition at line 155 of file kb.h.

Referenced by Kb(), and run().

12.22.5.29 QString Kb::pollrate

Definition at line 18 of file kb.h.

Referenced by KbWidget::devUpdate(), and Kb().

12.22.5.30 QString Kb::prefsPath [private]

Definition at line 177 of file kb.h.

Referenced by Kb(), load(), and save().

12.22.5.31 KbMode* Kb::prevMode [private]

Definition at line 186 of file kb.h.

Referenced by deletePrevious(), frameUpdate(), and hwSave().

12.22.5.32 KbProfile* Kb::prevProfile [private]

Definition at line 185 of file kb.h.

Referenced by frameUpdate().

12.22.5.33 QString Kb::usbModel

Definition at line 16 of file kb.h.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), Kb(), KbWidget::name(), and MainWindow::showFwUpdateNotification().

12.22.5.34 QString Kb::usbSerial

Definition at line 16 of file kb.h.

Referenced by KbWidget::devUpdate(), Kb(), and matches().

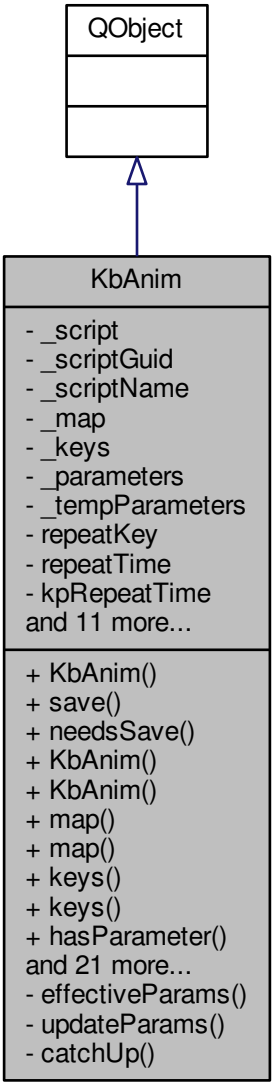
The documentation for this class was generated from the following files:

- src/ckb/kb.h
- src/ckb/kb.cpp
- src/ckb/release/moc_kb.cpp

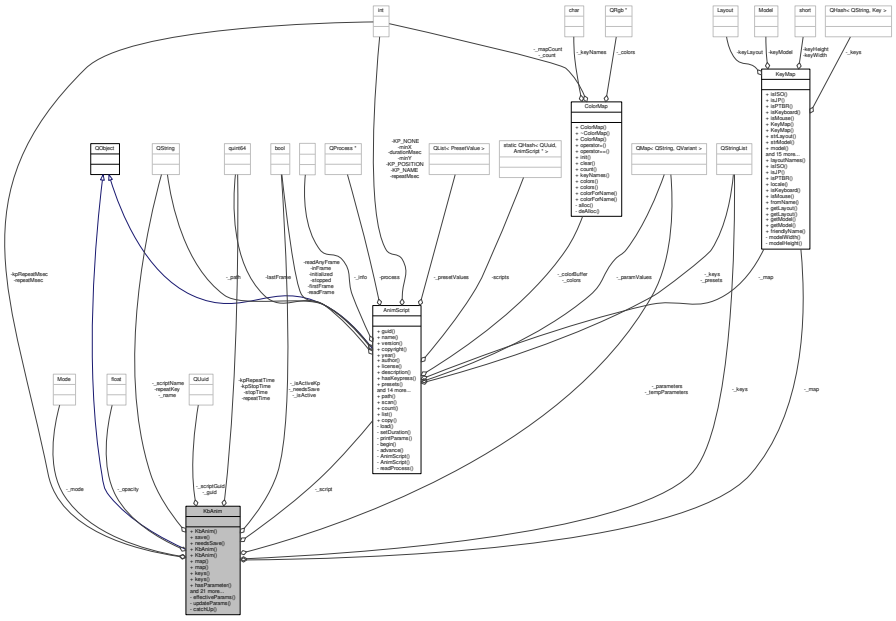
12.23 KbAnim Class Reference

`#include <src/ckb/kbanim.h>`

Inheritance diagram for KbAnim:



Collaboration diagram for KbAnim:



Public Types

- enum Mode {
  Normal, Add, Subtract, Multiply,
  Divide }

Public Member Functions

- KbAnim (QObject *parent, const KeyMap &map, const QUuid id, CkbSettings &settings)
- void save (CkbSettings &settings)
- bool needsSave () const
- KbAnim (QObject *parent, const KeyMap &map, const QString &name, const QStringList &keys, const AnimScript *script)
- KbAnim (QObject *parent, const KeyMap &map, const KbAnim &other)
- const KeyMap & map ()
- void map (const KeyMap &newMap)
- const QStringList & keys ()
- void keys (const QStringList &newKeys)
- bool hasParameter (const QString &name) const
- QVariant parameter (const QString &name) const
- void parameter (const QString &name, const QVariant &value)
- void commitParams ()
- void resetParams ()
- void reInit ()
- void trigger (quint64 timestamp, bool ignoreParameter=false)
- void keypress (const QString &key, bool pressed, quint64 timestamp)
- void stop ()
- bool isActive () const
- bool isRunning () const
- void blend (ColorMap &animMap, quint64 timestamp)
- const QUuid & guid () const

- void newId ()
- const QString & name () const
- void name (const QString &newName)
- float opacity () const
- void opacity (float newOpacity)
- Mode mode () const
- void mode (Mode newMode)
- const AnimScript * script () const
- const QString & scriptName () const

Private Member Functions

- QMap< QString, QVariant > effectiveParams ()
- void updateParams ()
- void catchUp (quint64 timestamp)

Private Attributes

- AnimScript * _script
- QUuid _scriptGuid
- QString _scriptName
- KeyMap _map
- QStringList _keys
- QMap< QString, QVariant > _parameters
- QMap< QString, QVariant > _tempParameters
- QString repeatKey
- quint64 repeatTime
- quint64 kpRepeatTime
- quint64 stopTime
- quint64 kpStopTime
- int repeatMsec
- int kpRepeatMsec
- QUuid _guid
- QString _name
- float _opacity
- Mode _mode
- bool _isActive
- bool _isActiveKp
- bool _needsSave

12.23.1 Detailed Description

Definition at line 11 of file kbanim.h.

12.23.2 Member Enumeration Documentation

12.23.2.1 enum KbAnim::Mode

Enumerator

Normal

Add

Subtract

Multiply

Divide

Definition at line 17 of file kbanim.h.

```
17              {
18         Normal,
19         Add,
20         Subtract,
21         Multiply,
22         Divide
23     };
```

12.23.3 Constructor & Destructor Documentation

12.23.3.1 KbAnim::KbAnim (QObject * *parent,* const KeyMap & *map,* const QUuid *id,* CkbSettings & *settings*)

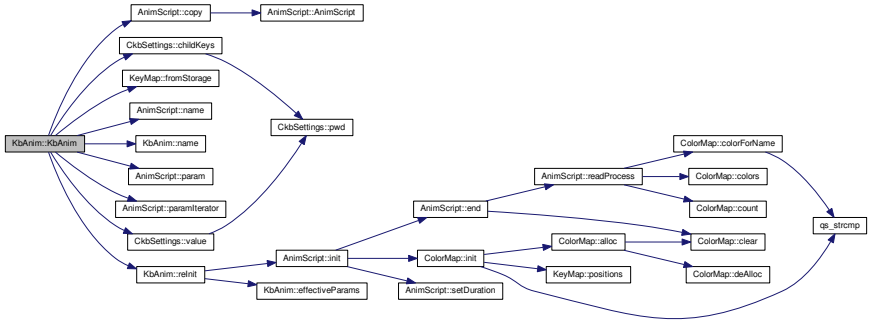
Definition at line 8 of file kbanim.cpp.

References _guid, _keys, _map, _mode, _name, _opacity, _parameters, _script, _scriptGuid, _scriptName, CkbSettings::childKeys(), AnimScript::copy(), AnimScript::Param::defaultValue, KeyMap::fromStorage(), AnimScript::Param::INVALID, AnimScript::Param::LABEL, AnimScript::Param::name, AnimScript::name(), name(), Normal, AnimScript::param(), AnimScript::paramIterator(), reInit(), AnimScript::Param::type, and CkbSettings::value().

```
8                                                                                                   :
9      QObject(parent), _script(0), _map(map),
10      repeatTime(0), kpRepeatTime(0), stopTime(0),
       kpStopTime(0), repeatMsec(0), kpRepeatMsec(0),
11      _guid(id), _isActive(false), _isActiveKp(false),
       _needsSave(false)
12 {
13     SGroup group(settings, _guid.toString().toUpper());
14     _keys = settings.value("Keys").toStringList();
15     // Convert key list from storage names if needed
16     if(!settings.value("UseRealNames").toBool()){
17         QMutableListIterator<QString> i(_keys);
18         while(i.hasNext()){
19             i.next();
20             QString& key = i.value();
21             key = _map.fromStorage(key);
22         }
23     }
24     _name = settings.value("Name").toString().trimmed();
25     _opacity = settings.value("Opacity").toString().toDouble();
26     if(_opacity < 0.)
27         _opacity = 0.;
28     else if(_opacity > 1.)
29         _opacity = 1.;
30     bool modeOk = false;
31     _mode = (Mode)metaObject()->enumerator(metaObject()->indexOfEnumerator("Mode")).keysToValue(
       settings.value("BlendMode").toString().toLatin1(), &modeOk);
32     if(!modeOk)
33         _mode = Normal;
34     _scriptName = settings.value("ScriptName").toString().trimmed();
35     _scriptGuid = settings.value("ScriptGuid").toString();
36     {
37         SGroup group(settings, "Parameters");
38         foreach(const QString& param, settings.childKeys())
39             _parameters[param.toLower()] = settings.value(param);
40     }
41
42     if(!_scriptGuid.isNull()){
43         _script = AnimScript::copy(this, _scriptGuid);
44         if(_script){
45             // Remove nonexistant parameters
46             foreach(const QString& name, _parameters.keys()){
47                 AnimScript::Param param = _script->param(name);
48                 if(param.type == AnimScript::Param::INVALID || param.
       type == AnimScript::Param::LABEL)
49                     _parameters.remove(name);
50             }
51             // Add defaults for unset parameters
52             QListIterator<AnimScript::Param> i = _script->paramIterator();
53             while(i.hasNext()){
54                 AnimScript::Param param = i.next();
55                 if(!_parameters.contains(param.name) && param.
       type != AnimScript::Param::LABEL)
```

```
56                      _parameters[param.name] = param.defaultValue;
57              }
58              _scriptName = _script->name();
59              reInit();
60          }
61      }
62  }
```

Here is the call graph for this function:



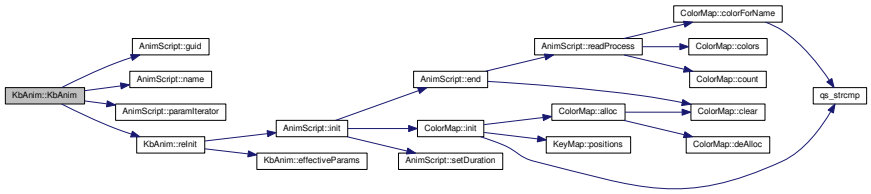
12.23.3.2 KbAnim::KbAnim (QObject * *parent,* const KeyMap & *map,* const QString & *name,* const QStringList & *keys,* const AnimScript * *script*)

Definition at line 84 of file kbanim.cpp.

References _parameters, _script, _scriptGuid, _scriptName, AnimScript::Param::defaultValue, AnimScript::guid(), AnimScript::Param::LABEL, AnimScript::Param::name, AnimScript::name(), AnimScript::paramIterator(), reInit(), and AnimScript::Param::type.

```
84
                    :
85      QObject(parent),
86      _script(AnimScript::copy(this, script->guid())),
        _map(map), _keys(keys),
87      repeatTime(0), kpRepeatTime(0), stopTime(0),
        kpStopTime(0), repeatMsec(0), kpRepeatMsec(0),
88      _guid(QUuid::createUuid()), _name(name), _opacity(1.),
        _mode(Normal), _isActive(false), _isActiveKp(false),
        _needsSave(true)
89  {
90      if(_script){
91          // Set default parameters
92          QListIterator<AnimScript::Param> i = _script->paramIterator();
93          while(i.hasNext()){
94              AnimScript::Param param = i.next();
95              if(param.type != AnimScript::Param::LABEL)
96                  _parameters[param.name] = param.defaultValue;
97          }
98          _scriptGuid = script->guid();
99          _scriptName = script->name();
100         reInit();
101     }
102 }
```

Here is the call graph for this function:



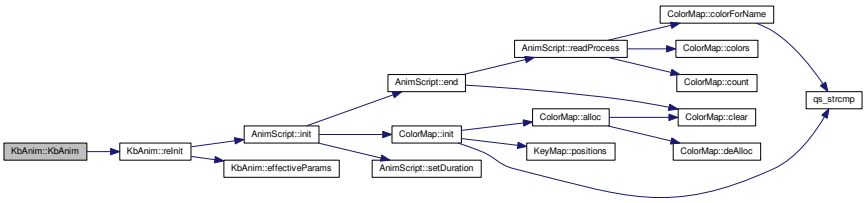
12.23.3.3 KbAnim::KbAnim (QObject * *parent,* const KeyMap & *map,* const KbAnim & *other*)

Definition at line 104 of file kbanim.cpp.

References reInit().

```
104                                                                    :
105     QObject(parent),
106     _script(AnimScript::copy(this, other.script()->
      guid())), _scriptGuid(_script->guid()), _scriptName(
      _script->name()),
107     _map(map), _keys(other._keys), _parameters(other.
      _parameters),
108     repeatTime(0), kpRepeatTime(0), stopTime(0),
      kpStopTime(0), repeatMsec(0), kpRepeatMsec(0),
109     _guid(other._guid), _name(other._name), _opacity(other.
      _opacity), _mode(other._mode), _isActive(false),
      _isActiveKp(false), _needsSave(true)
110 {
111     reInit();
112 }
```

Here is the call graph for this function:



12.23.4 Member Function Documentation

12.23.4.1 void KbAnim::blend (ColorMap & *animMap,* quint64 *timestamp*)

Definition at line 365 of file kbanim.cpp.

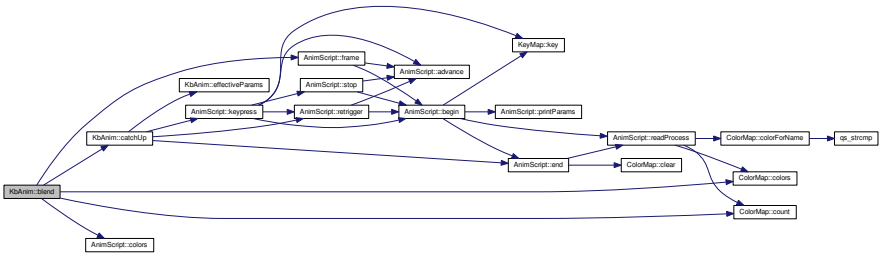
References _mode, _opacity, _script, catchUp(), ColorMap::colors(), AnimScript::colors(), ColorMap::count(), AnimScript::frame(), and functions.

Referenced by KbLight::frameUpdate().

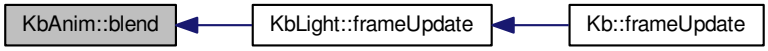
```
365                                                    {
366     if(!_script)
367         return;
368
369     // Fetch the next frame from the script
370     catchUp(timestamp);
```

```
371     _script->frame(timestamp);
372
373     // Blend the script's map with the current map
374     int blendMode = (int)_mode;
375     blendFunc blend = functions[blendMode];
376     float fOpacity = _opacity / 255.f;  // save some math by pre-dividing the 255 for qAlpha
377
378     const ColorMap& scriptMap = _script->colors();
379     int count = animMap.count();
380     if(scriptMap.count() != count){
381         qDebug() << "Script map didn't match base map (" << count << " vs " << scriptMap.
      count() << "). This should never happen.";
382         return;
383     }
384     QRgb* background = animMap.colors();
385     const QRgb* foreground = scriptMap.colors();
386     for(int i = 0; i < count; i++){
387         // Mix the colors in with the color map according to blend mode and alpha
388         QRgb& bg = background[i];
389         QRgb fg = foreground[i];
390         int alpha = qAlpha(fg);
391         if(alpha == 0)
392             continue;
393         if(blendMode == 0){
394             // Blend: normal
395             // This is the most common use case and it requires much less arithmetic
396             if(alpha == 255){
397                 bg = fg;
398             } else {
399                 float r = qRed(bg), g = qGreen(bg), b = qBlue(bg);
400                 float a = alpha * fOpacity;
401                 r = r * (1.f - a) + qRed(fg) * a;
402                 g = g * (1.f - a) + qGreen(fg) * a;
403                 b = b * (1.f - a) + qBlue(fg) * a;
404                 bg = qRgb(round(r), round(g), round(b));
405             }
406         } else {
407             // Use blend function
408             float r = qRed(bg) / 255.f, g = qGreen(bg) / 255.f, b = qBlue(bg) / 255.f;
409             float a = alpha * fOpacity;
410             r = r * (1.f - a) + blend(r, qRed(fg) / 255.f) * a;
411             g = g * (1.f - a) + blend(g, qGreen(fg) / 255.f) * a;
412             b = b * (1.f - a) + blend(b, qBlue(fg) / 255.f) * a;
413             bg = qRgb(round(r * 255.f), round(g * 255.f), round(b * 255.f));
414         }
415     }
416 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.4.2 void KbAnim::catchUp (quint64 *timestamp*) [private]

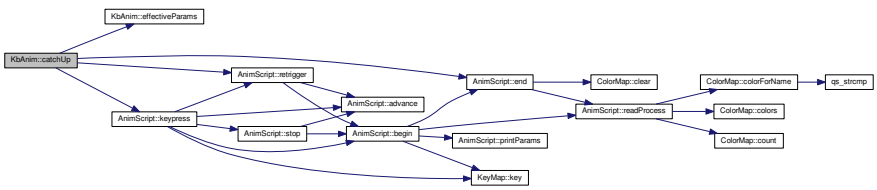
Definition at line 166 of file kbanim.cpp.

References _isActive, _isActiveKp, _script, effectiveParams(), AnimScript::end(), AnimScript::keypress(), kpRepeatMsec, kpRepeatTime, kpStopTime, repeatKey, repeatMsec, repeatTime, AnimScript::retrigger(), and stopTime.

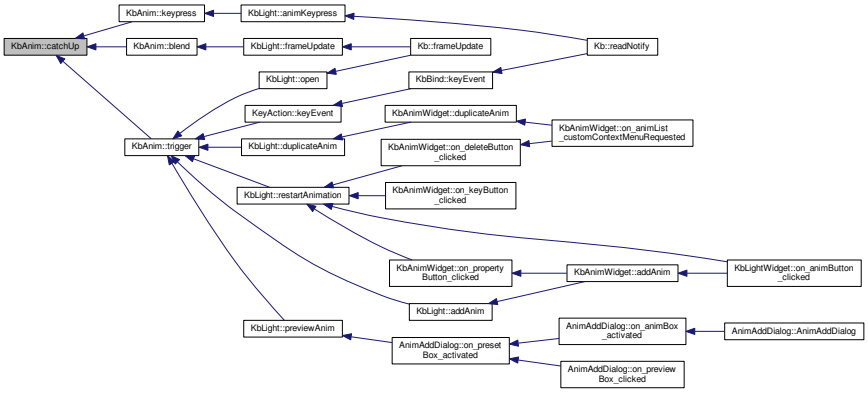
Referenced by blend(), keypress(), and trigger().

```
166                                     {
167     QMap<QString, QVariant> parameters = effectiveParams();
168     // Stop the animation if its time has run out
169     if(stopTime != 0 && timestamp >= stopTime){
170         repeatMsec = repeatTime = 0;
171         if(!parameters.contains("repeat")){
172             // If repeats aren't allowed, stop the animation entirely
173             _script->end();
174             _isActive = false;
175             return;
176         } else
177             // Otherwise, simply stop repeating
178             stopTime = 0;
179     }
180     if(kpStopTime != 0 && timestamp >= kpStopTime){
181         kpRepeatMsec = kpRepeatTime = 0;
182         if(!parameters.contains("kprepeat")){
183             _script->end();
184             _isActiveKp = false;
185             return;
186         } else
187             kpStopTime = 0;
188     }
189
190     // Restart (or start, if there was a delay) the animation if its repeat time is up
191     while(repeatTime > 0 && timestamp >= repeatTime){
192         _script->retrigger(repeatTime);
193         if(repeatMsec <= 0){
194             repeatTime = 0;
195             break;
196         }
197         repeatTime += repeatMsec;
198     }
199     while(kpRepeatTime > 0 && timestamp >= kpRepeatTime){
200         _script->keypress(repeatKey, 1, kpRepeatTime);
201         if(kpRepeatMsec <= 0){
202             kpRepeatTime = 0;
203             break;
204         }
205         kpRepeatTime += kpRepeatMsec;
206     }
207 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.4.3 void KbAnim::commitParams ( )

Definition at line 121 of file kbanim.cpp.

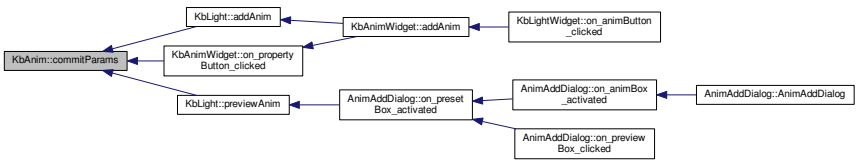
References _needsSave, _parameters, _tempParameters, and effectiveParams().

Referenced by KbLight::addAnim(), KbAnimWidget::on_propertyButton_clicked(), and KbLight::previewAnim().

```
121                          {
122     _needsSave = true;
123     _parameters = effectiveParams();
124     _tempParameters.clear();
125 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.23.4.4 QMap< QString, QVariant > KbAnim::effectiveParams ( ) [private]

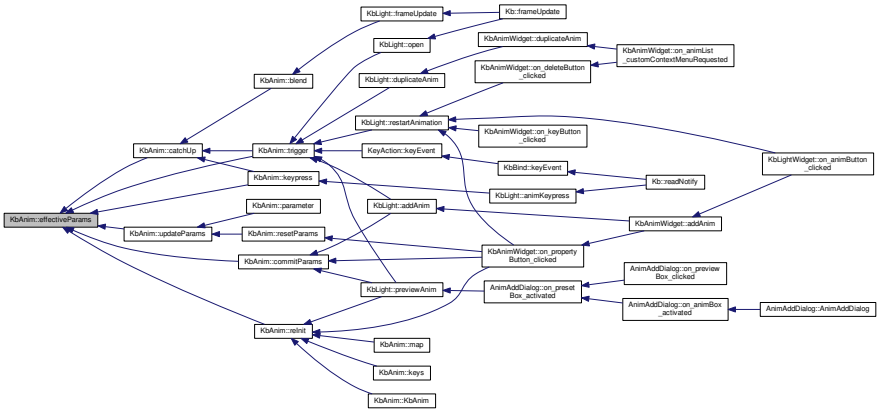
Definition at line 138 of file kbanim.cpp.

References _parameters, and _tempParameters.

Referenced by catchUp(), commitParams(), keypress(), reInit(), trigger(), and updateParams().

```
138                                                {
139     QMap<QString, QVariant> res = _parameters;
140     // Apply all uncommited parameters
141     QMapIterator<QString, QVariant> i(_tempParameters);
142     while(i.hasNext()){
143         i.next();
144         res[i.key()] = i .value();
145     }
146     return res;
147 }
```

Here is the caller graph for this function:



12.23.4.5 const QUuid& KbAnim::guid ( ) const [inline]

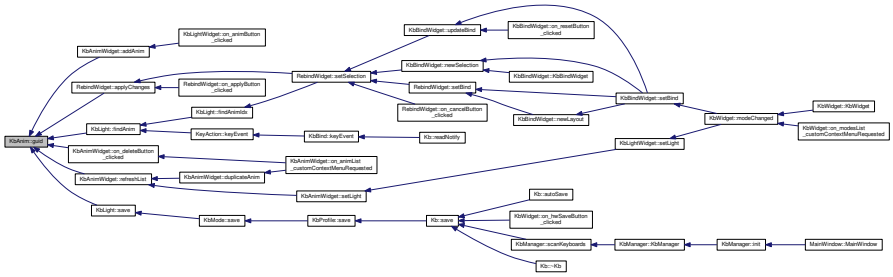
Definition at line 71 of file kbanim.h.

References _guid.

Referenced by KbAnimWidget::addAnim(), RebindWidget::applyChanges(), KbLight::findAnim(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::refreshList(), and KbLight::save().

```
71 { return _guid; }
```

Here is the caller graph for this function:



12.23.4.6 bool KbAnim::hasParameter (const QString & *name*) const [inline]

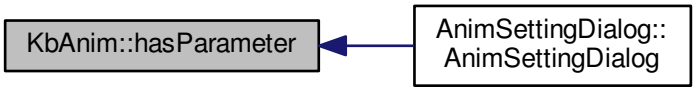
Definition at line 44 of file kbanim.h.

References _parameters.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
44 { return _parameters.contains(name); }
```

Here is the caller graph for this function:



12.23.4.7 bool KbAnim::isActive ( ) const [inline]

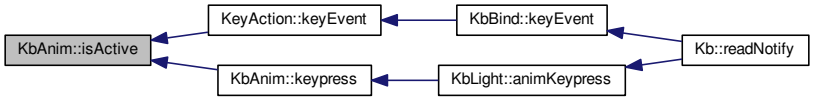
Definition at line 63 of file kbanim.h.

References _isActive, and _isActiveKp.

Referenced by KeyAction::keyEvent(), and keypress().

```
63 { return _isActive || _isActiveKp; }
```

Here is the caller graph for this function:



12.23.4.8 bool KbAnim::isRunning ( ) const

Definition at line 325 of file kbanim.cpp.

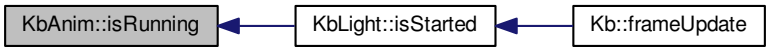
References _script, and AnimScript::hasFrame().

Referenced by KbLight::isStarted().

```
325                             {
326     if(!_script)
327         return true;    // If the script wasn't loaded, pretend it's running anyway so it won't lock up
328     return _script->hasFrame();
329 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.23.4.9 void KbAnim::keypress (const QString & *key,* bool *pressed,* quint64 *timestamp*)

Definition at line 250 of file kbanim.cpp.

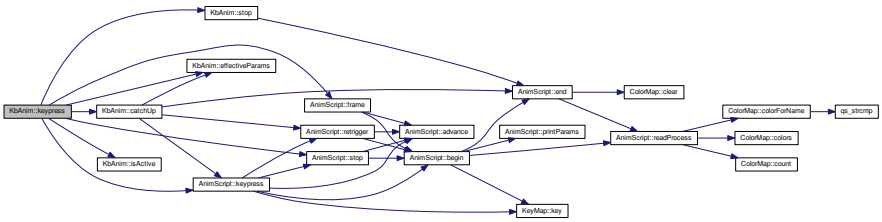
References _isActive, _isActiveKp, _script, catchUp(), effectiveParams(), AnimScript::frame(), isActive(), AnimScript::keypress(), kpRepeatMsec, kpRepeatTime, kpStopTime, repeatKey, repeatMsec, repeatTime, stop(), AnimScript::stop(), and stopTime.

Referenced by KbLight::animKeypress().

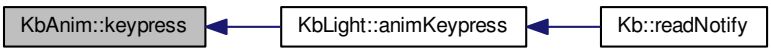
```
250                                                                     {
251     if(!_script)
252         return;
253     QMap<QString, QVariant> parameters = effectiveParams();
254     if(pressed && parameters.value("kpmodestop").toBool()){
255         // If stop on key press is enabled, stop mode-wide animation
256         catchUp(timestamp);
257         _script->stop(timestamp);
258         stopTime = repeatTime = repeatMsec = 0;
259         _isActive = false;
260     } else {
261         if(!parameters.value("kptrigger").toBool())
262             return;
263         catchUp(timestamp);
264     }
265
266     if(pressed){
267         // Key pressed
268         _isActiveKp = true;
269         if(parameters.value("kpmode", 0).toInt() == 2 && isActive())
270             // If mode is start once and a key has already been pressed, do nothing
271             return;
272         int delay = round(parameters.value("kpdelay").toDouble() * 1000.);
273         if(delay > 0){
274             // If delay is enabled, wait to trigger the event
275             timestamp += delay;
276             kpRepeatTime = timestamp;
277         } else
278             _script->keypress(key, pressed, timestamp);
279
280         int repeat = round(parameters.value("kprepeat").toDouble() * 1000.);
281         if(repeat <= 0){
282             // If no repeat allowed, calculate stop time in seconds
283             kpRepeatMsec = -1;
284             double stop = parameters.value("kpstop").toDouble();
285             if(stop <= 0.)
```

```
286                 kpStopTime = 0;
287             else
288                 kpStopTime = timestamp + round(stop * 1000.);
289         } else {
290             // If repeat is allowed, calculate stop time in repetitions
291             kpRepeatMsec = repeat;
292             if(delay <= 0)
293                 kpRepeatTime = timestamp + repeat;
294             int stop = parameters.value("kpstop").toInt();
295             if(stop < 0)
296                 kpStopTime = 0;
297             else
298                 kpStopTime = timestamp + repeat * (1 + stop);
299         }
300         repeatKey = key;
301     } else {
302         // Key released
303         _isActiveKp = false;
304         _script->keypress(key, pressed, timestamp);
305         if(parameters.value("kprelease").toBool())
306             // Stop repeating keypress if "Stop on key release" is enabled
307             kpStopTime = timestamp;
308     }
309     _script->frame(timestamp);
310 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.4.10 const QStringList& KbAnim::keys ( ) [inline]

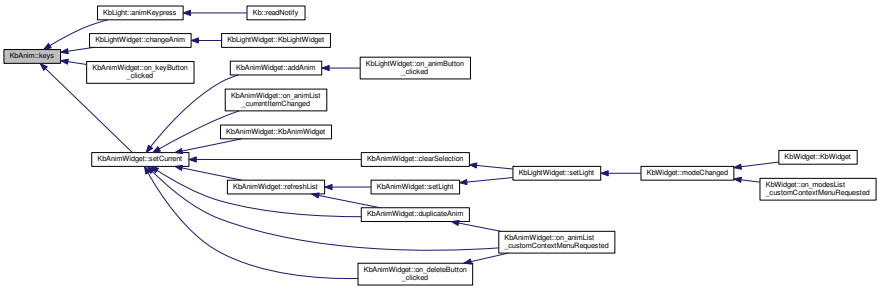
Definition at line 40 of file kbanim.h.

References _keys.

Referenced by KbLight::animKeypress(), KbLightWidget::changeAnim(), KbAnimWidget::on_keyButton_clicked(), and KbAnimWidget::setCurrent().

```
40 { return _keys; }
```

Here is the caller graph for this function:



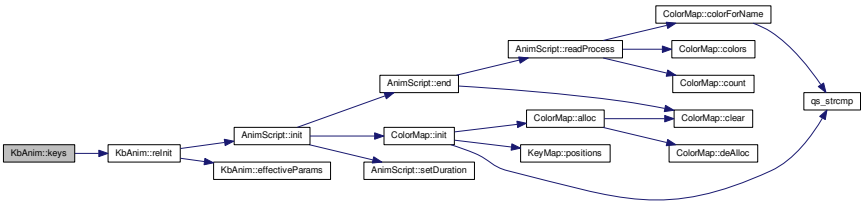
12.23.4.11 void KbAnim::keys (const QStringList & *newKeys*)

Definition at line 161 of file kbanim.cpp.

References _keys, and reInit().

```
161                                          {
162     _keys = newKeys;
163     reInit();
164 }
```

Here is the call graph for this function:



12.23.4.12 const KeyMap& KbAnim::map ( ) [inline]

Definition at line 37 of file kbanim.h.

References _map.

Referenced by KbLight::map().

```
37 { return _map; }
```

Here is the caller graph for this function:

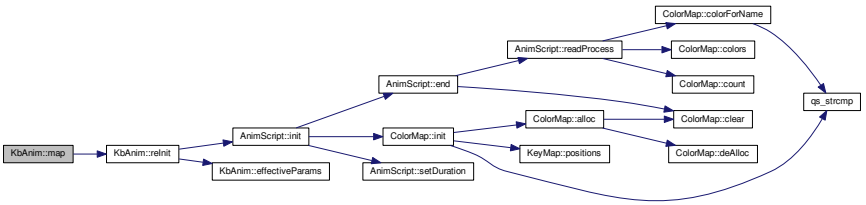
12.23.4.13 void KbAnim::map (const KeyMap & *newMap*)

Definition at line 156 of file kbanim.cpp.

References _map, and reInit().

```
156                                     {
157     _map = newMap;
158     reInit();
159 }
```

Here is the call graph for this function:



12.23.4.14 Mode KbAnim::mode ( ) const [inline]

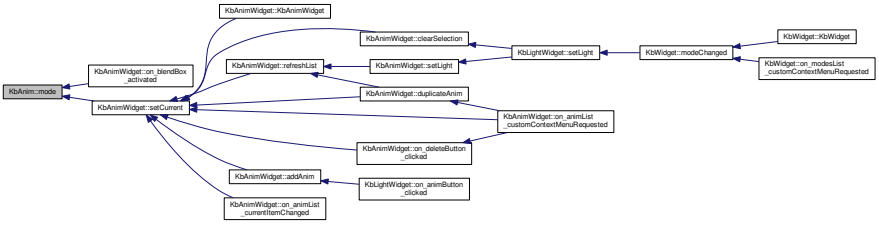
Definition at line 77 of file kbanim.h.

References _mode.

Referenced by KbAnimWidget::on_blendBox_activated(), and KbAnimWidget::setCurrent().

```
77 { return _mode; }
```

Here is the caller graph for this function:



12.23.4.15 void KbAnim::mode (Mode *newMode*) [inline]

Definition at line 78 of file kbanim.h.

References _mode, and _needsSave.

```
78 { _needsSave = true; _mode = newMode; }
```

12.23.4.16 const QString& KbAnim::name ( ) const [inline]

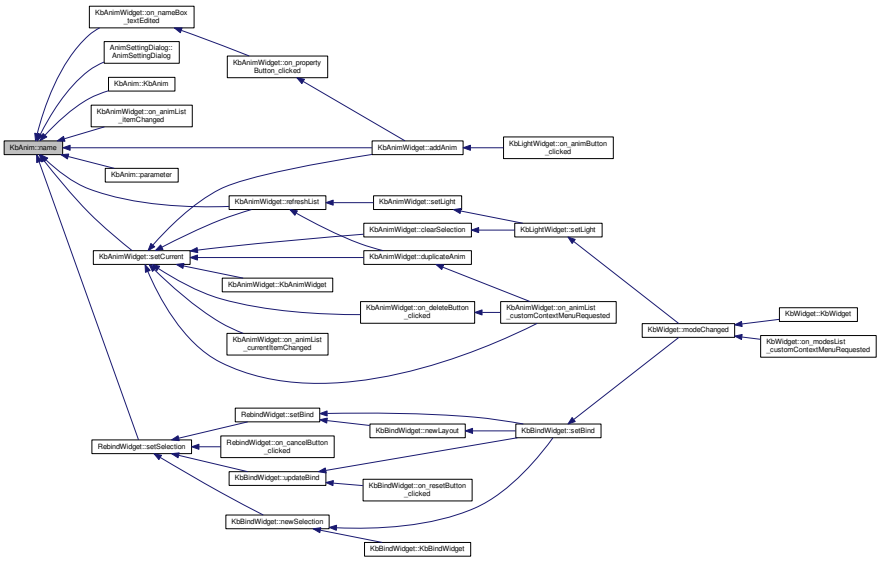
Definition at line 73 of file kbanim.h.

References _name.

Referenced by KbAnimWidget::addAnim(), AnimSettingDialog::AnimSettingDialog(), KbAnim(), KbAnimWidget::on_animList_itemChanged(), KbAnimWidget::on_nameBox_textEdited(), parameter(), KbAnimWidget::refreshList(), KbAnimWidget::setCurrent(), and RebindWidget::setSelection().

```
73 { return _name; }
```

Here is the caller graph for this function:



12.23.4.17 void KbAnim::name (const QString & *newName*) [inline]

Definition at line 74 of file kbanim.h.

References _name, and _needsSave.

```
74 { _needsSave = true; _name = newName; }
```

12.23.4.18 bool KbAnim::needsSave ( ) const [inline]

Definition at line 29 of file kbanim.h.

References _needsSave.

Referenced by KbLight::needsSave().

```
29 { return _needsSave; }
```

Here is the caller graph for this function:

12.23.4.19 void KbAnim::newId ( ) [inline]

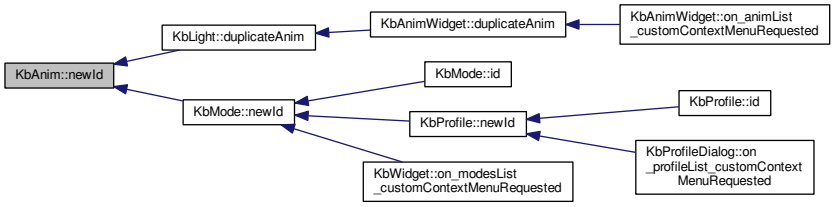
Definition at line 72 of file kbanim.h.

References _guid, and _needsSave.

Referenced by KbLight::duplicateAnim(), and KbMode::newId().

```
72 { _needsSave = true; _guid = QUuid::createUuid(); }
```

Here is the caller graph for this function:



12.23.4.20 float KbAnim::opacity ( ) const [inline]

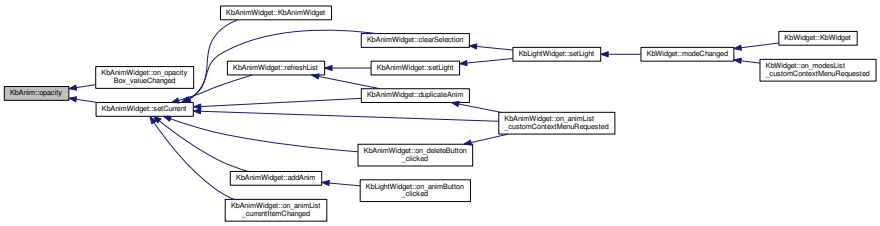
Definition at line 75 of file kbanim.h.

References _opacity.

Referenced by KbAnimWidget::on_opacityBox_valueChanged(), and KbAnimWidget::setCurrent().

```
75 { return _opacity; }
```

Here is the caller graph for this function:



12.23.4.21 void KbAnim::opacity (float *newOpacity*) [inline]

Definition at line 76 of file kbanim.h.

References _needsSave, and _opacity.

```
76 { _needsSave = true; _opacity = newOpacity; }
```

12.23.4.22 QVariant KbAnim::parameter (const QString & *name*) const [inline]

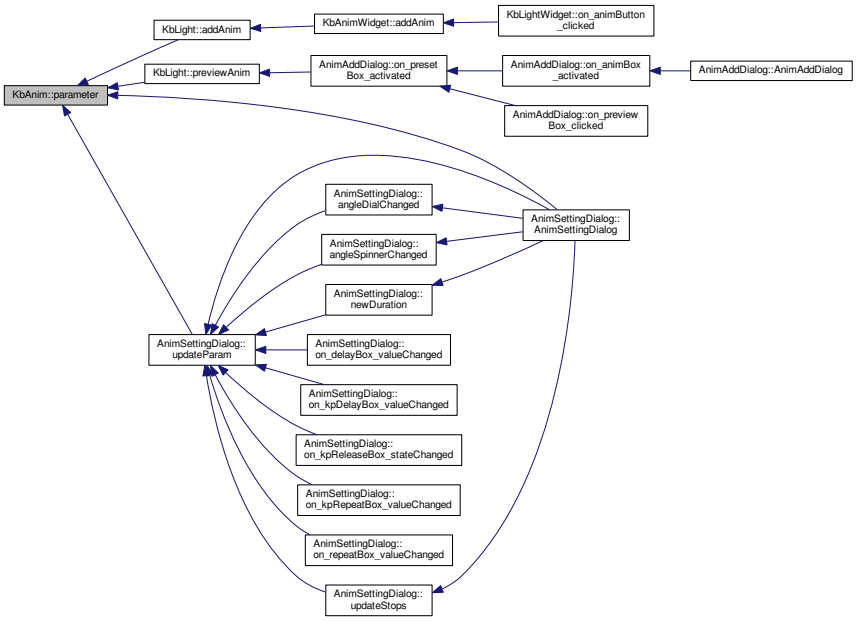
Definition at line 45 of file kbanim.h.

References _parameters.

Referenced by KbLight::addAnim(), AnimSettingDialog::AnimSettingDialog(), KbLight::previewAnim(), and AnimSettingDialog::updateParam().

```
45 { return _parameters.value(name); }
```

Here is the caller graph for this function:



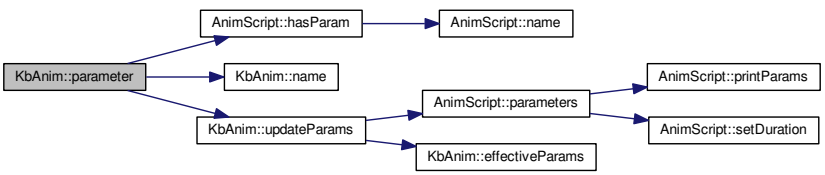
12.23.4.23 void KbAnim::parameter (const QString & *name,* const QVariant & *value*)

Definition at line 114 of file kbanim.cpp.

References _script, _tempParameters, AnimScript::hasParam(), name(), and updateParams().

```
114                                                                 {
115     if(!_script->hasParam(name))
116         return;
117     _tempParameters[name] = value;
118     updateParams();
119 }
```

Here is the call graph for this function:



12.23.4.24 void KbAnim::relnit ( )

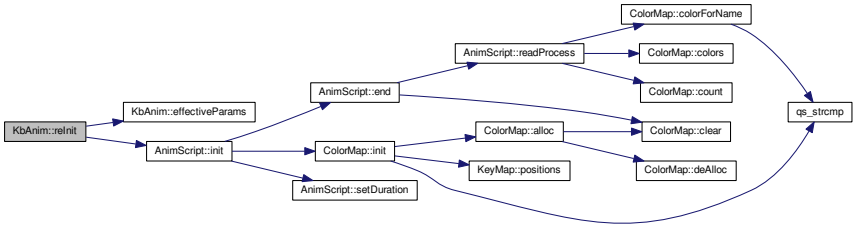
Definition at line 149 of file kbanim.cpp.

References _isActive, _isActiveKp, _keys, _map, _script, effectiveParams(), AnimScript::init(), and repeatKey.

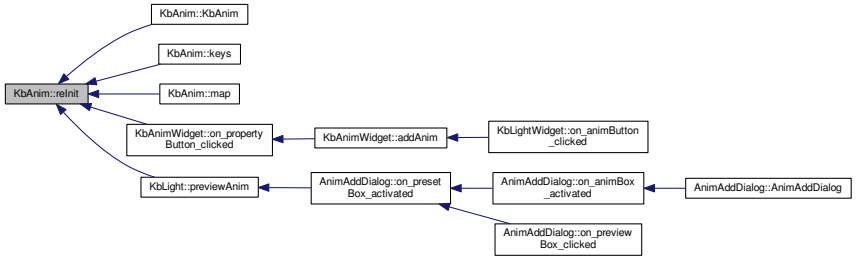
Referenced by KbAnim(), keys(), map(), KbAnimWidget::on_propertyButton_clicked(), and KbLight::previewAnim().

```
149                     {
150     if(_script)
151         _script->init(_map, _keys, effectiveParams());
152     repeatKey = "";
153     _isActive = _isActiveKp = false;
154 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.4.25 void KbAnim::resetParams ( )

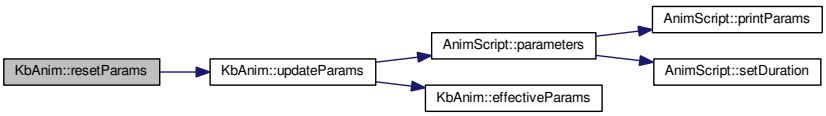
Definition at line 127 of file kbanim.cpp.

References _tempParameters, and updateParams().

Referenced by KbAnimWidget::on_propertyButton_clicked().

```
127                     {
128     _tempParameters.clear();
129     updateParams();
130 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.23.4.26 void KbAnim::save (CkbSettings & *settings*)

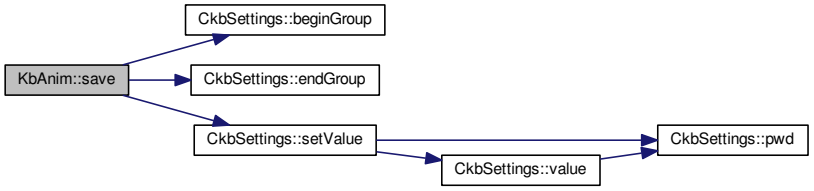
Definition at line 64 of file kbanim.cpp.

References _guid, _keys, _mode, _name, _needsSave, _opacity, _parameters, _scriptGuid, _scriptName, CkbSettings::beginGroup(), CkbSettings::endGroup(), and CkbSettings::setValue().

Referenced by KbLight::save().

```
64                                      {
65     _needsSave = false;
66     settings.beginGroup(_guid.toString().toUpper());
67     settings.setValue("UseRealNames", true);
68     settings.setValue("Keys", _keys);
69     settings.setValue("Name", _name);
70     settings.setValue("Opacity", QString::number(_opacity));
71     settings.setValue("BlendMode", metaObject()->enumerator(metaObject()->indexOfEnumerator("Mode")
       ).valueToKey(_mode));
72     settings.setValue("ScriptName", _scriptName);
73     settings.setValue("ScriptGuid", _scriptGuid.toString().toUpper());
74     settings.beginGroup("Parameters");
75     QMapIterator<QString, QVariant> i(_parameters);
76     while(i.hasNext()){
77         i.next();
78         settings.setValue(i.key(), i.value());
79     }
80     settings.endGroup();
81     settings.endGroup();
82 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.23.4.27 const AnimScript* KbAnim::script ( ) const [inline]

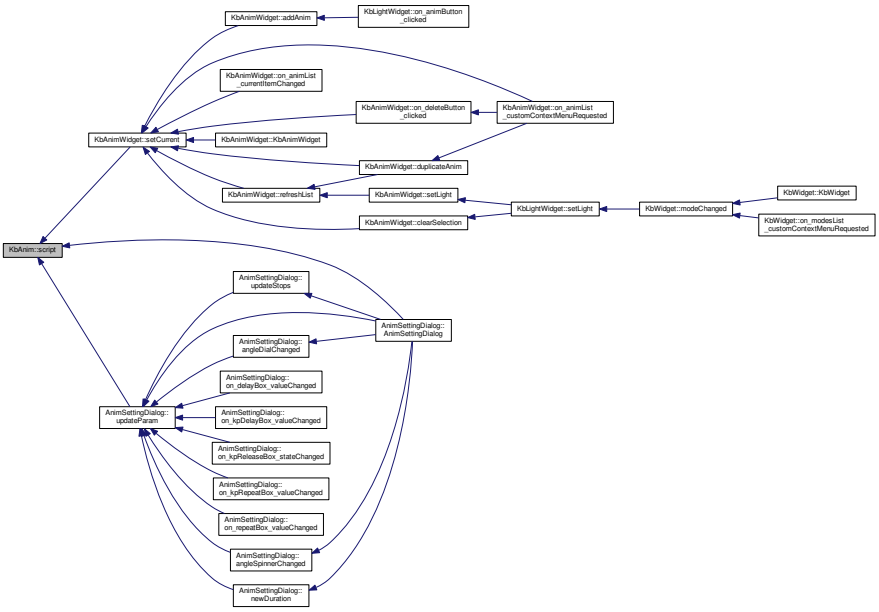
Definition at line 81 of file kbanim.h.

References _script.

Referenced by AnimSettingDialog::AnimSettingDialog(), KbAnimWidget::setCurrent(), and AnimSettingDialog::updateParam().

```
81 { return _script; }
```

Here is the caller graph for this function:



12.23.4.28 const QString& KbAnim::scriptName ( ) const [inline]

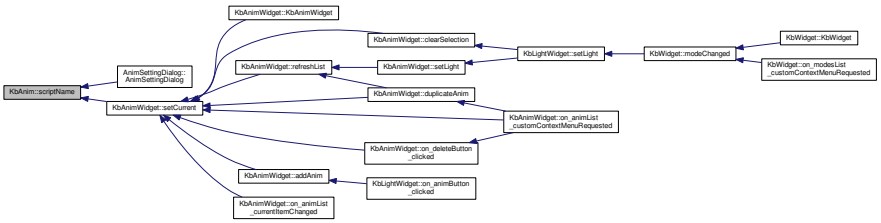
Definition at line 82 of file kbanim.h.

References _scriptName.

Referenced by AnimSettingDialog::AnimSettingDialog(), and KbAnimWidget::setCurrent().

```
82 { return _scriptName; }
```

Here is the caller graph for this function:



12.23.4.29 void KbAnim::stop ( )

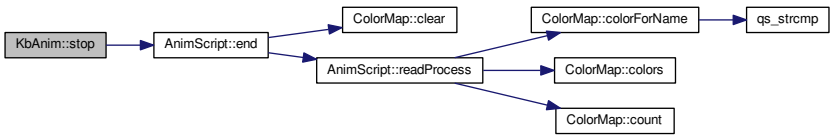
Definition at line 312 of file kbanim.cpp.

References _isActive, _isActiveKp, _script, AnimScript::end(), kpRepeatMsec, kpRepeatTime, kpStopTime, repeatKey, repeatMsec, repeatTime, and stopTime.

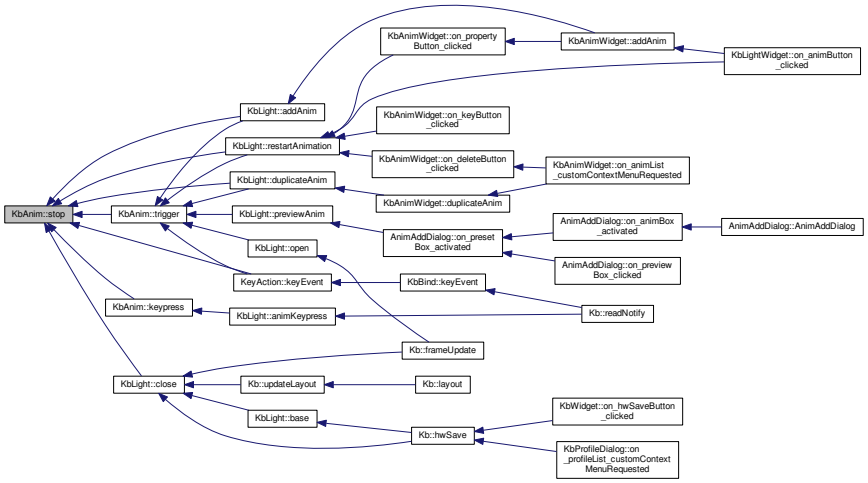
Referenced by KbLight::addAnim(), KbLight::close(), KbLight::duplicateAnim(), KeyAction::keyEvent(), keypress(), KbLight::restartAnimation(), and trigger().

```
312                 {
313     if(_script)
314         _script->end();
315     repeatTime = 0;
316     kpRepeatTime = 0;
317     repeatMsec = 0;
318     kpRepeatMsec = 0;
319     stopTime = 0;
320     kpStopTime = 0;
321     repeatKey = "";
322     _isActive = _isActiveKp = false;
323 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.4.30 void KbAnim::trigger (quint64 *timestamp*, bool *ignoreParameter* = `false`)

Definition at line 209 of file kbanim.cpp.

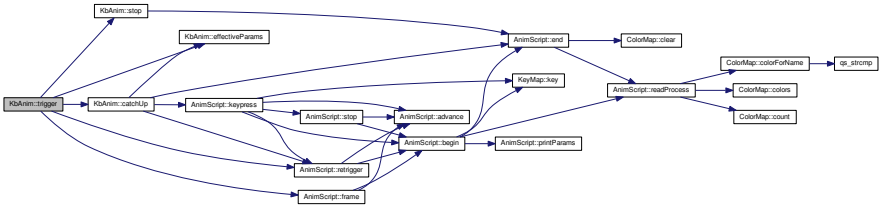
References _isActive, _script, catchUp(), effectiveParams(), AnimScript::frame(), repeatMsec, repeatTime, AnimScript::retrigger(), stop(), and stopTime.

Referenced by KbLight::addAnim(), KbLight::duplicateAnim(), KeyAction::keyEvent(), KbLight::open(), KbLight::previewAnim(), and KbLight::restartAnimation().

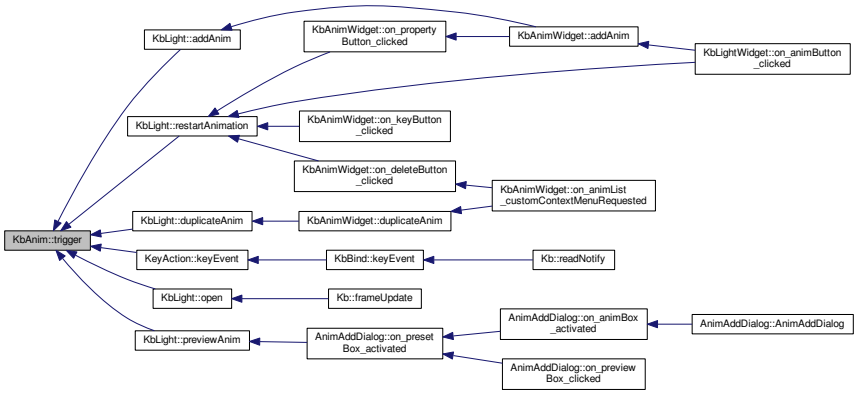
```
209                                                 {
210     if(!_script)
211         return;
212     QMap<QString, QVariant> parameters = effectiveParams();
213 
214     catchUp(timestamp);
215     if(parameters.value("trigger").toBool() || ignoreParameter){
216         _isActive = true;
217         int delay = round(parameters.value("delay").toDouble() * 1000.);
218         if(delay > 0){
219             // If delay is enabled, wait to trigger the event
220             timestamp += delay;
221             repeatTime = timestamp;
222         } else
223             _script->retrigger(timestamp, true);
224 
225         int repeat = round(parameters.value("repeat").toDouble() * 1000.);
226         if(repeat <= 0){
227             // If no repeat allowed, calculate stop time in seconds
228             repeatMsec = -1;
229             double stop = parameters.value("stop").toDouble();
230             if(stop <= 0)
231                 stopTime = 0;
232             else
233                 stopTime = timestamp + round(stop * 1000.);
234         } else {
235             // If repeat is allowed, calculate stop time in repetitions
236             repeatMsec = repeat;
237             if(delay <= 0)
238                 repeatTime = timestamp + repeat;
239             int stop = parameters.value("stop").toInt();
240             if(stop < 0)
241                 stopTime = 0;
242             else
243                 stopTime = timestamp + repeat * (1 + stop);
244         }
245     }
246     // Ask the script for a frame even if we didn't do anything here. This way ckb knows the script is
    responding.
247     _script->frame(timestamp);
```

```
248 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.4.31 void KbAnim::updateParams ( ) [private]

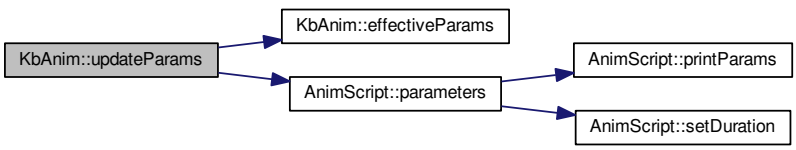
Definition at line 132 of file kbanim.cpp.

References _script, effectiveParams(), AnimScript::parameters(), and repeatKey.

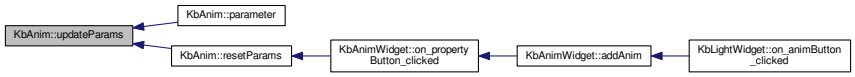
Referenced by parameter(), and resetParams().

```
132                          {
133     if(_script)
134         _script->parameters(effectiveParams());
135     repeatKey = "";
136 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.23.5 Field Documentation

12.23.5.1 QUuid KbAnim::_guid [private]

Definition at line 110 of file kbanim.h.

Referenced by guid(), KbAnim(), newId(), and save().

12.23.5.2 bool KbAnim::_isActive [private]

Definition at line 114 of file kbanim.h.

Referenced by catchUp(), isActive(), keypress(), reInit(), stop(), and trigger().

12.23.5.3 bool KbAnim::_isActiveKp [private]

Definition at line 114 of file kbanim.h.

Referenced by catchUp(), isActive(), keypress(), reInit(), and stop().

12.23.5.4 QStringList KbAnim::_keys [private]

Definition at line 92 of file kbanim.h.

Referenced by KbAnim(), keys(), reInit(), and save().

12.23.5.5 KeyMap KbAnim::_map [private]

Definition at line 91 of file kbanim.h.

Referenced by KbAnim(), map(), and reInit().

12.23.5.6 Mode KbAnim::_mode [private]

Definition at line 113 of file kbanim.h.

Referenced by blend(), KbAnim(), mode(), and save().

12.23.5.7 QString KbAnim::_name [private]

Definition at line 111 of file kbanim.h.

Referenced by KbAnim(), name(), and save().

12.23.5.8 bool KbAnim::_needsSave [private]

Definition at line 115 of file kbanim.h.

Referenced by commitParams(), mode(), name(), needsSave(), newId(), opacity(), and save().

12.23.5.9 float KbAnim::_opacity [private]

Definition at line 112 of file kbanim.h.

Referenced by blend(), KbAnim(), opacity(), and save().

12.23.5.10 QMap<QString, QVariant> KbAnim::_parameters [private]

Definition at line 94 of file kbanim.h.

Referenced by commitParams(), effectiveParams(), hasParameter(), KbAnim(), parameter(), and save().

12.23.5.11 AnimScript* KbAnim::_script [private]

Definition at line 86 of file kbanim.h.

Referenced by blend(), catchUp(), isRunning(), KbAnim(), keypress(), parameter(), reInit(), script(), stop(), trigger(), and updateParams().

12.23.5.12 QUuid KbAnim::_scriptGuid [private]

Definition at line 88 of file kbanim.h.

Referenced by KbAnim(), and save().

12.23.5.13 QString KbAnim::_scriptName [private]

Definition at line 89 of file kbanim.h.

Referenced by KbAnim(), save(), and scriptName().

12.23.5.14 QMap<QString, QVariant> KbAnim::_tempParameters [private]

Definition at line 96 of file kbanim.h.

Referenced by commitParams(), effectiveParams(), parameter(), and resetParams().

12.23.5.15 int KbAnim::kpRepeatMsec [private]

Definition at line 106 of file kbanim.h.

Referenced by catchUp(), keypress(), and stop().

12.23.5.16 quint64 KbAnim::kpRepeatTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), and stop().

12.23.5.17 quint64 KbAnim::kpStopTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), and stop().

12.23.5.18 QString KbAnim::repeatKey [private]

Definition at line 104 of file kbanim.h.

Referenced by catchUp(), keypress(), reInit(), stop(), and updateParams().

12.23.5.19 int KbAnim::repeatMsec [private]

Definition at line 106 of file kbanim.h.

Referenced by catchUp(), keypress(), stop(), and trigger().

12.23.5.20 quint64 KbAnim::repeatTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), stop(), and trigger().

12.23.5.21 quint64 KbAnim::stopTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), stop(), and trigger().

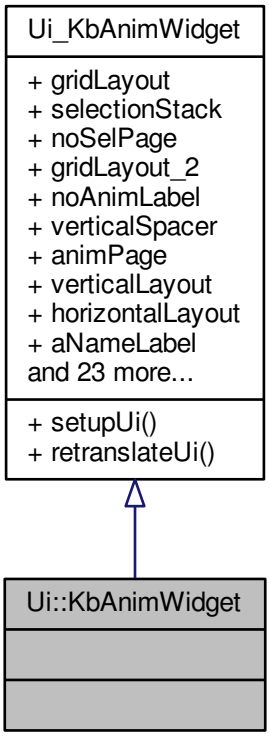
The documentation for this class was generated from the following files:

- src/ckb/kbanim.h
- src/ckb/kbanim.cpp

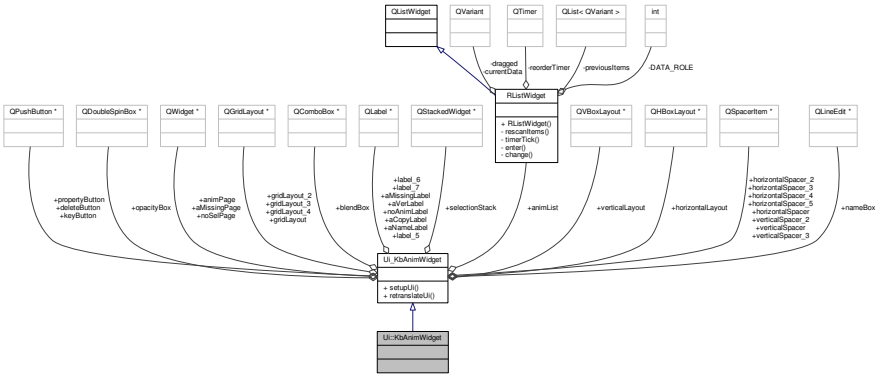
12.24 Ui::KbAnimWidget Class Reference

`#include <src/ckb/ui_kbanimwidget.h>`

Inheritance diagram for Ui::KbAnimWidget:



Collaboration diagram for Ui::KbAnimWidget:



Public Member Functions

- void setupUi (QWidget *KbAnimWidget)
- void retranslateUi (QWidget *KbAnimWidget)

Data Fields

- QGridLayout * gridLayout
- QStackedWidget * selectionStack
- QWidget * noSelPage
- QGridLayout * gridLayout_2
- QLabel * noAnimLabel
- QSpacerItem * verticalSpacer
- QWidget * animPage
- QVBoxLayout * verticalLayout
- QHBoxLayout * horizontalLayout
- QLabel * aNameLabel
- QLabel * aVerLabel
- QSpacerItem * horizontalSpacer
- QLabel * aCopyLabel
- QSpacerItem * verticalSpacer_3
- QGridLayout * gridLayout_3
- QLineEdit * nameBox
- QDoubleSpinBox * opacityBox
- QComboBox * blendBox
- QLabel * label_6
- QSpacerItem * horizontalSpacer_4
- QSpacerItem * horizontalSpacer_2
- QLabel * label_5
- QSpacerItem * horizontalSpacer_3
- QLabel * label_7
- QPushButton * keyButton
- QPushButton * deleteButton
- QPushButton * propertyButton
- QSpacerItem * horizontalSpacer_5
- QSpacerItem * verticalSpacer_2
- QWidget * aMissingPage
- QGridLayout * gridLayout_4
- QLabel * aMissingLabel
- RListWidget * animList

12.24.1 Detailed Description

Definition at line 269 of file ui_kbanimwidget.h.

12.24.2 Member Function Documentation

12.24.2.1 void Ui_KbAnimWidget::retranslateUi (QWidget * *KbAnimWidget*) [inline], [inherited]

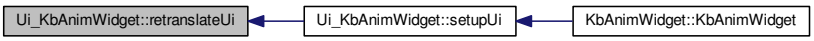
Definition at line 241 of file ui_kbanimwidget.h.

References Ui_KbAnimWidget::aCopyLabel, Ui_KbAnimWidget::aMissingLabel, Ui_KbAnimWidget::aNameLabel, Ui_KbAnimWidget::aVerLabel, Ui_KbAnimWidget::blendBox, Ui_KbAnimWidget::deleteButton, Ui_KbAnimWidget::keyButton, Ui_KbAnimWidget::label_5, Ui_KbAnimWidget::label_6, Ui_KbAnimWidget::label_7, Ui_KbAnimWidget::noAnimLabel, Ui_KbAnimWidget::opacityBox, and Ui_KbAnimWidget::propertyButton.

Referenced by Ui_KbAnimWidget::setupUi().

```
242     {
243         KbAnimWidget->setWindowTitle(QApplication::translate("KbAnimWidget", "Form", Q_NULLPTR));
244         noAnimLabel->setText(QApplication::translate("KbAnimWidget", "No animations in this mode
    ", Q_NULLPTR));
245         aNameLabel->setText(QApplication::translate("KbAnimWidget", "Animation name", Q_NULLPTR))
    ;
246         aVerLabel->setText(QApplication::translate("KbAnimWidget", "v0.0", Q_NULLPTR));
247         aCopyLabel->setText(QApplication::translate("KbAnimWidget", "Copyright \302\251",
    Q_NULLPTR));
248         opacityBox->setSuffix(QApplication::translate("KbAnimWidget", "%", Q_NULLPTR));
249         blendBox->clear();
250         blendBox->insertItems(0, QStringList()
251          << QApplication::translate("KbAnimWidget", "Normal", Q_NULLPTR)
252          << QApplication::translate("KbAnimWidget", "Add", Q_NULLPTR)
253          << QApplication::translate("KbAnimWidget", "Subtract", Q_NULLPTR)
254          << QApplication::translate("KbAnimWidget", "Multiply", Q_NULLPTR)
255          << QApplication::translate("KbAnimWidget", "Divide", Q_NULLPTR)
256         );
257         label_6->setText(QApplication::translate("KbAnimWidget", "Blend mode:", Q_NULLPTR));
258         label_5->setText(QApplication::translate("KbAnimWidget", "Opacity:", Q_NULLPTR));
259         label_7->setText(QApplication::translate("KbAnimWidget", "Name:", Q_NULLPTR));
260         keyButton->setText(QApplication::translate("KbAnimWidget", "Set keys", Q_NULLPTR));
261         deleteButton->setText(QApplication::translate("KbAnimWidget", "Delete animation",
    Q_NULLPTR));
262         propertyButton->setText(QApplication::translate("KbAnimWidget", "Edit properties",
    Q_NULLPTR));
263         aMissingLabel->setText(QApplication::translate("KbAnimWidget", "Mising animation",
    Q_NULLPTR));
264     } // retranslateUi
```

Here is the caller graph for this function:



12.24.2.2 void Ui_KbAnimWidget::setupUi (QWidget * *KbAnimWidget*) [inline],[inherited]

Definition at line 69 of file ui_kbanimwidget.h.

References Ui_KbAnimWidget::aCopyLabel, Ui_KbAnimWidget::aMissingLabel, Ui_KbAnimWidget::aMissingPage, Ui_KbAnimWidget::aNameLabel, Ui_KbAnimWidget::animList, Ui_KbAnimWidget::animPage, Ui_KbAnimWidget::aVerLabel, Ui_KbAnimWidget::blendBox, Ui_KbAnimWidget::deleteButton, Ui_KbAnimWidget::gridLayout, Ui_KbAnimWidget::gridLayout_2, Ui_KbAnimWidget::gridLayout_3, Ui_KbAnimWidget::gridLayout_4, Ui_KbAnimWidget::horizontalLayout, Ui_KbAnimWidget::horizontalSpacer, Ui_KbAnimWidget::horizontalSpacer_2, Ui_KbAnimWidget::horizontalSpacer_3, Ui_KbAnimWidget::horizontalSpacer_4, Ui_KbAnimWidget::horizontalSpacer_5, Ui_KbAnimWidget::keyButton, Ui_KbAnimWidget::label_5, Ui_KbAnimWidget::label_6, Ui_KbAnimWidget::label_7, Ui_KbAnimWidget::nameBox, Ui_KbAnimWidget::noAnimLabel, Ui_KbAnimWidget::noSelPage, Ui_KbAnimWidget::opacityBox, Ui_KbAnimWidget::propertyButton, Ui_KbAnimWidget::retranslateUi(), Ui_KbAnimWidget::selectionStack, Ui_KbAnimWidget::verticalLayout, Ui_KbAnimWidget::verticalSpacer, Ui_KbAnimWidget::verticalSpacer_2, and Ui_KbAnimWidget::verticalSpacer_3.

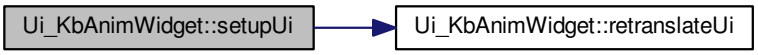
Referenced by KbAnimWidget::KbAnimWidget().

```
70     {
71         if (KbAnimWidget->objectName().isEmpty())
72             KbAnimWidget->setObjectName(QStringLiteral("KbAnimWidget"));
73         KbAnimWidget->resize(613, 300);
74         gridLayout = new QGridLayout(KbAnimWidget);
75         gridLayout->setObjectName(QStringLiteral("gridLayout"));
76         gridLayout->setContentsMargins(0, 0, 0, 0);
77         selectionStack = new QStackedWidget(KbAnimWidget);
78         selectionStack->setObjectName(QStringLiteral("selectionStack"));
79         noSelPage = new QWidget();
80         noSelPage->setObjectName(QStringLiteral("noSelPage"));
81         gridLayout_2 = new QGridLayout(noSelPage);
82         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
83         gridLayout_2->setContentsMargins(6, 6, 6, 6);
84         noAnimLabel = new QLabel(noSelPage);
```

```
85          noAnimLabel->setObjectName(QStringLiteral("noAnimLabel"));
86
87          gridLayout_2->addWidget(noAnimLabel, 0, 0, 1, 1);
88
89          verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
90
91          gridLayout_2->addItem(verticalSpacer, 1, 0, 1, 1);
92
93          selectionStack->addWidget(noSelPage);
94          animPage = new QWidget();
95          animPage->setObjectName(QStringLiteral("animPage"));
96          verticalLayout = new QVBoxLayout(animPage);
97          verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
98          verticalLayout->setContentsMargins(6, 0, 0, 0);
99          horizontalLayout = new QHBoxLayout();
100          horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
101          aNameLabel = new QLabel(animPage);
102          aNameLabel->setObjectName(QStringLiteral("aNameLabel"));
103          QFont font;
104          font.setBold(true);
105          font.setWeight(75);
106          aNameLabel->setFont(font);
107
108          horizontalLayout->addWidget(aNameLabel);
109
110          aVerLabel = new QLabel(animPage);
111          aVerLabel->setObjectName(QStringLiteral("aVerLabel"));
112
113          horizontalLayout->addWidget(aVerLabel);
114
115          horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
116
117          horizontalLayout->addItem(horizontalSpacer);
118
119          aCopyLabel = new QLabel(animPage);
120          aCopyLabel->setObjectName(QStringLiteral("aCopyLabel"));
121
122          horizontalLayout->addWidget(aCopyLabel);
123
124
125          verticalLayout->addLayout(horizontalLayout);
126
127          verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
128
129          verticalLayout->addItem(verticalSpacer_3);
130
131          gridLayout_3 = new QGridLayout();
132          gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
133          nameBox = new QLineEdit(animPage);
134          nameBox->setObjectName(QStringLiteral("nameBox"));
135          QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
136          sizePolicy.setHorizontalStretch(0);
137          sizePolicy.setVerticalStretch(0);
138          sizePolicy.setHeightForWidth(nameBox->sizePolicy().hasHeightForWidth());
139          nameBox->setSizePolicy(sizePolicy);
140          nameBox->setMinimumSize(QSize(150, 0));
141          nameBox->setMaximumSize(QSize(150, 16777215));
142
143          gridLayout_3->addWidget(nameBox, 0, 3, 1, 2);
144
145          opacityBox = new QDoubleSpinBox(animPage);
146          opacityBox->setObjectName(QStringLiteral("opacityBox"));
147          opacityBox->setDecimals(1);
148          opacityBox->setValue(100);
149
150          gridLayout_3->addWidget(opacityBox, 1, 3, 1, 1);
151
152          blendBox = new QComboBox(animPage);
153          blendBox->setObjectName(QStringLiteral("blendBox"));
154
155          gridLayout_3->addWidget(blendBox, 2, 3, 1, 2);
156
157          label_6 = new QLabel(animPage);
158          label_6->setObjectName(QStringLiteral("label_6"));
159
160          gridLayout_3->addWidget(label_6, 2, 1, 1, 1);
161
162          horizontalSpacer_4 = new QSpacerItem(0, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
163
164          gridLayout_3->addItem(horizontalSpacer_4, 0, 0, 1, 1);
165
166          horizontalSpacer_2 = new QSpacerItem(30, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
```

```
167
168         gridLayout_3->addItem(horizontalSpacer_2, 0, 2, 1, 1);
169
170         label_5 = new QLabel(animPage);
171         label_5->setObjectName(QStringLiteral("label_5"));
172
173         gridLayout_3->addWidget(label_5, 1, 1, 1, 1);
174
175         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
176
177         gridLayout_3->addItem(horizontalSpacer_3, 0, 5, 1, 1);
178
179         label_7 = new QLabel(animPage);
180         label_7->setObjectName(QStringLiteral("label_7"));
181
182         gridLayout_3->addWidget(label_7, 0, 1, 1, 1);
183
184         keyButton = new QPushButton(animPage);
185         keyButton->setObjectName(QStringLiteral("keyButton"));
186
187         gridLayout_3->addWidget(keyButton, 1, 6, 1, 1);
188
189         deleteButton = new QPushButton(animPage);
190         deleteButton->setObjectName(QStringLiteral("deleteButton"));
191
192         gridLayout_3->addWidget(deleteButton, 2, 6, 1, 1);
193
194         propertyButton = new QPushButton(animPage);
195         propertyButton->setObjectName(QStringLiteral("propertyButton"));
196
197         gridLayout_3->addWidget(propertyButton, 0, 6, 1, 1);
198
199         horizontalSpacer_5 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
200
201         gridLayout_3->addItem(horizontalSpacer_5, 1, 4, 1, 1);
202
203
204         verticalLayout->addLayout(gridLayout_3);
205
206         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
207
208         verticalLayout->addItem(verticalSpacer_2);
209
210         selectionStack->addWidget(animPage);
211         aMissingPage = new QWidget();
212         aMissingPage->setObjectName(QStringLiteral("aMissingPage"));
213         gridLayout_4 = new QGridLayout(aMissingPage);
214         gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
215         aMissingLabel = new QLabel(aMissingPage);
216         aMissingLabel->setObjectName(QStringLiteral("aMissingLabel"));
217         aMissingLabel->setFont(font);
218
219         gridLayout_4->addWidget(aMissingLabel, 0, 0, 1, 1);
220
221         selectionStack->addWidget(aMissingPage);
222
223         gridLayout->addWidget(selectionStack, 0, 2, 2, 1);
224
225         animList = new RListWidget(KbAnimWidget);
226         animList->setObjectName(QStringLiteral("animList"));
227         animList->setMaximumSize(QSize(130, 16777215));
228         animList->setContextMenuPolicy(Qt::CustomContextMenu);
229         animList->setFrameShadow(QFrame::Plain);
230         animList->setDragDropMode(QAbstractItemView::InternalMove);
231         animList->setMovement(QListView::Snap);
232
233         gridLayout->addWidget(animList, 0, 0, 2, 1);
234
235
236         retranslateUi(KbAnimWidget);
237
238         QMetaObject::connectSlotsByName(KbAnimWidget);
239     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.24.3 Field Documentation

12.24.3.1 QLabel* Ui_KbAnimWidget::aCopyLabel [inherited]

Definition at line 47 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.2 QLabel* Ui_KbAnimWidget::aMissingLabel [inherited]

Definition at line 66 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.3 QWidget* Ui_KbAnimWidget::aMissingPage [inherited]

Definition at line 64 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.4 QLabel* Ui_KbAnimWidget::aNameLabel [inherited]

Definition at line 44 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.5 RListWidget* Ui_KbAnimWidget::animList [inherited]

Definition at line 67 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::addAnim(), KbAnimWidget::clearSelection(), KbAnimWidget::duplicateAnim(), KbAnimWidget::KbAnimWidget(), KbAnimWidget::on_animList_customContextMenuRequested(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::on_nameBox_textEdited(), KbAnimWidget::refreshList(), KbAnimWidget::reorderAnims(), and Ui_KbAnimWidget::setupUi().

12.24.3.6 QWidget* Ui_KbAnimWidget::animPage [inherited]

Definition at line 41 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.7 QLabel* Ui_KbAnimWidget::aVerLabel [inherited]

Definition at line 45 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.8 QComboBox* Ui_KbAnimWidget::blendBox [inherited]

Definition at line 52 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.9 QPushButton* Ui_KbAnimWidget::deleteButton [inherited]

Definition at line 60 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), and Ui_KbAnimWidget::setupUi().

12.24.3.10 QGridLayout* Ui_KbAnimWidget::gridLayout [inherited]

Definition at line 35 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.11 QGridLayout* Ui_KbAnimWidget::gridLayout_2 [inherited]

Definition at line 38 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.12 QGridLayout* Ui_KbAnimWidget::gridLayout_3 [inherited]

Definition at line 49 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.13 QGridLayout* Ui_KbAnimWidget::gridLayout_4 [inherited]

Definition at line 65 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.14 QHBoxLayout* Ui_KbAnimWidget::horizontalLayout [inherited]

Definition at line 43 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.15 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer [inherited]

Definition at line 46 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.16 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_2 [inherited]

Definition at line 55 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.17 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_3 [inherited]

Definition at line 57 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.18 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_4 [inherited]

Definition at line 54 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.19 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_5 [inherited]

Definition at line 62 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.20 QPushButton* Ui_KbAnimWidget::keyButton [inherited]

Definition at line 59 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setSelectedKeys(), and Ui_KbAnimWidget::setupUi().

12.24.3.21 QLabel* Ui_KbAnimWidget::label_5 [inherited]

Definition at line 56 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), and Ui_KbAnimWidget::setupUi().

12.24.3.22 QLabel* Ui_KbAnimWidget::label_6 [inherited]

Definition at line 53 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), and Ui_KbAnimWidget::setupUi().

12.24.3.23 QLabel* Ui_KbAnimWidget::label_7 [inherited]

Definition at line 58 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), and Ui_KbAnimWidget::setupUi().

12.24.3.24 QLineEdit* Ui_KbAnimWidget::nameBox [inherited]

Definition at line 50 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::on_animList_itemChanged(), KbAnimWidget::on_propertyButton_clicked(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.25 QLabel* Ui_KbAnimWidget::noAnimLabel [inherited]

Definition at line 39 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::addAnim(), KbAnimWidget::duplicateAnim(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::refreshList(), Ui_KbAnimWidget::retranslateUi(), and Ui_KbAnimWidget::setupUi().

12.24.3.26 QWidget* Ui_KbAnimWidget::noSelPage [inherited]

Definition at line 37 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.27 QDoubleSpinBox* Ui_KbAnimWidget::opacityBox [inherited]

Definition at line 51 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.28 QPushButton* Ui_KbAnimWidget::propertyButton [inherited]

Definition at line 61 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::retranslateUi(), and Ui_KbAnimWidget::setupUi().

12.24.3.29 QStackedWidget* Ui_KbAnimWidget::selectionStack [inherited]

Definition at line 36 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::setCurrent(), and Ui_KbAnimWidget::setupUi().

12.24.3.30 QVBoxLayout* Ui_KbAnimWidget::verticalLayout [inherited]

Definition at line 42 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.31 QSpacerItem* Ui_KbAnimWidget::verticalSpacer [inherited]

Definition at line 40 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.32 QSpacerItem* Ui_KbAnimWidget::verticalSpacer_2 [inherited]

Definition at line 63 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

12.24.3.33 QSpacerItem* Ui_KbAnimWidget::verticalSpacer_3 [inherited]

Definition at line 48 of file ui_kbanimwidget.h.

Referenced by Ui_KbAnimWidget::setupUi().

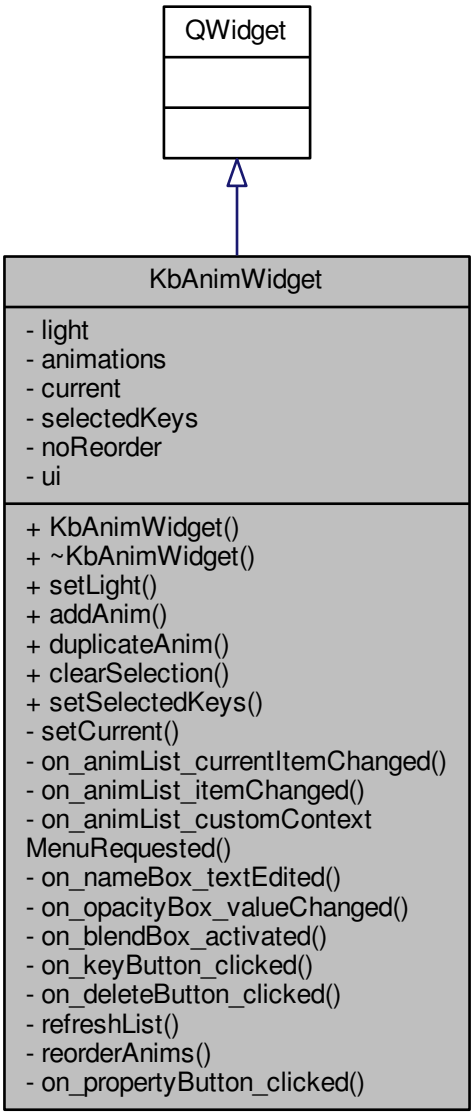
The documentation for this class was generated from the following file:

- src/ckb/ui_kbanimwidget.h

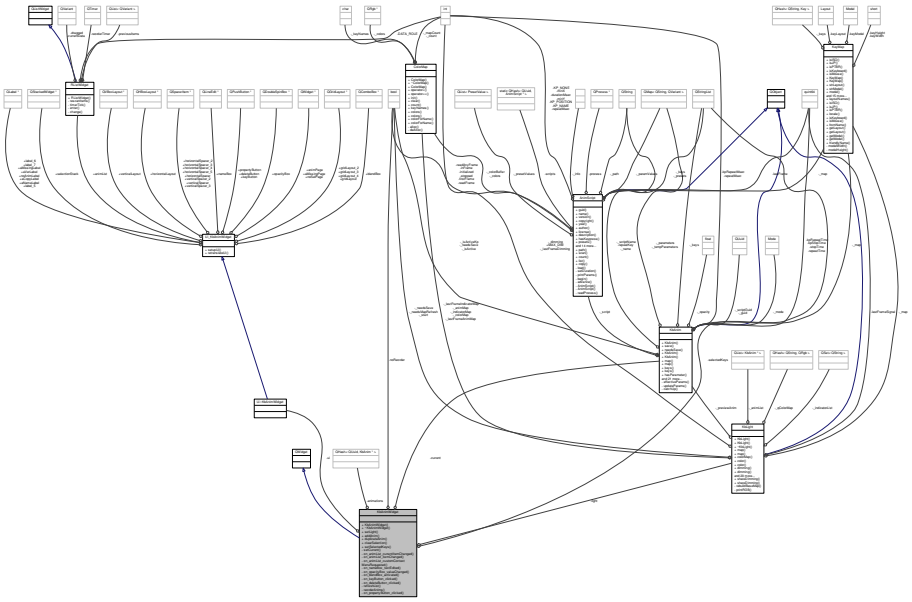
12.25 KbAnimWidget Class Reference

`#include <src/ckb/kbanimwidget.h>`

Inheritance diagram for KbAnimWidget:



Collaboration diagram for KbAnimWidget:



Signals

- void animChanged (KbAnim ∗selection)
- void didUpdateSelection (QStringList keys)

Public Member Functions

- KbAnimWidget (QWidget ∗parent=0)
- ∼KbAnimWidget ()
- void setLight (KbLight ∗newLight)
- void addAnim (const AnimScript ∗base, const QStringList &keyList, const QString &name, const QMap< QString, QVariant > &preset)
- void duplicateAnim (KbAnim ∗old)
- void clearSelection ()
- void setSelectedKeys (const QStringList &keys)

Private Slots

- void on_animList_currentItemChanged (QListWidgetItem ∗current, QListWidgetItem ∗previous)
- void on_animList_itemChanged (QListWidgetItem ∗item)
- void on_animList_customContextMenuRequested (const QPoint &pos)
- void on_nameBox_textEdited (const QString &arg1)
- void on_opacityBox_valueChanged (double arg1)
- void on_blendBox_activated (int index)
- void on_keyButton_clicked ()
- void on_deleteButton_clicked ()
- void refreshList ()
- void reorderAnims ()
- void on_propertyButton_clicked ()

Private Member Functions

- void setCurrent (KbAnim *newCurrent)

Private Attributes

- KbLight * light
- QHash< QUuid, KbAnim * > animations
- KbAnim * current
- QStringList selectedKeys
- bool noReorder
- Ui::KbAnimWidget * ui

12.25.1 Detailed Description

Definition at line 15 of file kbanimwidget.h.

12.25.2 Constructor & Destructor Documentation

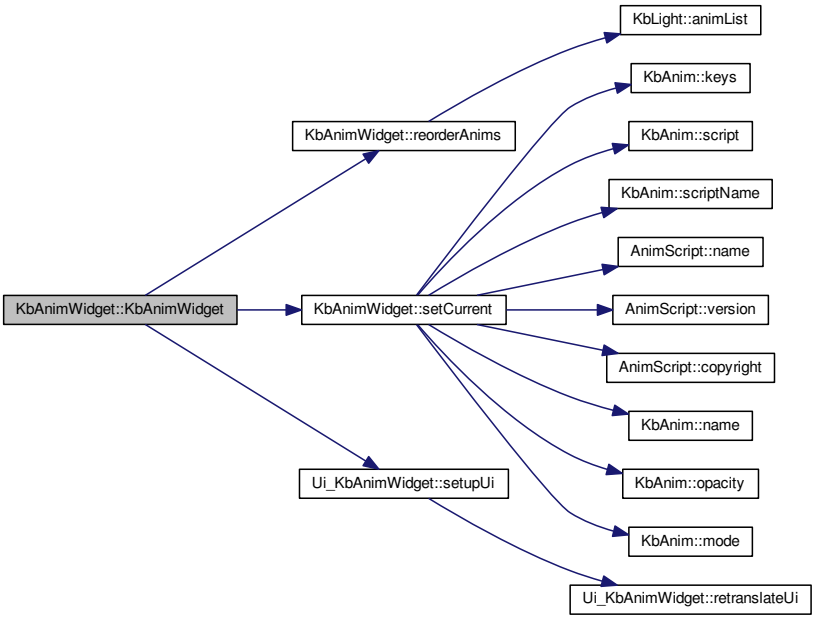
12.25.2.1 KbAnimWidget::KbAnimWidget (QWidget * *parent* = 0) [explicit]

Definition at line 7 of file kbanimwidget.cpp.

References Ui_KbAnimWidget::animList, reorderAnims(), setCurrent(), Ui_KbAnimWidget::setupUi(), and ui.

```
7                                                :
8     QWidget(parent), light(0), current(0), noReorder(false),
9     ui(new Ui::KbAnimWidget)
10 {
11     ui->setupUi(this);
12     ui->animList->setVisible(false);
13     setCurrent(0);
14     connect(ui->animList, SIGNAL(orderChanged()), this, SLOT(
      reorderAnims()));
15 }
```

Here is the call graph for this function:



12.25.2.2 KbAnimWidget::~KbAnimWidget ( )

Definition at line 17 of file kbanimwidget.cpp.

References ui.

```
17                              {
18     delete ui;
19 }
```

12.25.3 Member Function Documentation

12.25.3.1 void KbAnimWidget::addAnim (const AnimScript * *base*, const QStringList & *keyList*, const QString & *name*, const QMap< QString, QVariant > & *preset*)

Definition at line 82 of file kbanimwidget.cpp.

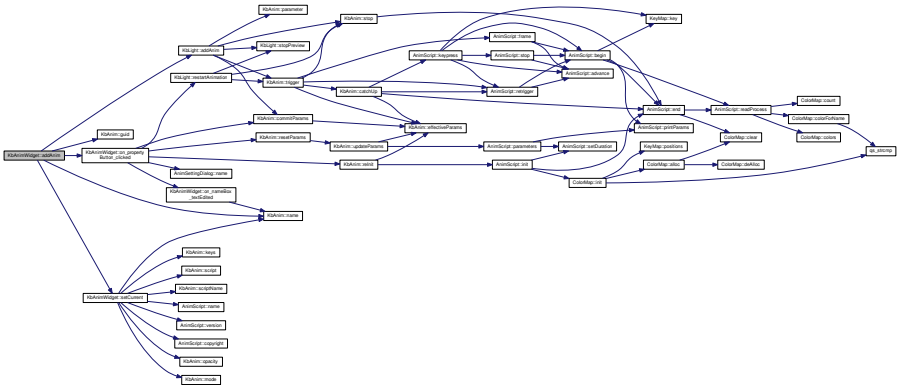
References KbLight::addAnim(), animations, Ui_KbAnimWidget::animList, KbAnim::guid(), light, KbAnim::name(), Ui_KbAnimWidget::noAnimLabel, noReorder, on_propertyButton_clicked(), setCurrent(), and ui.

Referenced by KbLightWidget::on_animButton_clicked().

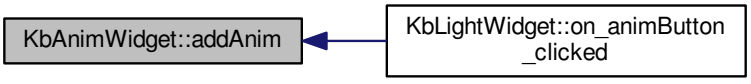
```
82
                                    {
83     if(!light)
84         return;
85     noReorder = true;
86     KbAnim* animation = light->addAnim(base, keyList, name, preset);
87     QListWidgetItem* item = new QListWidgetItem(animation->name(), ui->
      animList);
88     item->setData(Qt::UserRole, animation->guid());
89     item->setFlags(item->flags() | Qt::ItemIsEditable);
90     animations[animation->guid()] = animation;
91     ui->animList->addItem(item);
92     ui->animList->setCurrentItem(item);
```

```
93      ui->animList->setVisible(true);
94      ui->noAnimLabel->setVisible(false);
95
96      setCurrent(animation);
97      noReorder = false;
98
99      // Activate settings dialog
100      on_propertyButton_clicked();
101 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



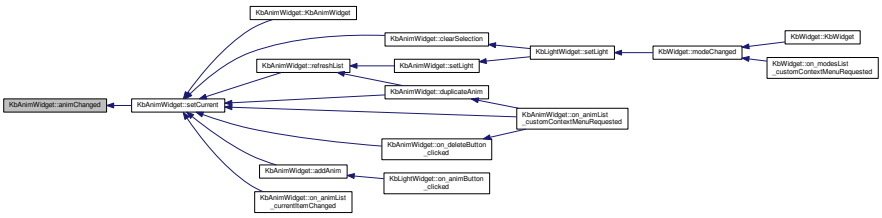
12.25.3.2 void KbAnimWidget::animChanged (KbAnim * *selection*) [signal]

Definition at line 213 of file moc_kbanimwidget.cpp.

Referenced by setCurrent().

```
214 {
215      void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
216      QMetaObject::activate(this, &staticMetaObject, 0, _a);
217 }
```

Here is the caller graph for this function:



12.25.3.3 void KbAnimWidget::clearSelection ( )

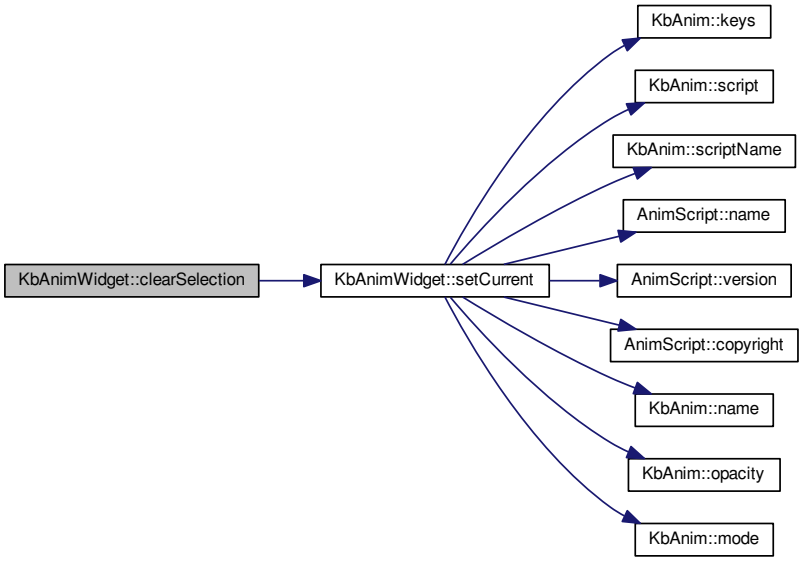
Definition at line 77 of file kbanimwidget.cpp.

References Ui_KbAnimWidget::animList, setCurrent(), and ui.

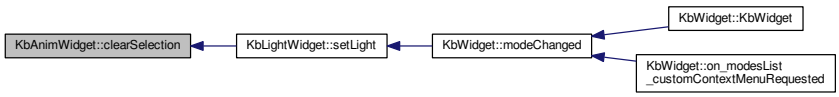
Referenced by KbLightWidget::setLight().

```
77                                  {
78     ui->animList->setCurrentItem(0);
79     setCurrent(0);
80 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



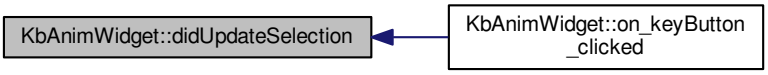
12.25.3.4 void KbAnimWidget::didUpdateSelection (QStringList *keys*) [signal]

Definition at line 220 of file moc_kbanimwidget.cpp.

Referenced by on_keyButton_clicked().

```
221 {
222     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
223     QMetaObject::activate(this, &staticMetaObject, 1, _a);
224 }
```

Here is the caller graph for this function:



12.25.3.5 void KbAnimWidget::duplicateAnim (KbAnim * *old*)

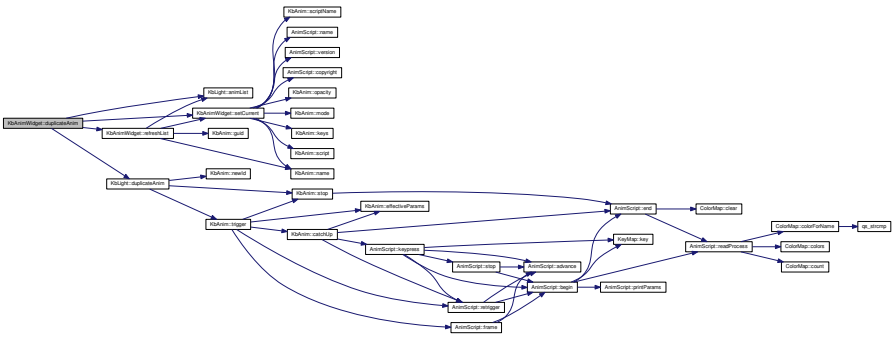
Definition at line 103 of file kbanimwidget.cpp.

References KbLight::animList(), Ui_KbAnimWidget::animList, KbLight::duplicateAnim(), light, Ui_KbAnimWidget::noAnimLabel, noReorder, refreshList(), setCurrent(), and ui.

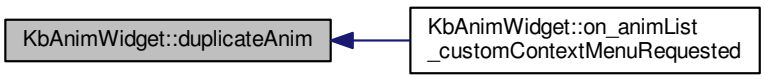
Referenced by on_animList_customContextMenuRequested().

```
103                                          {
104     if(!light)
105         return;
106     noReorder = true;
107     KbAnim* animation = light->duplicateAnim(old);
108     // Refresh the list. insertItem doesn't seem to place the item in the correct position on its own...
109     refreshList();
110     ui->animList->setCurrentRow(light->animList().indexOf(animation));
111     ui->animList->setVisible(true);
112     ui->noAnimLabel->setVisible(false);
113 
114     setCurrent(animation);
115     noReorder = false;
116 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



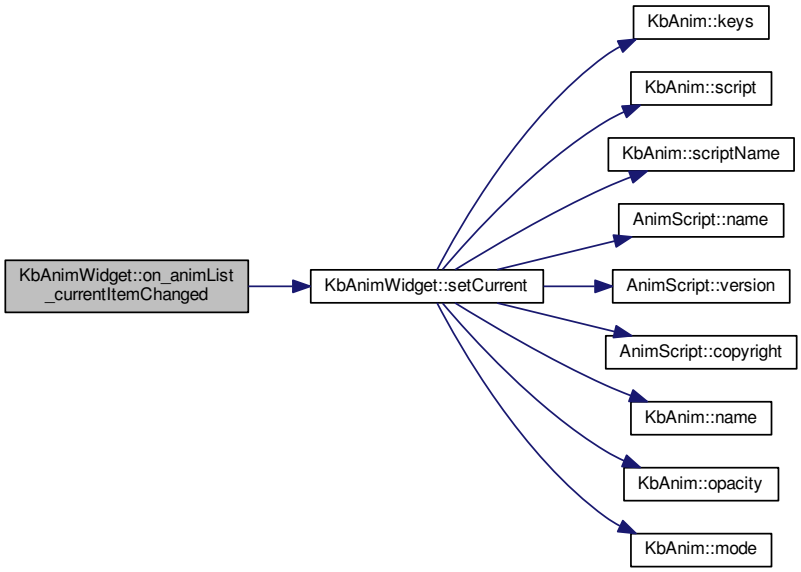
12.25.3.6 void KbAnimWidget::on_animList_currentItemChanged (QListWidgetItem * *current*, QListWidgetItem * *previous*) [private], [slot]

Definition at line 151 of file kbanimwidget.cpp.

References animations, and setCurrent().

```
151                                                                                                                 {
152     if(!current)
153         setCurrent(0);
154     else
155         setCurrent(animations[current->data(Qt::UserRole).toUuid()]);
156 }
```

Here is the call graph for this function:



12.25.3.7 void KbAnimWidget::on_animList_customContextMenuRequested (const QPoint & *pos*) [private], [slot]

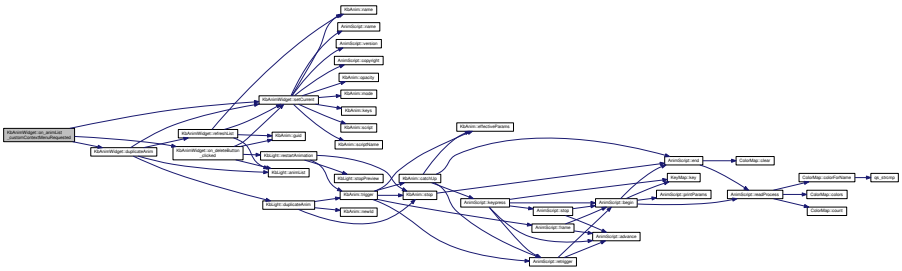
Definition at line 169 of file kbanimwidget.cpp.

References animations, Ui_KbAnimWidget::animList, current, duplicateAnim(), on_deleteButton_clicked(), setCurrent(), and ui.

```
169                                                                                                                 {
170     QListWidgetItem* item = ui->animList->itemAt(pos);
171     if(!item)
172         return;
173     setCurrent(animations[item->data(Qt::UserRole).toUuid()]);
174
175     QMenu menu(this);
176     QAction* rename = new QAction("Rename...", this);
177     QAction* duplicate = new QAction("Duplicate", this);
178     QAction* del = new QAction("Delete", this);
179     menu.addAction(rename);
180     menu.addAction(duplicate);
181     menu.addAction(del);
182     QAction* result = menu.exec(QCursor::pos());
183     if(result == rename)
184         ui->animList->editItem(item);
```

```
185     else if(result == duplicate)
186         duplicateAnim(current);
187     else if(result == del)
188         on_deleteButton_clicked();
189 }
```

Here is the call graph for this function:



12.25.3.8 void KbAnimWidget::on_animList_itemChanged (QListWidgetItem * *item*) [private], [slot]

Definition at line 158 of file kbanimwidget.cpp.

References anim, animations, current, KbAnim::name(), Ui_KbAnimWidget::nameBox, noReorder, and ui.

```
158                                                                 {
159     if(item){
160         KbAnim* anim = animations[item->data(Qt::UserRole).toUuid()];
161         if(anim){
162             anim->name(item->text().trimmed());
163             if(anim == current && !noReorder)
164                 ui->nameBox->setText(anim->name());
165         }
166     }
167 }
```

Here is the call graph for this function:

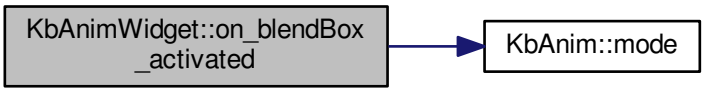
12.25.3.9 void KbAnimWidget::on_blendBox_activated (int *index*) [private], [slot]

Definition at line 205 of file kbanimwidget.cpp.

References current, and KbAnim::mode().

```
205                                                 {
206     if(current)
207         current->mode((KbAnim::Mode)index);
208 }
```

Here is the call graph for this function:



12.25.3.10 void KbAnimWidget::on_deleteButton_clicked ( ) [private], [slot]

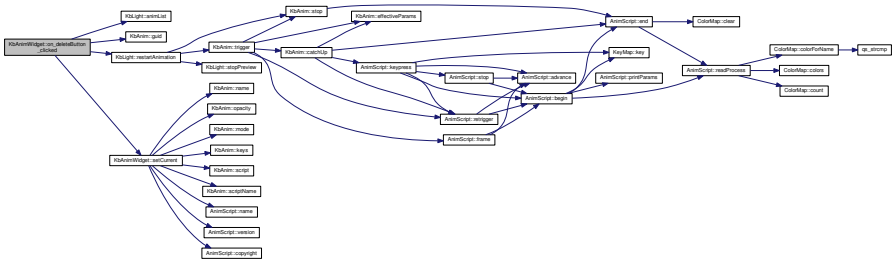
Definition at line 226 of file kbanimwidget.cpp.

References animations, KbLight::animList(), Ui_KbAnimWidget::animList, current, KbAnim::guid(), light, Ui_KbAnimWidget::noAnimLabel, KbLight::restartAnimation(), setCurrent(), and ui.

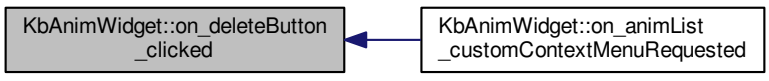
Referenced by on_animList_customContextMenuRequested().

```
226                                          {
227     if(current){
228         animations.remove(current->guid());
229         QList<KbAnim*> animList = light->animList();
230         animList.removeAll(current);
231         light->animList(animList);
232         current->deleteLater();
233         setCurrent(0);
234         delete ui->animList->currentItem();
235         if(animations.count() == 0){
236             ui->animList->setVisible(false);
237             ui->noAnimLabel->setVisible(true);
238         }
239     }
240     light->restartAnimation();
241 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



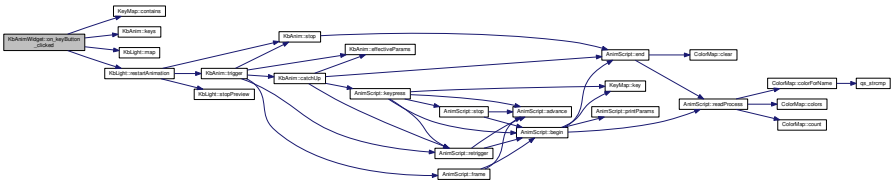
12.25.3.11 void KbAnimWidget::on_keyButton_clicked ( ) [private], [slot]

Definition at line 210 of file kbanimwidget.cpp.

References KeyMap::contains(), current, didUpdateSelection(), KbAnim::keys(), light, KbLight::map(), KbLight::restartAnimation(), and selectedKeys.

```
210                                         {
211     if(current){
212         QStringList keys = selectedKeys;
213         // If any keys were selected previously that aren't in the keymap now, leave them in
214         // This is important for layout compatibility - e.g. being able to select both bslash_iso and
        bslash, because no single layout contains both
215         const KeyMap& map = light->map();
216         foreach(const QString& key, current->keys()){
217             if(!map.contains(key))
218                 keys << key;
219         }
220         current->keys(keys);
221         emit didUpdateSelection(selectedKeys);
222     }
223     light->restartAnimation();
224 }
```

Here is the call graph for this function:



12.25.3.12 void KbAnimWidget::on_nameBox_textEdited (const QString & *arg1*) [private], [slot]

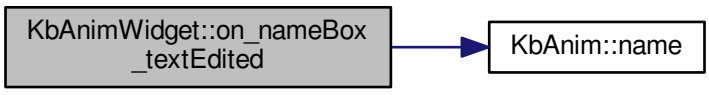
Definition at line 191 of file kbanimwidget.cpp.

References Ui_KbAnimWidget::animList, current, KbAnim::name(), noReorder, and ui.

Referenced by on_propertyButton_clicked().

```
191                                                     {
192     if(current){
193         noReorder = true;
194         current->name(arg1.trimmed());
195         ui->animList->currentItem()->setText(current->name());
196         noReorder = false;
197     }
198 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

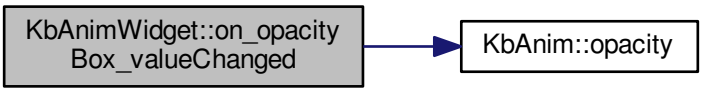
12.25.3.13 void KbAnimWidget::on_opacityBox_valueChanged (double *arg1*) [private], [slot]

Definition at line 200 of file kbanimwidget.cpp.

References current, and KbAnim::opacity().

```
200                                                     {
201     if(current)
202         current->opacity(arg1 / 100.);
203 }
```

Here is the call graph for this function:



12.25.3.14 void KbAnimWidget::on_propertyButton_clicked ( ) [private], [slot]

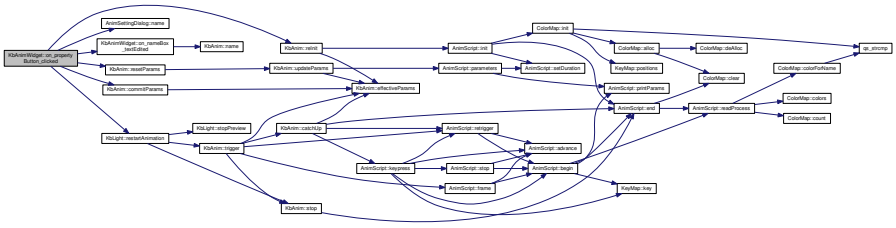
Definition at line 243 of file kbanimwidget.cpp.

References KbAnim::commitParams(), current, light, AnimSettingDialog::name(), Ui_KbAnimWidget::nameBox, on_nameBox_textEdited(), KbAnim::reInit(), KbAnim::resetParams(), KbLight::restartAnimation(), and ui.

Referenced by addAnim().

```
243                                         {
244     if(!current)
245         return;
246     // Present animation property popup
247     AnimSettingDialog dialog(this, current);
248     dialog.exec();
249     if(dialog.result() != QDialog::Accepted){
250         current->resetParams();
251         return;
252     }
253     // Apply settings and restart all animations
254     current->commitParams();
255     current->reInit();
256     light->restartAnimation();
257     // Update name
258     ui->nameBox->setText(dialog.name());
259     on_nameBox_textEdited(dialog.name());
260 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.25.3.15 void KbAnimWidget::refreshList ( ) [private], [slot]

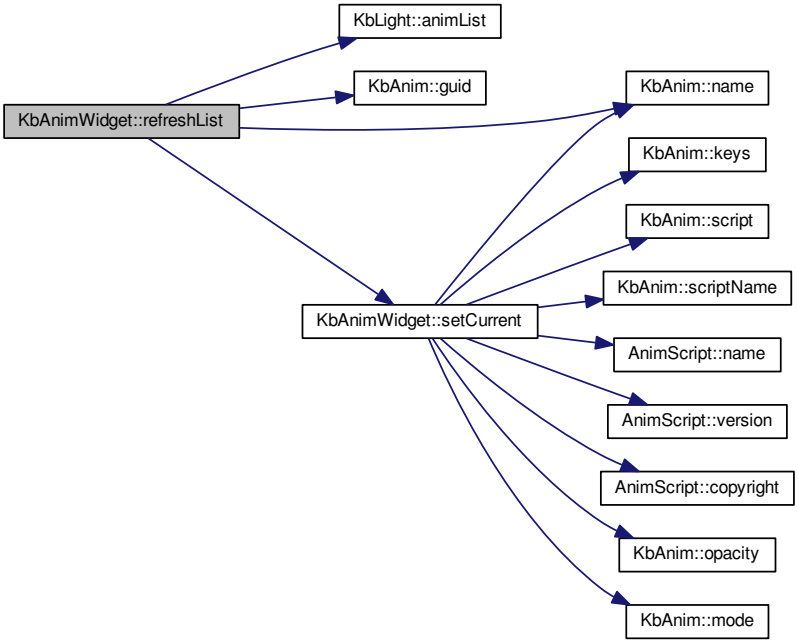
Definition at line 32 of file kbanimwidget.cpp.

References anim, animations, KbLight::animList(), Ui_KbAnimWidget::animList, KbAnim::guid(), light, KbAnim::name(), Ui_KbAnimWidget::noAnimLabel, noReorder, setCurrent(), and ui.

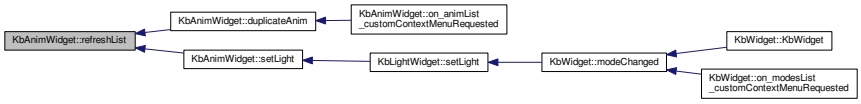
Referenced by duplicateAnim(), and setLight().

```
32                                {
33     noReorder = true;
34     setCurrent(0);
35     ui->animList->clear();
36     animations.clear();
37     // Add the animations from the new lighting mode
38     if(!light){
39         ui->animList->setVisible(false);
40         ui->noAnimLabel->setVisible(true);
41         return;
42     }
43     QList<KbAnim*> newAnimations = light->animList();
44     if(newAnimations.count() == 0){
45         ui->animList->setVisible(false);
46         ui->noAnimLabel->setVisible(true);
47         return;
48     }
49     ui->animList->setVisible(true);
50     foreach(KbAnim* anim, newAnimations){
51         QListWidgetItem* item = new QListWidgetItem(anim->name(), ui->
       animList);
52         item->setData(Qt::UserRole, anim->guid());
53         item->setFlags(item->flags() | Qt::ItemIsEditable);
54         animations[anim->guid()] = anim;
55         ui->animList->addItem(item);
56     }
57     ui->noAnimLabel->setVisible(false);
58     noReorder = false;
59 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.25.3.16 void KbAnimWidget::reorderAnims ( ) [private], [slot]

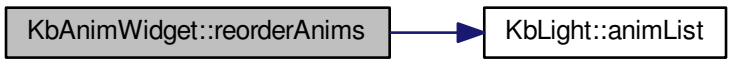
Definition at line 61 of file kbanimwidget.cpp.

References anim, animations, KbLight::animList(), Ui_KbAnimWidget::animList, light, noReorder, and ui.

Referenced by KbAnimWidget().

```
61                                {
62     if(light && !noReorder){
63         // Clear and rebuild the list of animations in case the animation moved
64         int count = ui->animList->count();
65         QList<KbAnim*> animList;
66         for(int i = 0; i < count; i++){
67             QListWidgetItem* item = ui->animList->item(i);
68             KbAnim* anim = animations[item->data(Qt::UserRole).toUuid()];
69             if(anim && !animList.contains(anim))
70                 animList.append(anim);
71             item->setFlags(item->flags() | Qt::ItemIsEditable);
72         }
73         light->animList(animList);
74     }
75 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.25.3.17 void KbAnimWidget::setCurrent (KbAnim * *newCurrent*) [private]

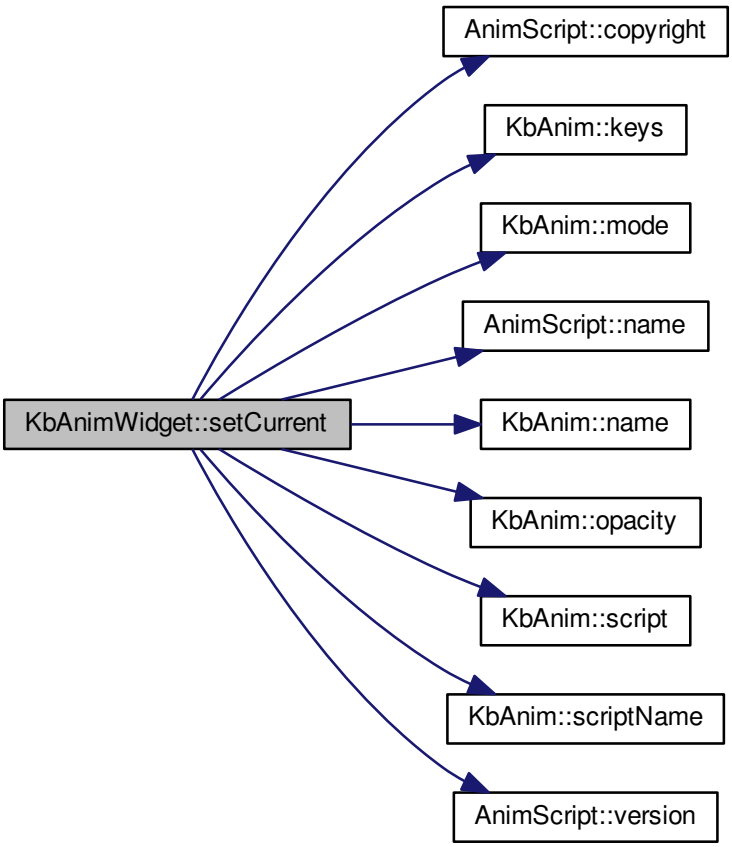
Definition at line 118 of file kbanimwidget.cpp.

References Ui_KbAnimWidget::aCopyLabel, Ui_KbAnimWidget::aMissingLabel, Ui_KbAnimWidget::aNameLabel, animChanged(), Ui_KbAnimWidget::aVerLabel, Ui_KbAnimWidget::blendBox, AnimScript::copyright(), current, KbAnim::keys(), KbAnim::mode(), AnimScript::name(), KbAnim::name(), Ui_KbAnimWidget::nameBox, KbAnim::opacity(), Ui_KbAnimWidget::opacityBox, KbAnim::script(), KbAnim::scriptName(), selectedKeys, Ui_KbAnimWidget::selectionStack, ui, and AnimScript::version().

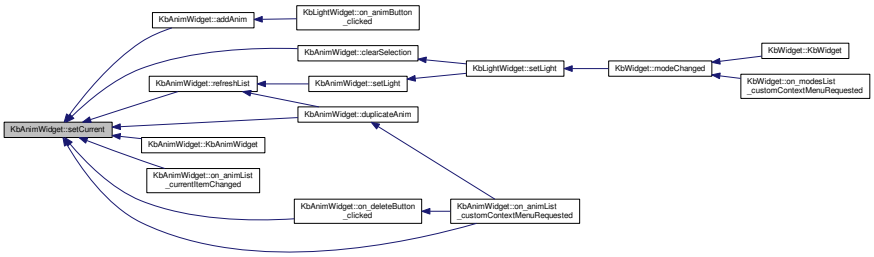
Referenced by addAnim(), clearSelection(), duplicateAnim(), KbAnimWidget(), on_animList_currentItemChanged(), on_animList_customContextMenuRequested(), on_deleteButton_clicked(), and refreshList().

```
118                                                 {
119     if(newCurrent != current)
120         emit animChanged(current = newCurrent);
121     if(!current){
122         selectedKeys.clear();
123         ui->selectionStack->setCurrentIndex(0);
124         return;
125     }
126     selectedKeys = current->keys();
127     const AnimScript* script = current->script();
128     if(!script){
129         ui->selectionStack->setCurrentIndex(2);
130         ui->aMissingLabel->setText("The \"" + current->
      scriptName() + "\" script could not be loaded. Please check your animation directory.");
131         return;
132     }
133     ui->selectionStack->setCurrentIndex(1);
134     ui->aNameLabel->setText(script->name());
135     ui->aVerLabel->setText("v" + script->version());
136     ui->aCopyLabel->setText(script->copyright());
137
138     ui->nameBox->setText(current->name());
139     ui->opacityBox->setValue(current->opacity() * 100.);
140     ui->blendBox->setCurrentIndex((int)current->mode());
141 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.25.3.18 void KbAnimWidget::setLight (KbLight * *newLight*)

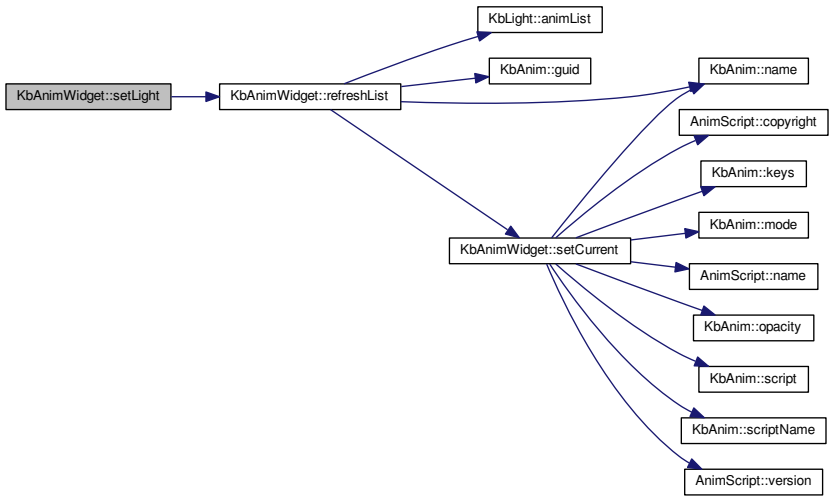
Definition at line 21 of file kbanimwidget.cpp.

References light, and refreshList().

Referenced by KbLightWidget::setLight().

```
21                                          {
22     if(light != newLight){
23         if(light)
24             disconnect(light, SIGNAL(didLoad()), this, SLOT(refreshList()));
25         if(newLight)
26             connect(newLight, SIGNAL(didLoad()), this, SLOT(refreshList()));
27         light = newLight;
28     }
29     refreshList();
30 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.25.3.19 void KbAnimWidget::setSelectedKeys (const QStringList & *keys*)

Definition at line 143 of file kbanimwidget.cpp.

References Ui_KbAnimWidget::keyButton, selectedKeys, and ui.

Referenced by KbLightWidget::newSelection().

```
143                                                     {
144     selectedKeys = keys;
145     if(keys.count() == 0)
146         ui->keyButton->setVisible(false);
147     else
148         ui->keyButton->setVisible(true);
149 }
```

Here is the caller graph for this function:

12.25.4 Field Documentation

12.25.4.1 QHash<QUuid, KbAnim*> KbAnimWidget::animations [private]

Definition at line 53 of file kbanimwidget.h.

Referenced by addAnim(), on_animList_currentItemChanged(), on_animList_customContextMenuRequested(), on_animList_itemChanged(), on_deleteButton_clicked(), refreshList(), and reorderAnims().

12.25.4.2 KbAnim* KbAnimWidget::current [private]

Definition at line 55 of file kbanimwidget.h.

Referenced by on_animList_customContextMenuRequested(), on_animList_itemChanged(), on_blendBox_activated(), on_deleteButton_clicked(), on_keyButton_clicked(), on_nameBox_textEdited(), on_opacityBox_valueChanged(), on_propertyButton_clicked(), and setCurrent().

12.25.4.3 KbLight* KbAnimWidget::light [private]

Definition at line 52 of file kbanimwidget.h.

Referenced by addAnim(), duplicateAnim(), on_deleteButton_clicked(), on_keyButton_clicked(), on_propertyButton_clicked(), refreshList(), reorderAnims(), and setLight().

12.25.4.4 bool KbAnimWidget::noReorder [private]

Definition at line 58 of file kbanimwidget.h.

Referenced by addAnim(), duplicateAnim(), on_animList_itemChanged(), on_nameBox_textEdited(), refreshList(), and reorderAnims().

12.25.4.5 QStringList KbAnimWidget::selectedKeys [private]

Definition at line 57 of file kbanimwidget.h.

Referenced by on_keyButton_clicked(), setCurrent(), and setSelectedKeys().

12.25.4.6 Ui::KbAnimWidget* KbAnimWidget::ui [private]

Definition at line 60 of file kbanimwidget.h.

Referenced by addAnim(), clearSelection(), duplicateAnim(), KbAnimWidget(), on_animList_customContextMenuRequested(), on_animList_itemChanged(), on_deleteButton_clicked(), on_nameBox_textEdited(), on_propertyButton_clicked(), refreshList(), reorderAnims(), setCurrent(), setSelectedKeys(), and ~KbAnimWidget().

The documentation for this class was generated from the following files:

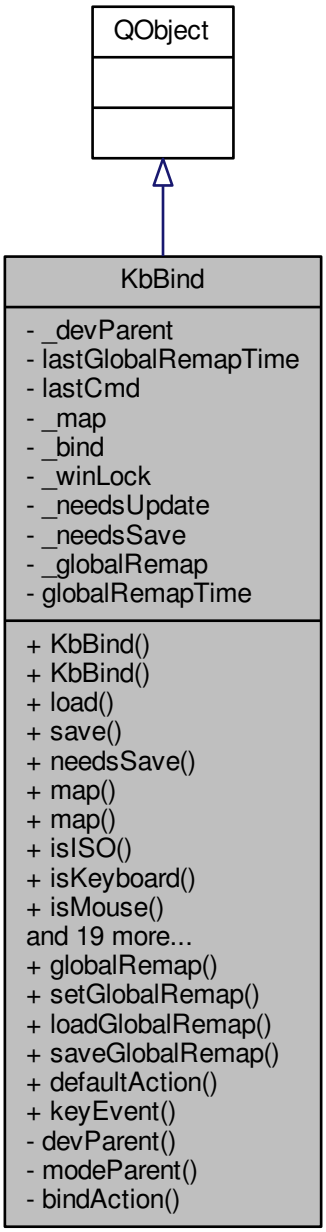
- src/ckb/kbanimwidget.h

- src/ckb/kbanimwidget.cpp
- src/ckb/release/moc_kbanimwidget.cpp

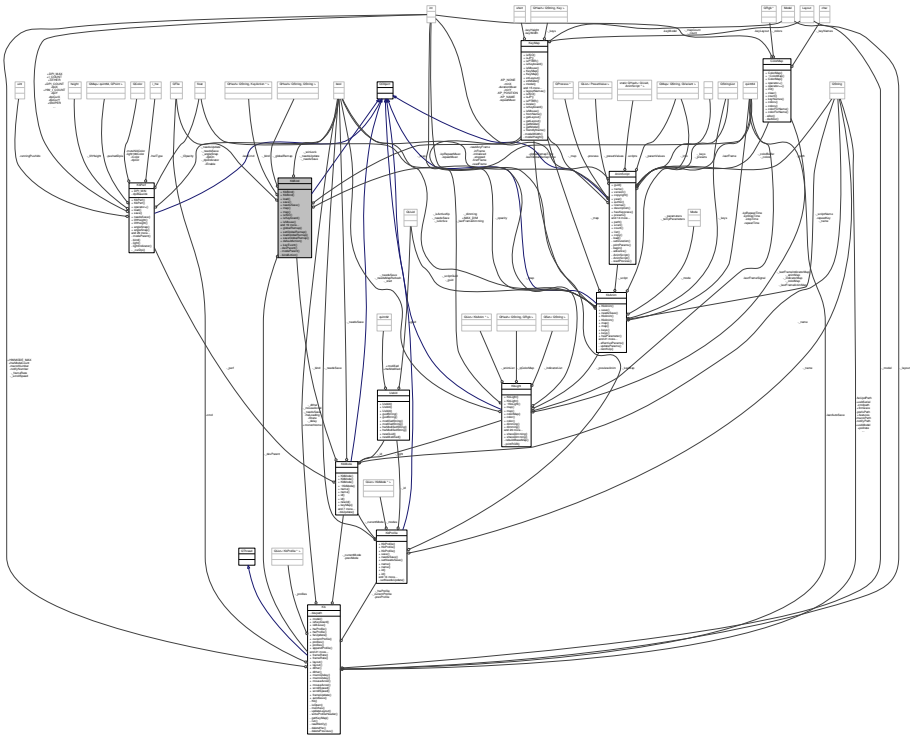
12.26 KbBind Class Reference

`#include <src/ckb/kbbind.h>`

Inheritance diagram for KbBind:



Collaboration diagram for KbBind:



Public Slots

- void keyEvent (const QString &key, bool down)

Signals

- void didLoad ()
- void layoutChanged ()
- void updated ()

Public Member Functions

- KbBind (KbMode *modeParent, Kb *devParent, const KeyMap &keyMap)
- KbBind (KbMode *modeParent, Kb *devParent, const KeyMap &keyMap, const KbBind &other)

  KbBind.
- void load (CkbSettings &settings)
- void save (CkbSettings &settings)
- bool needsSave () const
- const KeyMap & map ()
- void map (const KeyMap &map)
- bool isISO () const
- bool isKeyboard () const
- bool isMouse () const
- KbPerf * perf ()
- KbLight * light ()
- QString action (const QString &key)
- QString defaultAction (const QString &key)

- QString friendlyName (const QString &key)
- QString friendlyActionName (const QString &key)
- void setAction (const QString &key, const QString &action)
- void setAction (const QStringList &keys, const QString &action)
- void resetAction (const QString &key)
- void resetAction (const QStringList &keys)
- void noAction (const QString &key)
- void noAction (const QStringList &keys)
- bool winLock ()
- void winLock (bool newWinLock)
- void update (QFile &cmd, bool force=false)
- void setNeedsUpdate ()
- int getMacroNumber ()

  KbBind::getMacroNumber.
- QString getMacroPath ()

  KbBind::getMacroPath.
- void handleNotificationChannel (bool start)

  handleNotificationChannel sends commands to ckb-daemon for (de-) activating the notify channel. Send a notify cmd to the keyboard to set or clear notification for reading macro definition. The file handle for the cmd pipe is stored in lastCmd.

Static Public Member Functions

- static QString globalRemap (const QString &key)
- static void setGlobalRemap (const QHash< QString, QString > keyToActual)
- static void loadGlobalRemap ()
- static void saveGlobalRemap ()
- static QString defaultAction (const QString &key, KeyMap::Model model)

Private Member Functions

- Kb * devParent () const
- KbMode * modeParent () const
- KeyAction * bindAction (const QString &key)

Private Attributes

- Kb * _devParent
- quint64 lastGlobalRemapTime
- QFile * lastCmd

  lastCmd is a cache-hack. Because the QFile ist opened in Kb, and we need it in the macro processing functions, we cache the value her in lastCmd.
- KeyMap _map
- QHash< QString, KeyAction * > _bind
- bool _winLock
- bool _needsUpdate
- bool _needsSave

Static Private Attributes

- static QHash< QString, QString > _globalRemap
- static quint64 globalRemapTime = 0

Friends

- class KeyAction

12.26.1 Detailed Description

Definition at line 20 of file kbbind.h.

12.26.2 Constructor & Destructor Documentation

12.26.2.1 KbBind::KbBind (KbMode * *modeParent*, Kb * *devParent*, const KeyMap & *keyMap*) [explicit]

Definition at line 12 of file kbbind.cpp.

```
12                                                                          :
13     QObject(modeParent), _devParent(parentBoard),
       lastGlobalRemapTime(globalRemapTime), _map(keyMap),
14     _winLock(false), _needsUpdate(true), _needsSave(true) {
15 }
```

12.26.2.2 KbBind::KbBind (KbMode * *modeParent*, Kb * *parentBoard*, const KeyMap & *keyMap*, const KbBind & *other*)

KbBind::KbBind // copy all existing Key bindings.

Parameters

modeParent	
devParent	
keyMap	
other	This is the KbBind object to copy from Use this constructor to copy an existing Binding
modeParent	
parentBoard	
keyMap	
other	

Create a new Hash table and copy all entries

clear the destination list (there are the original KeyActions as references, so do not delete them)

and move the KeyActions we just created

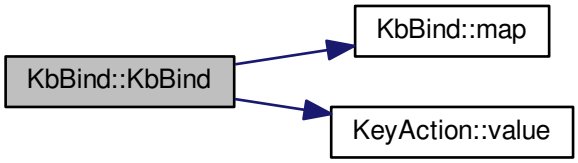
Definition at line 24 of file kbbind.cpp.

References _bind, KeyAction, map(), and KeyAction::value().

```
24                                                                                  :
25     QObject(modeParent), _devParent(parentBoard),
       lastGlobalRemapTime(globalRemapTime), _bind(other.
       _bind),
26     _winLock(false), _needsUpdate(true), _needsSave(true) {
27     map(keyMap);
28
30     QHash<QString, KeyAction*> newBind;
31     foreach(QString key, _bind.keys()) {
32         KeyAction* act = _bind.value(key);
33         if(act) {
34             newBind[key] = new KeyAction(act->value(), this);
35         }
36     }
37
39     _bind.clear();
40     foreach(QString key, newBind.keys()) {
41         KeyAction* act = newBind.value(key);
42         if(act) {
44             _bind[key] = new KeyAction(act->value(), this);
45         }
46     }
```

```
47     newBind.clear();      // here we *must not* delete the KeyActions, because they are referenced by _bind
       now
48 }
```

Here is the call graph for this function:



12.26.3 Member Function Documentation

12.26.3.1 QString KbBind::action (const QString & *key*)

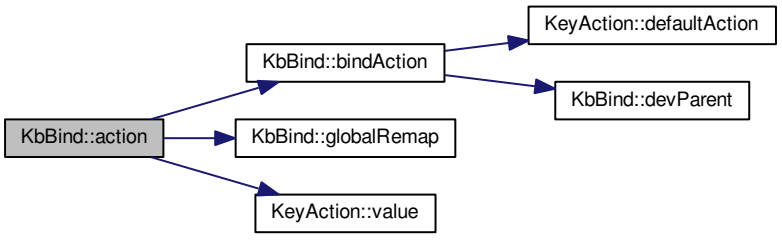
Definition at line 145 of file kbbind.cpp.

References bindAction(), globalRemap(), and KeyAction::value().

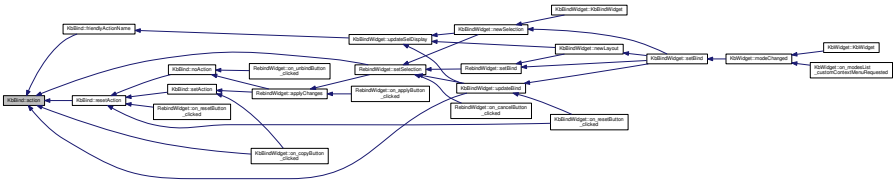
Referenced by friendlyActionName(), KbBindWidget::on_copyButton_clicked(), resetAction(), RebindWidget::setSelection(), and KbBindWidget::updateBind().

```
145                                         {
146     QString rKey = globalRemap(key);
147     return bindAction(rKey)->value();
148 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.2 KeyAction * KbBind::bindAction (const QString & *key*) [private]

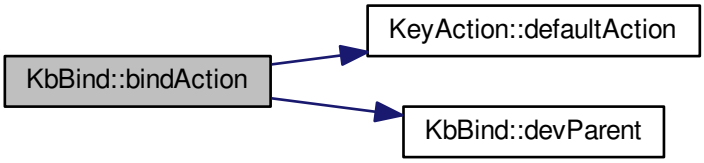
Definition at line 140 of file kbbind.cpp.

References _bind, KeyAction::defaultAction(), devParent(), and KeyAction.

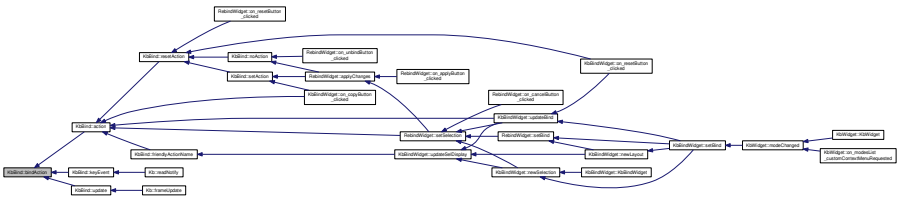
Referenced by action(), keyEvent(), and update().

```
140                                                 {
141     if(!_bind.contains(key)) return _bind[key] = new KeyAction(
        KeyAction::defaultAction(key, devParent()->model()), this);
142     return _bind[key];
143 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.3 QString KbBind::defaultAction (const QString & *key*)

Definition at line 155 of file kbbind.cpp.

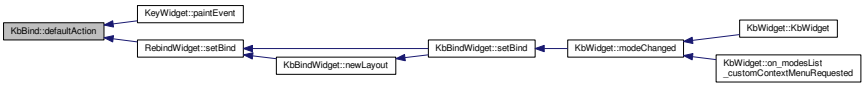
References devParent().

Referenced by KeyWidget::paintEvent(), and RebindWidget::setBind().

```
155                                                 {
156     return defaultAction(key, devParent()->model());
157 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



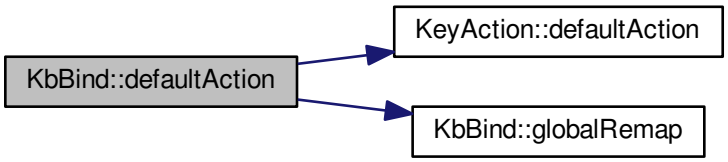
12.26.3.4 QString KbBind::defaultAction (const QString & *key,* KeyMap::Model *model*) [static]

Definition at line 150 of file kbbind.cpp.

References KeyAction::defaultAction(), and globalRemap().

```
150                                                                 {
151     QString rKey = globalRemap(key);
152     return KeyAction::defaultAction(rKey, model);
153 }
```

Here is the call graph for this function:



12.26.3.5 Kb* KbBind::devParent ( ) const [inline], [private]

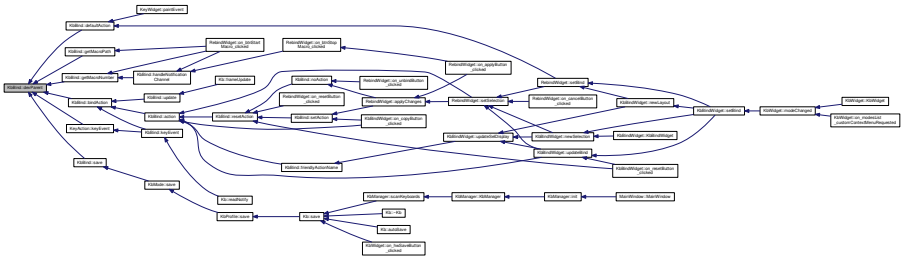
Definition at line 115 of file kbbind.h.

References _devParent.

Referenced by bindAction(), defaultAction(), getMacroNumber(), getMacroPath(), KeyAction::keyEvent(), and save().

```
115 { return _devParent; }
```

Here is the caller graph for this function:



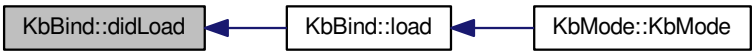
12.26.3.6 void KbBind::didLoad ( ) [signal]

Definition at line 155 of file moc_kbbind.cpp.

Referenced by load().

```
156 {
157     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
158 }
```

Here is the caller graph for this function:



12.26.3.7 QString KbBind::friendlyActionName (const QString & *key*)

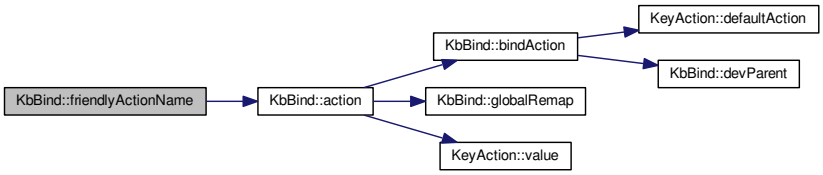
Definition at line 166 of file kbbind.cpp.

References _map, action(), and KeyAction.

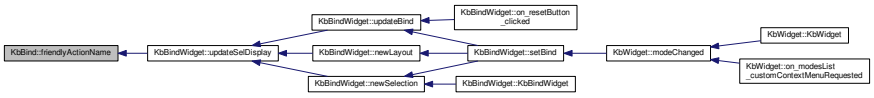
Referenced by KbBindWidget::updateSelDisplay().

```
166                                                 {
167     QString act = action(key);
168     return KeyAction(act).friendlyName(_map);
169 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



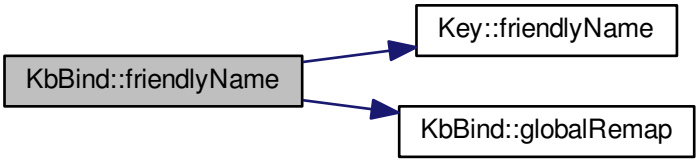
12.26.3.8 QString KbBind::friendlyName (const QString & *key*)

Definition at line 159 of file kbbind.cpp.

References _map, Key::friendlyName(), and globalRemap().

```
159                                                 {
160     const Key& pos = _map[globalRemap(key)];
161     if(!pos)
162         return "(Unknown)";
163     return pos.friendlyName();
164 }
```

Here is the call graph for this function:



12.26.3.9 int KbBind::getMacroNumber ( )

Returns

number of notification channel. Use it in combination with notifyon/off-Statement

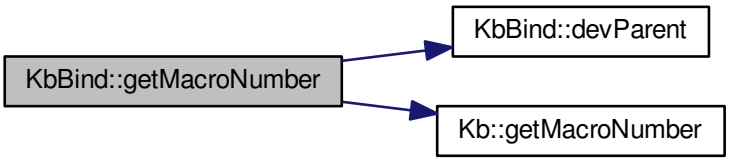
Definition at line 266 of file kbbind.cpp.

References devParent(), and Kb::getMacroNumber().

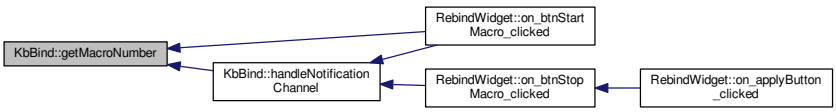
Referenced by handleNotificationChannel(), and RebindWidget::on_btnStartMacro_clicked().

```
266                             {
267     return devParent()->getMacroNumber();
268 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.10 QString KbBind::getMacroPath ( )

Returns

Filepath of macro notification pipe. If not set, returns initial value ""

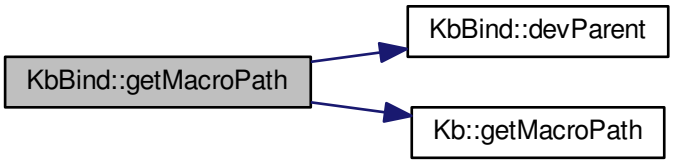
Definition at line 274 of file kbbind.cpp.

References devParent(), and Kb::getMacroPath().

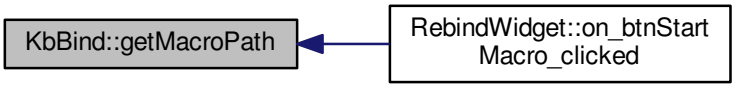
Referenced by RebindWidget::on_btnStartMacro_clicked().

```
274                             {
275     return devParent()->getMacroPath();
276 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.11 QString KbBind::globalRemap (const QString & *key*) [static]

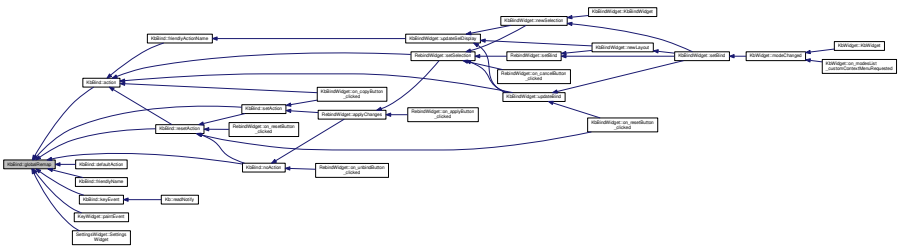
Definition at line 98 of file kbbind.cpp.

References _globalRemap.

Referenced by action(), defaultAction(), friendlyName(), keyEvent(), noAction(), KeyWidget::paintEvent(), resetAction(), setAction(), and SettingsWidget::SettingsWidget().

```
98                                                 {
99     if(!_globalRemap.contains(key))
100         return key;
101     return _globalRemap.value(key);
102 }
```

Here is the caller graph for this function:



12.26.3.12 void KbBind::handleNotificationChannel (bool *start*)

Parameters

start	is boolean. If true, notification channel is opened for all keys, otherwise channel ist closed.
start	If true, notification channel is opened for all keys, otherwise channel ist closed.

Definition at line 284 of file kbbind.cpp.

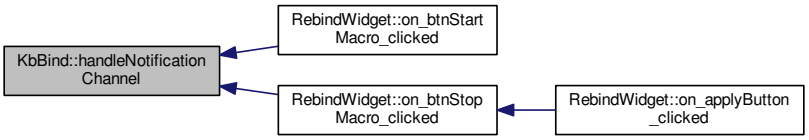
References getMacroNumber(), and lastCmd.

Referenced by RebindWidget::on_btnStartMacro_clicked(), and RebindWidget::on_btnStopMacro_clicked().

```
284                                                 {
285     if (getMacroNumber() > 0 && lastCmd) {
286         if (start) {
287             lastCmd->write (QString("\nnotifyon %1\n@%1 notify all:on\n").arg(
    getMacroNumber()).toLatin1());
288         } else {
289             lastCmd->write (QString("\n@%1 notify all:off\nnotifyoff %1\n").arg(
    getMacroNumber()).toLatin1());
290         }
291         lastCmd->flush();
292     } else qDebug() << QString("No cmd or valid handle for notification found, macroNumber = %1, lastCmd =
     %2")
293                        .arg(getMacroNumber()).arg(lastCmd? "set" : "unset");
294 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.26.3.13 bool KbBind::isISO ( ) const [inline]

Definition at line 43 of file kbbind.h.

References _map, and KeyMap::isISO().

```
43 { return _map.isISO(); }
```

Here is the call graph for this function:

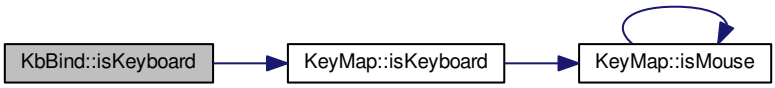
12.26.3.14 bool KbBind::isKeyboard ( ) const [inline]

Definition at line 44 of file kbbind.h.

References _map, and KeyMap::isKeyboard().

```
44 { return _map.isKeyboard(); }
```

Here is the call graph for this function:



12.26.3.15 bool KbBind::isMouse ( ) const [inline]

Definition at line 45 of file kbbind.h.

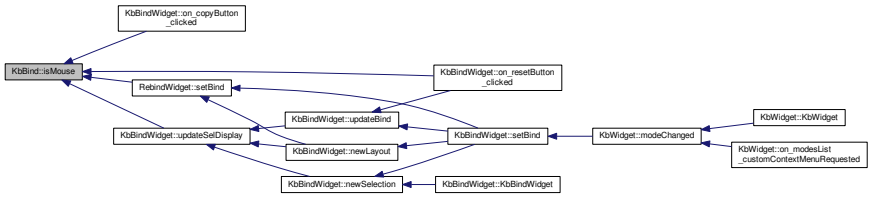
References _map, and KeyMap::isMouse().

Referenced by KbBindWidget::on_copyButton_clicked(), KbBindWidget::on_resetButton_clicked(), RebindWidget::setBind(), and KbBindWidget::updateSelDisplay().

```
45 { return _map.isMouse(); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.26.3.16 void KbBind::keyEvent (const QString & *key*, bool *down*) [slot]

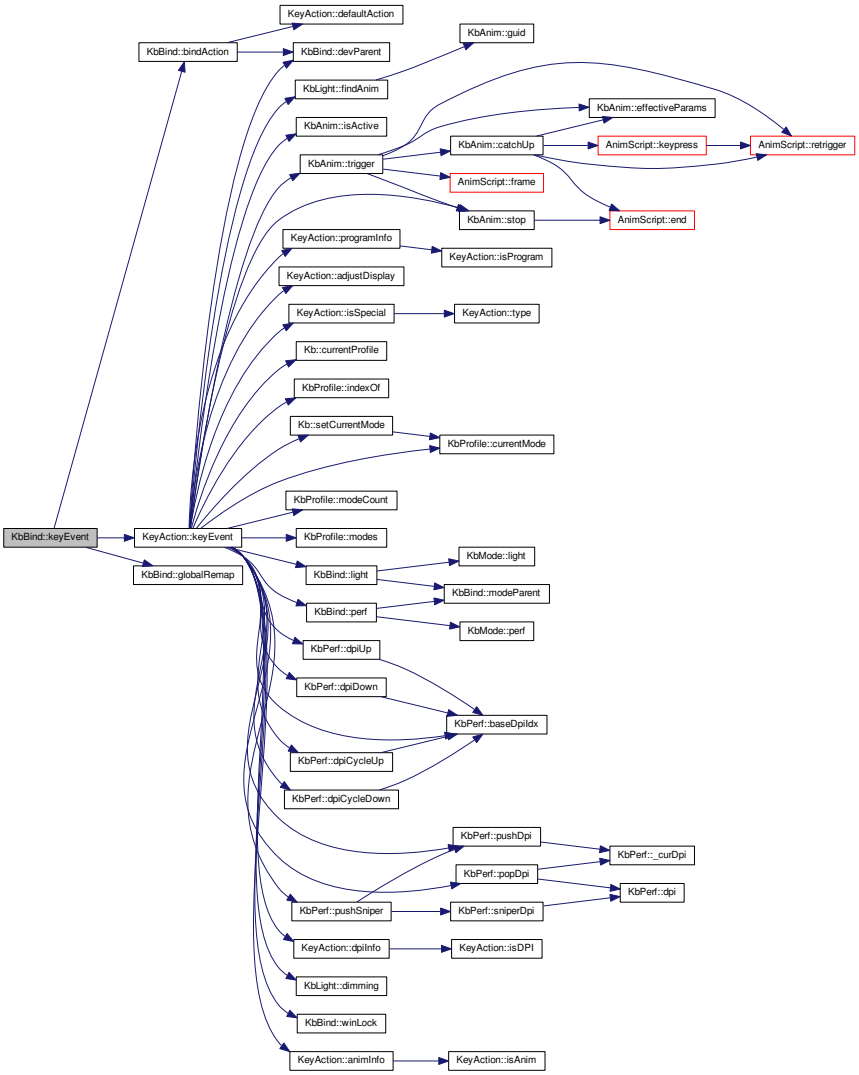
Definition at line 296 of file kbbind.cpp.

References bindAction(), globalRemap(), and KeyAction::keyEvent().

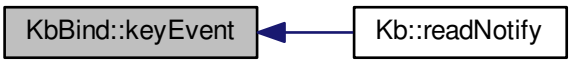
Referenced by Kb::readNotify().

```
296                                                 {
297     QString rKey = globalRemap(key);
298     KeyAction* act = bindAction(rKey);
299     if(act)
300         act->keyEvent(this, down);
301 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



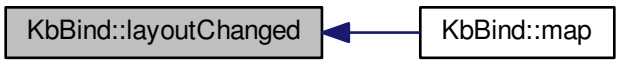
12.26.3.17 void KbBind::layoutChanged ( ) [signal]

Definition at line 161 of file moc_kbbind.cpp.

Referenced by map().

```
162 {
163     QMetaObject::activate(this, &staticMetaObject, 1, Q_NULLPTR);
164 }
```

Here is the caller graph for this function:



12.26.3.18 KbLight * KbBind::light ( )

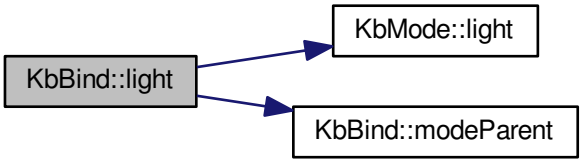
Definition at line 54 of file kbbind.cpp.

References KbMode::light(), and modeParent().

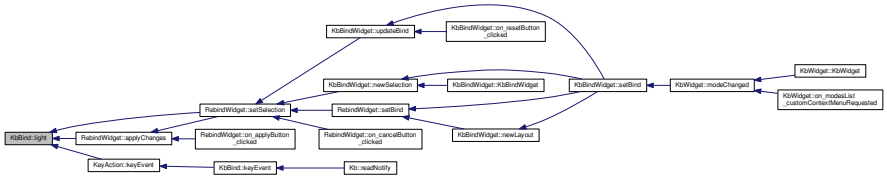
Referenced by RebindWidget::applyChanges(), KeyAction::keyEvent(), and RebindWidget::setSelection().

```
54                       {
55     return modeParent()->light();
56 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.19 void KbBind::load (CkbSettings & *settings*)

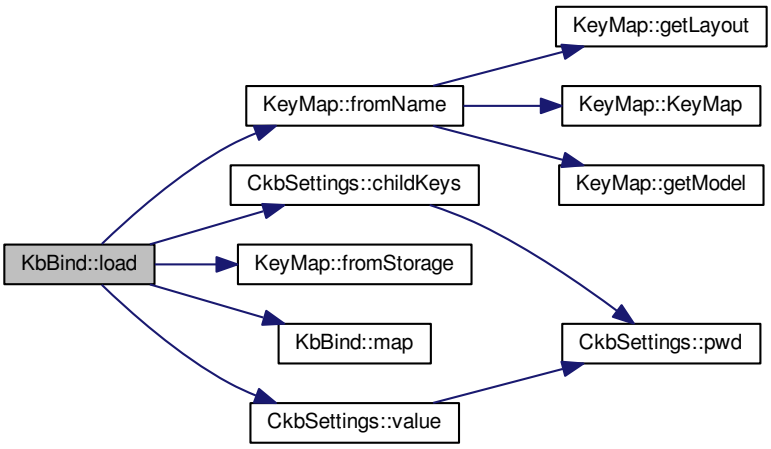
Definition at line 58 of file kbbind.cpp.

References _bind, _map, _needsSave, _winLock, CkbSettings::childKeys(), didLoad(), KeyMap::fromName(), KeyMap::fromStorage(), KeyAction, map(), and CkbSettings::value().

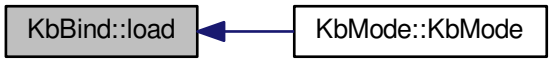
Referenced by KbMode::KbMode().

```
58                                      {
59     _needsSave = false;
60     SGroup group(settings, "Binding");
61     KeyMap currentMap = _map;
62     _map = KeyMap::fromName(settings.value("KeyMap").toString());
63     // Load key settings
64     bool useReal = settings.value("UseRealNames").toBool();
65     _bind.clear();
66     {
67         SGroup group(settings, "Keys");
68         foreach(QString key, settings.childKeys()){
69             QString name = key.toLower();
70             if(!useReal)
71                 name = _map.fromStorage(name);
72             QString bind = settings.value(key).toString();
73             _bind[name] = new KeyAction(bind, this);
74         }
75     }
76     _winLock = settings.value("WinLock").toBool();
77     emit didLoad();
78     map(currentMap);
79 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.20 void KbBind::loadGlobalRemap ( ) [static]

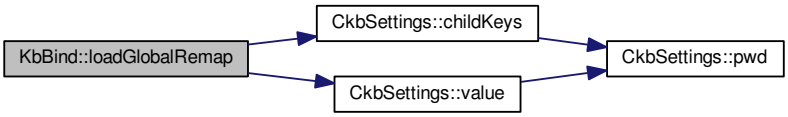
Definition at line 116 of file kbbind.cpp.

References _globalRemap, CkbSettings::childKeys(), globalRemapTime, and CkbSettings::value().

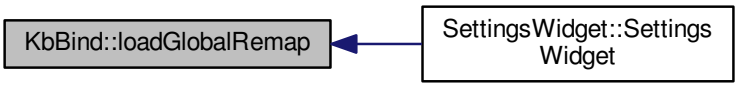
Referenced by SettingsWidget::SettingsWidget().

```
116                             {
117     _globalRemap.clear();
118     CkbSettings settings("Program/GlobalRemap");
119     foreach(const QString& key, settings.childKeys())
120         _globalRemap[key] = settings.value(key).toString();
121     globalRemapTime = QDateTime::currentMSecsSinceEpoch();
122 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.21 const KeyMap& KbBind::map ( ) [inline]

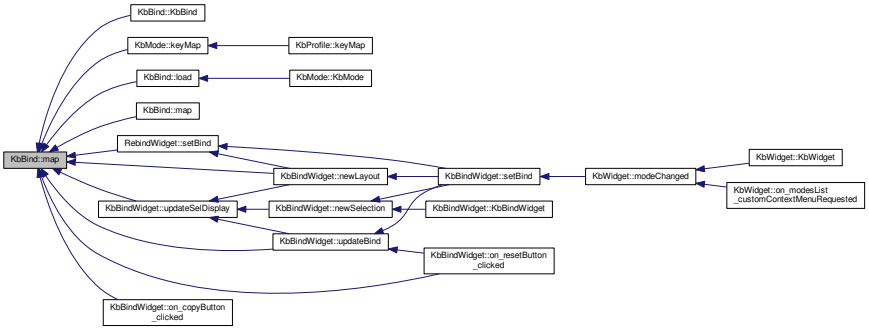
Definition at line 41 of file kbbind.h.

References _map.

Referenced by KbBind(), KbMode::keyMap(), load(), map(), KbBindWidget::newLayout(), KbBindWidget::on_copyButton_clicked(), KbBindWidget::on_resetButton_clicked(), RebindWidget::setBind(), KbBindWidget::updateBind(), and KbBindWidget::updateSelDisplay().

```
41 { return _map; }
```

Here is the caller graph for this function:



12.26.3.22 void KbBind::map (const KeyMap & *map*)

Definition at line 133 of file kbbind.cpp.

References _map, _needsSave, _needsUpdate, layoutChanged(), and map().

```
133                                  {
134     _map = map;
135     _needsUpdate = true;
136     _needsSave = true;
137     emit layoutChanged();
138 }
```

Here is the call graph for this function:

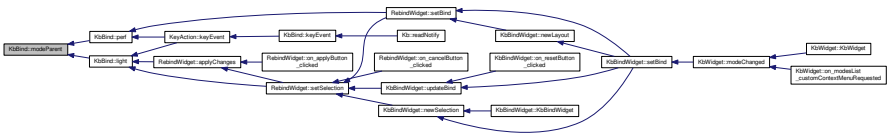
12.26.3.23 KbMode* KbBind::modeParent ( ) const [inline], [private]

Definition at line 116 of file kbbind.h.

Referenced by light(), and perf().

```
116 { return (KbMode*)parent(); }
```

Here is the caller graph for this function:



12.26.3.24 bool KbBind::needsSave ( ) const [inline]

Definition at line 38 of file kbbind.h.

References _needsSave.

Referenced by KbMode::needsSave().

```
38 { return _needsSave; }
```

Here is the caller graph for this function:

12.26.3.25 void KbBind::noAction (const QString & *key*)

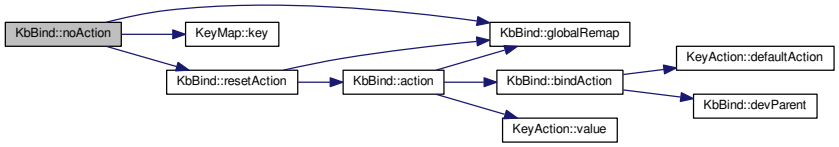
Definition at line 181 of file kbbind.cpp.

References _bind, _map, globalRemap(), KeyMap::key(), KeyAction, and resetAction().

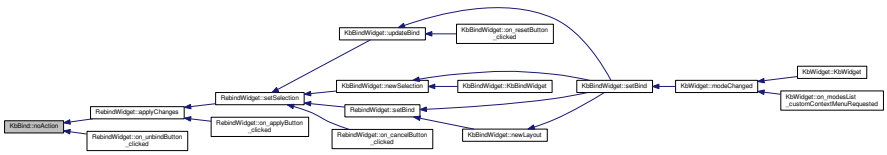
Referenced by RebindWidget::applyChanges(), and RebindWidget::on_unbindButton_clicked().

```
181                                         {
182     resetAction(key);
183     QString rKey = globalRemap(key);
184     if(!_map.key(rKey))
185         return;
186     _bind[rKey] = new KeyAction(this);
187 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.26 void KbBind::noAction (const QStringList & *keys*) [inline]

Definition at line 76 of file kbbind.h.

References noAction().

Referenced by noAction().

```
76 { foreach(const QString& key, keys) noAction(key); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.26.3.27 KbPerf * KbBind::perf ( )

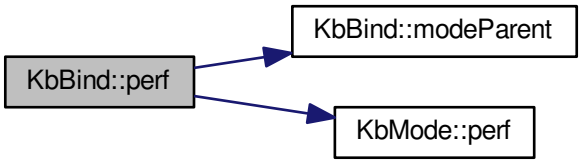
Definition at line 50 of file kbbind.cpp.

References modeParent(), and KbMode::perf().

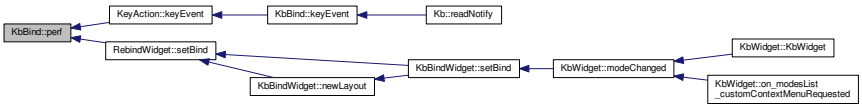
Referenced by KeyAction::keyEvent(), and RebindWidget::setBind().

```
50                     {
51     return modeParent()->perf();
52 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.28 void KbBind::resetAction (const QString & *key*)

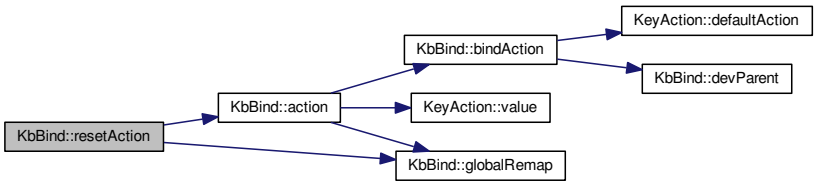
Definition at line 171 of file kbbind.cpp.

References _bind, _needsSave, _needsUpdate, action(), and globalRemap().

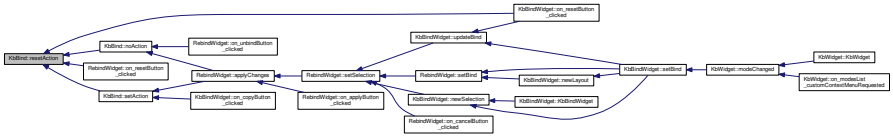
Referenced by noAction(), KbBindWidget::on_resetButton_clicked(), RebindWidget::on_resetButton_clicked(), and setAction().

```
171                                          {
172     QString rKey = globalRemap(key);
173     // Clean up existing action (if any)
174     KeyAction* action = _bind.value(rKey);
175     delete action;
176     _bind.remove(rKey);
177     _needsUpdate = true;
178     _needsSave = true;
179 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.29 void KbBind::resetAction (const QStringList & *keys*) [inline]

Definition at line 73 of file kbbind.h.

References resetAction().

Referenced by resetAction().

```
73 { foreach(const QString& key, keys) resetAction(key); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.26.3.30 void KbBind::save (CkbSettings & *settings*)

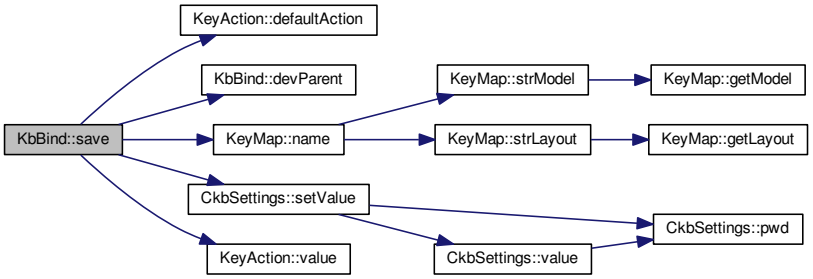
Definition at line 81 of file kbbind.cpp.

References _bind, _map, _needsSave, _winLock, KeyAction::defaultAction(), devParent(), KeyMap::name(), CkbSettings::setValue(), and KeyAction::value().

Referenced by KbMode::save().

```
81                                      {
82     _needsSave = false;
83     SGroup group(settings, "Binding");
84     settings.setValue("KeyMap", _map.name());
85     // Save key settings
86     settings.setValue("UseRealNames", true);
87     {
88         SGroup group(settings, "Keys");
89         foreach(QString key, _bind.keys()){
90             KeyAction* act = _bind.value(key);
91             if(act && act->value() != KeyAction::defaultAction(key,
     devParent()->model()))
92                 settings.setValue(key, act->value());
93         }
94     }
95     settings.setValue("WinLock", _winLock);
96 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.26.3.31 void KbBind::saveGlobalRemap ( ) [static]

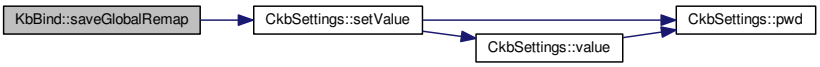
Definition at line 124 of file kbbind.cpp.

References _globalRemap, and CkbSettings::setValue().

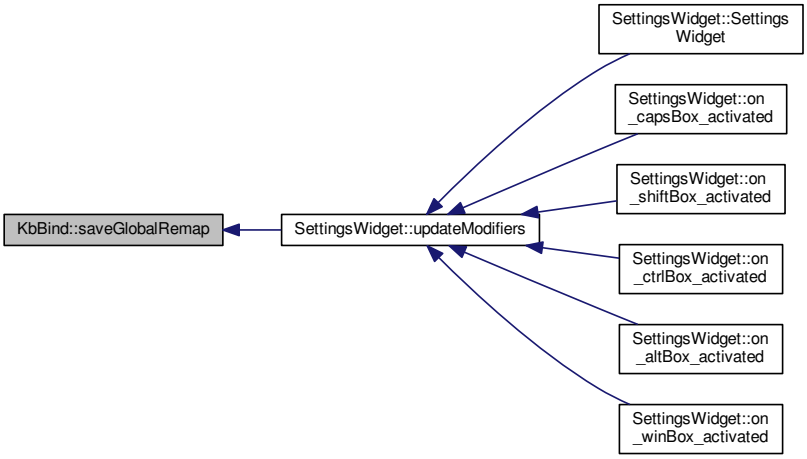
Referenced by SettingsWidget::updateModifiers().

```
124                             {
125     CkbSettings settings("Program/GlobalRemap", true);
126     QHashIterator<QString, QString> i(_globalRemap);
127     while(i.hasNext()){
128         i.next();
129         settings.setValue(i.key(), i.value());
130     }
131 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.32 void KbBind::setAction (const QString & *key,* const QString & *action*)

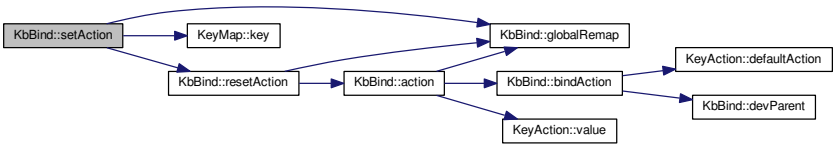
Definition at line 189 of file kbbind.cpp.

References _bind, _map, globalRemap(), KeyMap::key(), KeyAction, and resetAction().

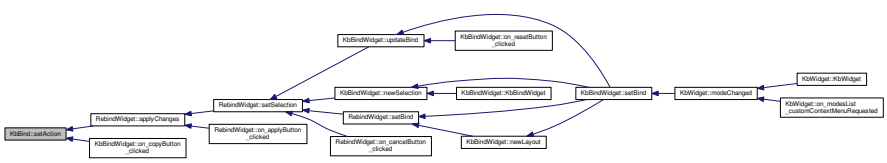
Referenced by RebindWidget::applyChanges(), and KbBindWidget::on_copyButton_clicked().

```
189                                                                 {
190     resetAction(key);
191     QString rKey = globalRemap(key);
192     if(!_map.key(rKey))
193         return;
194     _bind[rKey] = new KeyAction(action, this);
195 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.26.3.33 void KbBind::setAction (const QStringList & *keys*, const QString & *action*) [inline]

Definition at line 70 of file kbbind.h.

References setAction().

Referenced by setAction().

```
70 { foreach(const QString& key, keys) setAction(key, action); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.26.3.34 void KbBind::setGlobalRemap (const QHash< QString, QString > *keyToActual*) [static]

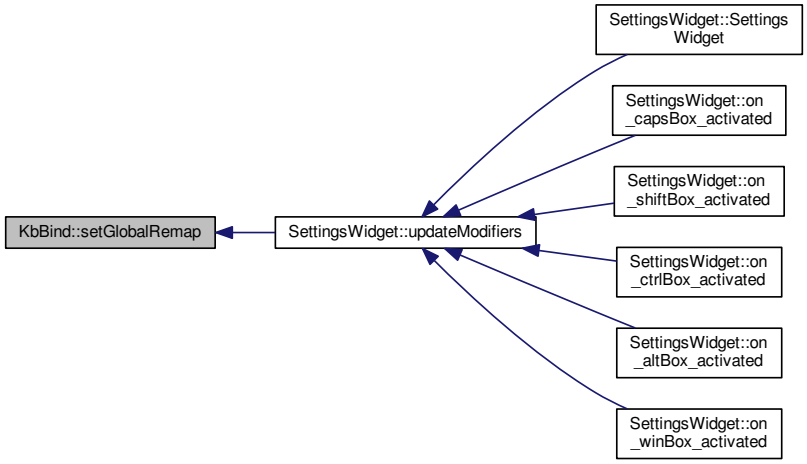
Definition at line 104 of file kbbind.cpp.

References _globalRemap, and globalRemapTime.

Referenced by SettingsWidget::updateModifiers().

```
104                                                                    {
105     _globalRemap.clear();
106     // Ignore any keys with the standard binding
107     QHashIterator<QString, QString> i(keyToActual);
108     while(i.hasNext()){
109         i.next();
110         if(i.key() != i.value())
111             _globalRemap[i.key()] = i.value();
112     }
113     globalRemapTime = QDateTime::currentMSecsSinceEpoch();
114 }
```

Here is the caller graph for this function:



12.26.3.35 void KbBind::setNeedsUpdate ( ) [inline]

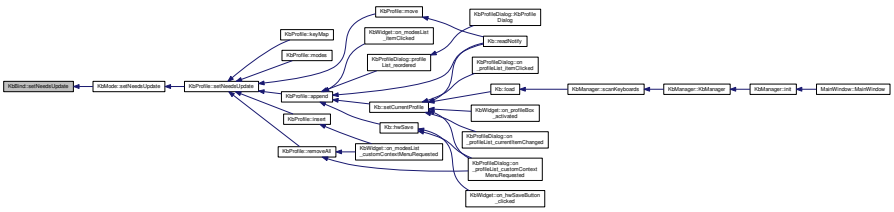
Definition at line 85 of file kbbind.h.

References _needsUpdate.

Referenced by KbMode::setNeedsUpdate().

```
85 { _needsUpdate = true; }
```

Here is the caller graph for this function:



12.26.3.36 void KbBind::update (QFile & *cmd,* bool *force* = false)

Definition at line 197 of file kbbind.cpp.

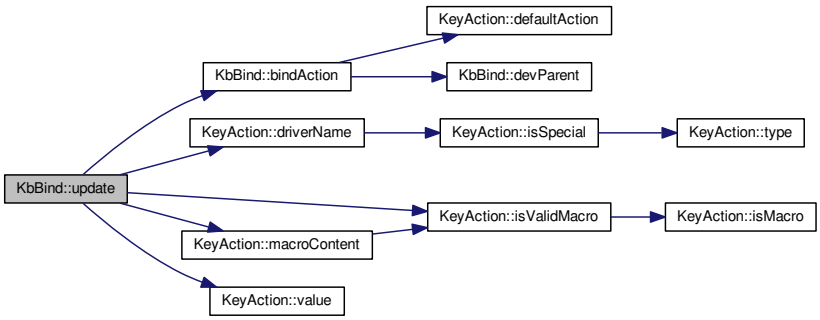
References _bind, _globalRemap, _needsUpdate, _winLock, bindAction(), KeyAction::driverName(), globalRemapTime, KeyAction::isValidMacro(), lastCmd, lastGlobalRemapTime, KeyAction::macroContent(), updated(), and KeyAction::value().

Referenced by Kb::frameUpdate().

```
197                                        {
198     if(!force && !_needsUpdate && lastGlobalRemapTime ==
      globalRemapTime)
199         return;
```

```
200     lastGlobalRemapTime = globalRemapTime;
201     emit updated();
202     _needsUpdate = false;
203     // Reset all keys and enable notifications for all
204     cmd.write("rebind all notify all");
205     // Make sure modifier keys are included as they may be remapped globally
206     QHash<QString, KeyAction*> bind(_bind);
207     if(!_bind.contains("caps")) bind["caps"] = 0;
208     if(!_bind.contains("lshift")) bind["lshift"] = 0;
209     if(!_bind.contains("rshift")) bind["rshift"] = 0;
210     if(!_bind.contains("lctrl")) bind["lctrl"] = 0;
211     if(!_bind.contains("rctrl")) bind["rctrl"] = 0;
212     if(!_bind.contains("lwin")) bind["lwin"] = 0;
213     if(!_bind.contains("rwin")) bind["rwin"] = 0;
214     if(!_bind.contains("lalt")) bind["lalt"] = 0;
215     if(!_bind.contains("ralt")) bind["ralt"] = 0;
216     if(!_bind.contains("fn")) bind["fn"] = 0;
217     QHashIterator<QString, KeyAction*> i(bind);
218
219     // Initialize String buffer for macro Key definitions (G-keys)
220     // "macro clear" is neccessary, if an older definition is unbound.
221     QString macros = "\nmacro clear\n";
222
223     // Write out rebound keys and collect infos for macro definitions
224     while(i.hasNext()){
225         i.next();
226         QString key = i.key();
227         KeyAction* act = i.value();
228         if(_globalRemap.contains(key))
229             act = bindAction(_globalRemap.value(key));
230         if(!act)
231             continue;
232         QString value = act->driverName();
233         if(value.isEmpty()){
234             // If the key is unbound or is a special action, unbind it
235             cmd.write(" unbind ");
236             cmd.write(key.toLatin1());
237             // if a macro definiton for the key is given,
238             // add the converted string to key-buffer "macro"
239             if (act->isValidMacro()) {
240                 if (act->macroContent().length() > 0) {
241                     macros.append("macro " + key.toLatin1() + ":" + act->
    macroContent().toLatin1() + "\n");
242                 }
243             }
244         } else {
245             // Otherwise, write the binding
246             cmd.write(" bind ");
247             cmd.write(key.toLatin1());
248             cmd.write(":");
249             cmd.write(value.toLatin1());
250         }
251     }
252     // If win lock is enabled, unbind windows keys
253     if(_winLock)
254         cmd.write(" unbind lwin rwin");
255
256     // At last, send Macro definitions if available.
257     // If no definitions are made, clear macro will be sent only to reset all macros,
258     cmd.write(macros.toLatin1());
259     lastCmd = &cmd;
260 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.26.3.37 void KbBind::updated ( ) [signal]

Definition at line 167 of file moc_kbbind.cpp.

Referenced by update().

```
168 {
169     QMetaObject::activate(this, &staticMetaObject, 2, Q_NULLPTR);
170 }
```

Here is the caller graph for this function:

12.26.3.38 bool KbBind::winLock ( ) [inline]

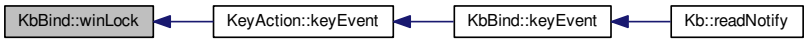
Definition at line 79 of file kbbind.h.

References _winLock.

Referenced by KeyAction::keyEvent().

```
79 { return _winLock; }
```

Here is the caller graph for this function:



12.26.3.39 void KbBind::winLock (bool *newWinLock*) [inline]

Definition at line 80 of file kbbind.h.

References _needsUpdate, and _winLock.

```
80 { _winLock = newWinLock; _needsUpdate = true; }
```

12.26.4 Friends And Related Function Documentation

12.26.4.1 friend class KeyAction [friend]

Definition at line 137 of file kbbind.h.

Referenced by bindAction(), friendlyActionName(), KbBind(), load(), noAction(), and setAction().

12.26.5 Field Documentation

12.26.5.1 QHash<QString, KeyAction*> KbBind::_bind [private]

Definition at line 132 of file kbbind.h.

Referenced by bindAction(), KbBind(), load(), noAction(), resetAction(), save(), setAction(), and update().

12.26.5.2 Kb* KbBind::_devParent [private]

Definition at line 114 of file kbbind.h.

Referenced by devParent().

12.26.5.3 QHash< QString, QString > KbBind::_globalRemap [static], [private]

Definition at line 120 of file kbbind.h.

Referenced by globalRemap(), loadGlobalRemap(), saveGlobalRemap(), setGlobalRemap(), and update().

12.26.5.4 KeyMap KbBind::_map [private]

Definition at line 130 of file kbbind.h.

Referenced by friendlyActionName(), friendlyName(), isISO(), isKeyboard(), isMouse(), load(), map(), noAction(), save(), and setAction().

12.26.5.5 bool KbBind::_needsSave [private]

Definition at line 136 of file kbbind.h.

Referenced by load(), map(), needsSave(), resetAction(), and save().

12.26.5.6 bool KbBind::_needsUpdate [private]

Definition at line 135 of file kbbind.h.

Referenced by map(), resetAction(), setNeedsUpdate(), update(), and winLock().

12.26.5.7 bool KbBind::_winLock [private]

Definition at line 134 of file kbbind.h.

Referenced by load(), save(), update(), and winLock().

12.26.5.8 quint64 KbBind::globalRemapTime = 0 [static], [private]

Definition at line 121 of file kbbind.h.

Referenced by loadGlobalRemap(), setGlobalRemap(), and update().

12.26.5.9 QFile* KbBind::lastCmd [private]

Definition at line 128 of file kbbind.h.

Referenced by handleNotificationChannel(), and update().

12.26.5.10 quint64 KbBind::lastGlobalRemapTime [private]

Definition at line 122 of file kbbind.h.

Referenced by update().

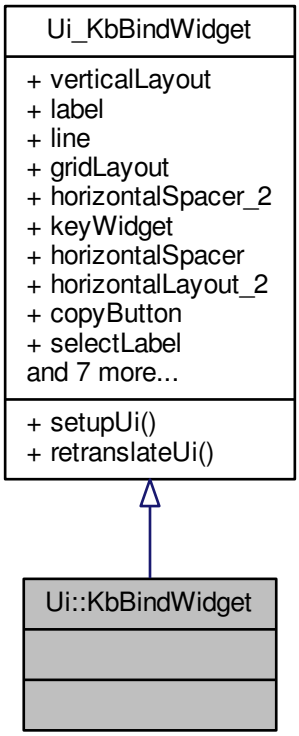
The documentation for this class was generated from the following files:

- src/ckb/kbbind.h
- src/ckb/kbbind.cpp
- src/ckb/release/moc_kbbind.cpp

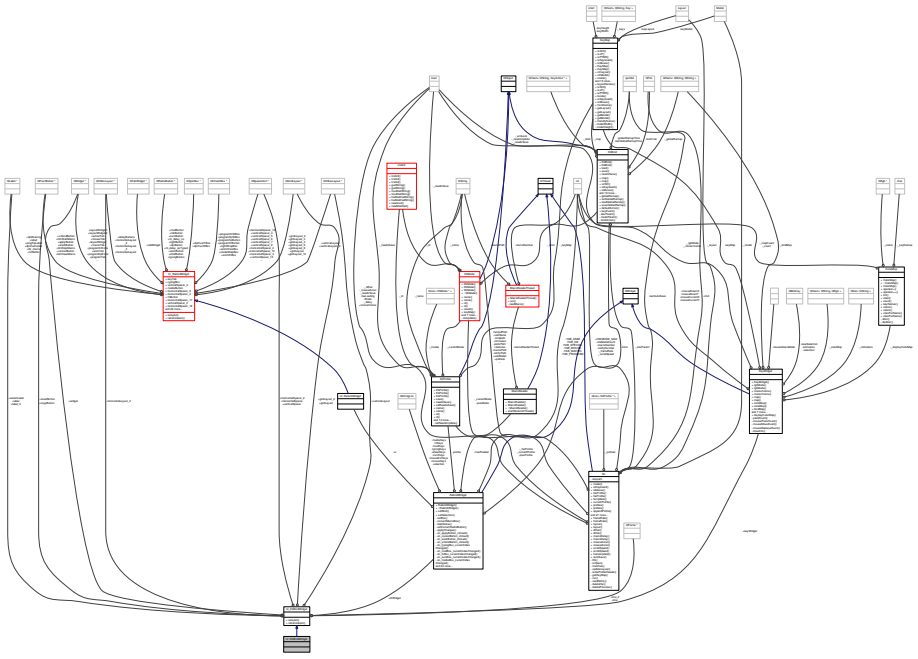
12.27 Ui::KbBindWidget Class Reference

`#include <src/ckb/ui_kbbindwidget.h>`

Inheritance diagram for Ui::KbBindWidget:



Collaboration diagram for Ui::KbBindWidget:



Public Member Functions

- void setupUi (QWidget *KbBindWidget)
- void retranslateUi (QWidget *KbBindWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QLabel * label
- QFrame * line
- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer_2
- KeyWidget * keyWidget
- QSpacerItem * horizontalSpacer
- QHBoxLayout * horizontalLayout_2
- QPushButton * copyButton
- QLabel * selectLabel
- QPushButton * resetButton
- QSpacerItem * verticalSpacer
- QLabel * label_2
- QFrame * line_2
- QWidget * widget
- QGridLayout * gridLayout_2
- RebindWidget * rbWidget

12.27.1 Detailed Description

Definition at line 202 of file ui_kbbindwidget.h.

12.27.2 Member Function Documentation

12.27.2.1 void Ui_KbBindWidget::retranslateUi (QWidget * *KbBindWidget*) [inline], [inherited]

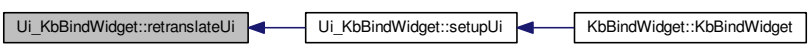
Definition at line 189 of file ui_kbbindwidget.h.

References Ui_KbBindWidget::copyButton, Ui_KbBindWidget::label, Ui_KbBindWidget::label_2, Ui_KbBindWidget::resetButton, and Ui_KbBindWidget::selectLabel.

Referenced by Ui_KbBindWidget::setupUi().

```
190     {
191         KbBindWidget->setWindowTitle(QApplication::translate("KbBindWidget", "Form", Q_NULLPTR));
192         label->setText(QApplication::translate("KbBindWidget", "Keys", Q_NULLPTR));
193         copyButton->setText(QApplication::translate("KbBindWidget", "Copy to mode...", Q_NULLPTR)
    );
194         selectLabel->setText(QApplication::translate("KbBindWidget", "Click to select keys",
    Q_NULLPTR));
195         resetButton->setText(QApplication::translate("KbBindWidget", "Reset", Q_NULLPTR));
196         label_2->setText(QApplication::translate("KbBindWidget", "Binding", Q_NULLPTR));
197     } // retranslateUi
```

Here is the caller graph for this function:



12.27.2.2 void Ui_KbBindWidget::setupUi (QWidget * *KbBindWidget*) [inline],[inherited]

Definition at line 51 of file ui_kbbindwidget.h.

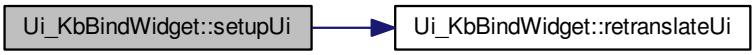
References Ui_KbBindWidget::copyButton, Ui_KbBindWidget::gridLayout, Ui_KbBindWidget::gridLayout_2, Ui_KbBindWidget::horizontalLayout_2, Ui_KbBindWidget::horizontalSpacer, Ui_KbBindWidget::horizontalSpacer_2, Ui_KbBindWidget::keyWidget, Ui_KbBindWidget::label, Ui_KbBindWidget::label_2, Ui_KbBindWidget::line, Ui_KbBindWidget::line_2, Ui_KbBindWidget::rbWidget, Ui_KbBindWidget::resetButton, Ui_KbBindWidget::retranslateUi(), Ui_KbBindWidget::selectLabel, Ui_KbBindWidget::verticalLayout, Ui_KbBindWidget::verticalSpacer, and Ui_KbBindWidget::widget.

Referenced by KbBindWidget::KbBindWidget().

```
52     {
53         if (KbBindWidget->objectName().isEmpty())
54             KbBindWidget->setObjectName(QStringLiteral("KbBindWidget"));
55         KbBindWidget->resize(581, 473);
56         verticalLayout = new QVBoxLayout(KbBindWidget);
57         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
58         verticalLayout->setContentsMargins(0, 0, 0, -1);
59         label = new QLabel(KbBindWidget);
60         label->setObjectName(QStringLiteral("label"));
61         QFont font;
62         font.setBold(true);
63         font.setWeight(75);
64         label->setFont(font);
65 
66         verticalLayout->addWidget(label);
67 
68         line = new QFrame(KbBindWidget);
69         line->setObjectName(QStringLiteral("line"));
70         line->setFrameShape(QFrame::HLine);
71         line->setFrameShadow(QFrame::Sunken);
72 
73         verticalLayout->addWidget(line);
74 
75         gridLayout = new QGridLayout();
76         gridLayout->setObjectName(QStringLiteral("gridLayout"));
77         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
78 
79         gridLayout->addItem(horizontalSpacer_2, 0, 4, 1, 1);
80 
81         keyWidget = new KeyWidget(KbBindWidget);
82         keyWidget->setObjectName(QStringLiteral("keyWidget"));
83         QSizePolicy sizePolicy(QSizePolicy::Preferred, QSizePolicy::Preferred);
84         sizePolicy.setHorizontalStretch(0);
85         sizePolicy.setVerticalStretch(0);
86         sizePolicy.setHeightForWidth(keyWidget->sizePolicy().hasHeightForWidth());
87         keyWidget->setSizePolicy(sizePolicy);
88 
89         gridLayout->addWidget(keyWidget, 0, 1, 1, 3);
90 
91         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
92 
93         gridLayout->addItem(horizontalSpacer, 0, 0, 1, 1);
94 
95         horizontalLayout_2 = new QHBoxLayout();
96         horizontalLayout_2->setSpacing(0);
97         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
98         copyButton = new QPushButton(KbBindWidget);
99         copyButton->setObjectName(QStringLiteral("copyButton"));
100         QSizePolicy sizePolicy1(QSizePolicy::Fixed, QSizePolicy::Fixed);
101         sizePolicy1.setHorizontalStretch(0);
102         sizePolicy1.setVerticalStretch(0);
103         sizePolicy1.setHeightForWidth(copyButton->sizePolicy().hasHeightForWidth());
104         copyButton->setSizePolicy(sizePolicy1);
105         copyButton->setMinimumSize(QSize(140, 0));
106         copyButton->setMaximumSize(QSize(140, 16777215));
107 
108         horizontalLayout_2->addWidget(copyButton);
109 
110         selectLabel = new QLabel(KbBindWidget);
111         selectLabel->setObjectName(QStringLiteral("selectLabel"));
112         QSizePolicy sizePolicy2(QSizePolicy::Expanding, QSizePolicy::Preferred);
113         sizePolicy2.setHorizontalStretch(0);
114         sizePolicy2.setVerticalStretch(0);
115         sizePolicy2.setHeightForWidth(selectLabel->sizePolicy().hasHeightForWidth());
116         selectLabel->setSizePolicy(sizePolicy2);
117         selectLabel->setAlignment(Qt::AlignCenter);
118 
119         horizontalLayout_2->addWidget(selectLabel);
```

```
120
121         resetButton = new QPushButton(KbBindWidget);
122         resetButton->setObjectName(QStringLiteral("resetButton"));
123         QSizePolicy sizePolicy3(QSizePolicy::Expanding, QSizePolicy::Fixed);
124         sizePolicy3.setHorizontalStretch(0);
125         sizePolicy3.setVerticalStretch(0);
126         sizePolicy3.setHeightForWidth(resetButton->sizePolicy().hasHeightForWidth());
127         resetButton->setSizePolicy(sizePolicy3);
128         resetButton->setMinimumSize(QSize(140, 0));
129         resetButton->setMaximumSize(QSize(140, 16777215));
130
131         horizontalLayout_2->addWidget(resetButton);
132
133
134         gridLayout->addLayout(horizontalLayout_2, 1, 1, 1, 3);
135
136
137         verticalLayout->addLayout(gridLayout);
138
139         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
140
141         verticalLayout->addItem(verticalSpacer);
142
143         label_2 = new QLabel(KbBindWidget);
144         label_2->setObjectName(QStringLiteral("label_2"));
145         label_2->setFont(font);
146
147         verticalLayout->addWidget(label_2);
148
149         line_2 = new QFrame(KbBindWidget);
150         line_2->setObjectName(QStringLiteral("line_2"));
151         line_2->setFrameShape(QFrame::HLine);
152         line_2->setFrameShadow(QFrame::Sunken);
153
154         verticalLayout->addWidget(line_2);
155
156         widget = new QWidget(KbBindWidget);
157         widget->setObjectName(QStringLiteral("widget"));
158         QSizePolicy sizePolicy4(QSizePolicy::Preferred, QSizePolicy::Fixed);
159         sizePolicy4.setHorizontalStretch(0);
160         sizePolicy4.setVerticalStretch(0);
161         sizePolicy4.setHeightForWidth(widget->sizePolicy().hasHeightForWidth());
162         widget->setSizePolicy(sizePolicy4);
163         widget->setMinimumSize(QSize(0, 280));
164         widget->setMaximumSize(QSize(16777215, 280));
165         gridLayout_2 = new QGridLayout(widget);
166         gridLayout_2->setSpacing(0);
167         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
168         gridLayout_2->setContentsMargins(0, 0, 0, 0);
169         rbWidget = new RebindWidget(widget);
170         rbWidget->setObjectName(QStringLiteral("rbWidget"));
171         QSizePolicy sizePolicy5(QSizePolicy::Expanding, QSizePolicy::Expanding);
172         sizePolicy5.setHorizontalStretch(0);
173         sizePolicy5.setVerticalStretch(0);
174         sizePolicy5.setHeightForWidth(rbWidget->sizePolicy().hasHeightForWidth());
175         rbWidget->setSizePolicy(sizePolicy5);
176
177         gridLayout_2->addWidget(rbWidget, 0, 0, 1, 1);
178
179
180         verticalLayout->addWidget(widget);
181
182         QWidget::setTabOrder(copyButton, resetButton);
183
184         retranslateUi(KbBindWidget);
185
186         QMetaObject::connectSlotsByName(KbBindWidget);
187     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.27.3 Field Documentation

12.27.3.1 QPushButton* Ui_KbBindWidget::copyButton [inherited]

Definition at line 41 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::retranslateUi(), and Ui_KbBindWidget::setupUi().

12.27.3.2 QGridLayout* Ui_KbBindWidget::gridLayout [inherited]

Definition at line 36 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.3 QGridLayout* Ui_KbBindWidget::gridLayout_2 [inherited]

Definition at line 48 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.4 QHBoxLayout* Ui_KbBindWidget::horizontalLayout_2 [inherited]

Definition at line 40 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.5 QSpacerItem* Ui_KbBindWidget::horizontalSpacer [inherited]

Definition at line 39 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.6 QSpacerItem* Ui_KbBindWidget::horizontalSpacer_2 [inherited]

Definition at line 37 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.7 KeyWidget* Ui_KbBindWidget::keyWidget [inherited]

Definition at line 38 of file ui_kbbindwidget.h.

Referenced by KbBindWidget::KbBindWidget(), KbBindWidget::newLayout(), KbBindWidget::setBind(), Ui_KbBindWidget::setupUi(), and KbBindWidget::updateBind().

12.27.3.8 QLabel* Ui_KbBindWidget::label [inherited]

Definition at line 34 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::retranslateUi(), and Ui_KbBindWidget::setupUi().

12.27.3.9 QLabel* Ui_KbBindWidget::label_2 [inherited]

Definition at line 45 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::retranslateUi(), and Ui_KbBindWidget::setupUi().

12.27.3.10 QFrame* Ui_KbBindWidget::line [inherited]

Definition at line 35 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.11 QFrame* Ui_KbBindWidget::line_2 [inherited]

Definition at line 46 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.12 RebindWidget* Ui_KbBindWidget::rbWidget [inherited]

Definition at line 49 of file ui_kbbindwidget.h.

Referenced by KbBindWidget::newLayout(), KbBindWidget::newSelection(), KbBindWidget::setBind(), Ui_KbBindWidget::setupUi(), and KbBindWidget::updateBind().

12.27.3.13 QPushButton* Ui_KbBindWidget::resetButton [inherited]

Definition at line 43 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::retranslateUi(), and Ui_KbBindWidget::setupUi().

12.27.3.14 QLabel* Ui_KbBindWidget::selectLabel [inherited]

Definition at line 42 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::retranslateUi(), Ui_KbBindWidget::setupUi(), and KbBindWidget::updateSelDisplay().

12.27.3.15 QVBoxLayout* Ui_KbBindWidget::verticalLayout [inherited]

Definition at line 33 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.16 QSpacerItem* Ui_KbBindWidget::verticalSpacer [inherited]

Definition at line 44 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

12.27.3.17 QWidget* Ui_KbBindWidget::widget [inherited]

Definition at line 47 of file ui_kbbindwidget.h.

Referenced by Ui_KbBindWidget::setupUi().

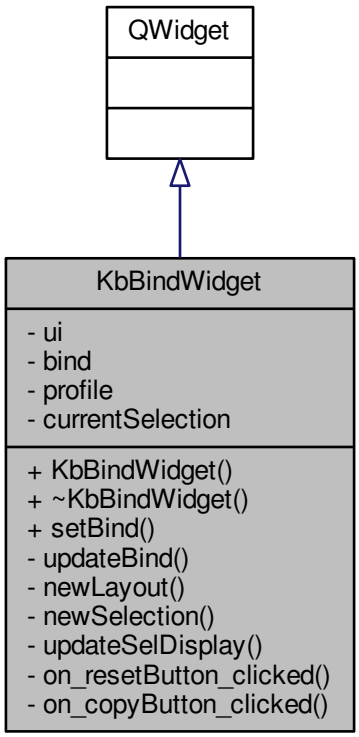
The documentation for this class was generated from the following file:

- src/ckb/ui_kbbindwidget.h

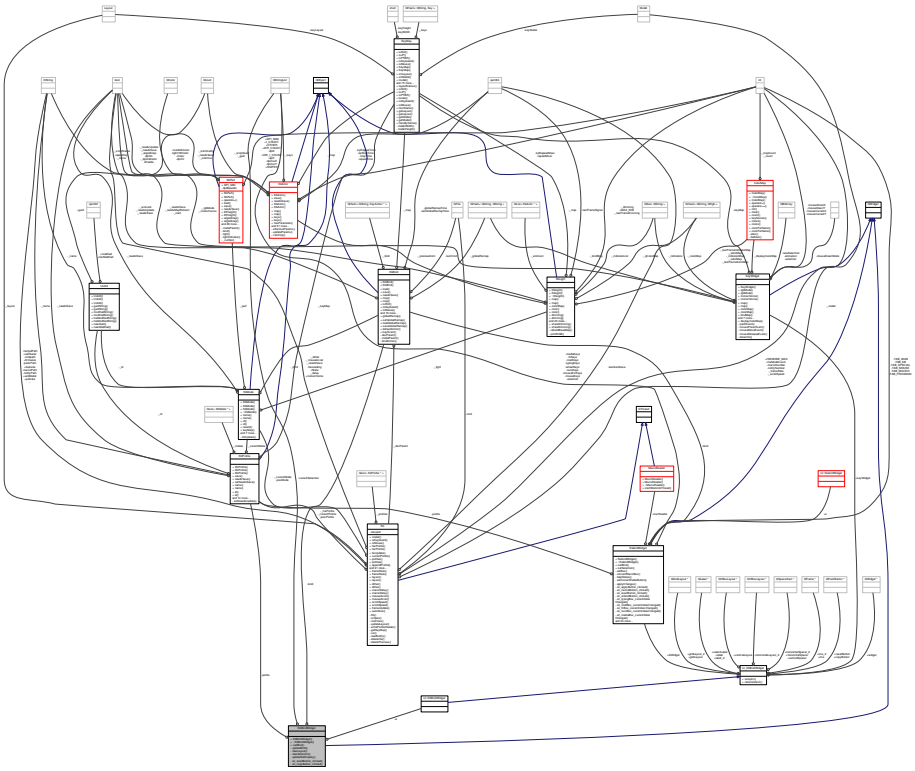
12.28 KbBindWidget Class Reference

`#include <src/ckb/kbbindwidget.h>`

Inheritance diagram for KbBindWidget:



Collaboration diagram for KbBindWidget:



Public Member Functions

- KbBindWidget (QWidget ∗parent=0)
- ∼KbBindWidget ()
- void setBind (KbBind ∗newBind, KbProfile ∗newProfile)

Private Slots

- void updateBind ()
- void newLayout ()
- void newSelection (QStringList selection)
- void updateSelDisplay ()
- void on_resetButton_clicked ()
- void on_copyButton_clicked ()

Private Attributes

- Ui::KbBindWidget ∗ ui
- KbBind ∗ bind
- KbProfile ∗ profile
- QStringList currentSelection

12.28.1 Detailed Description

Definition at line 12 of file kbbindwidget.h.

12.28.2 Constructor & Destructor Documentation

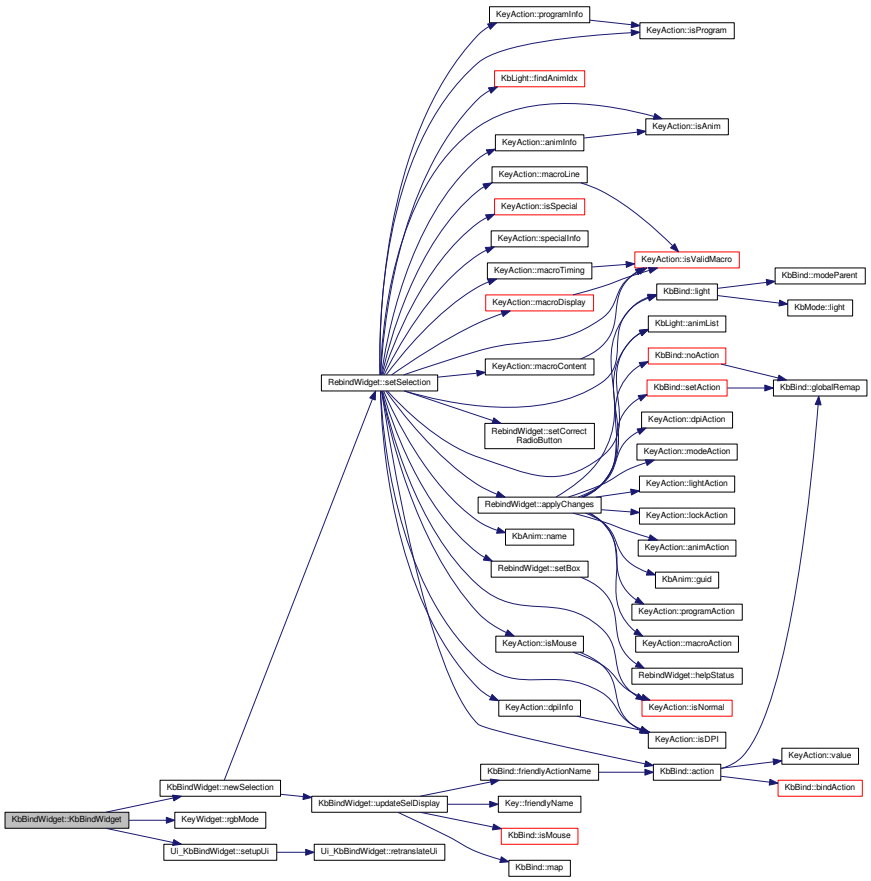
12.28.2.1 KbBindWidget::KbBindWidget (QWidget * *parent* = 0) [explicit]

Definition at line 7 of file kbbindwidget.cpp.

References Ui_KbBindWidget::keyWidget, newSelection(), KeyWidget::rgbMode(), Ui_KbBindWidget::setupUi(), and ui.

```
7                                              :
8     QWidget(parent),
9     ui(new Ui::KbBindWidget), bind(0), profile(0)
10 {
11     ui->setupUi(this);
12     ui->keyWidget->rgbMode(false);
13     connect(ui->keyWidget, SIGNAL(selectionChanged(QStringList)), this, SLOT(
       newSelection(QStringList)));
14 }
```

Here is the call graph for this function:



12.28.2.2 KbBindWidget::~KbBindWidget ( )

Definition at line 16 of file kbbindwidget.cpp.

References ui.

```
16                                  {
17     delete ui;
18 }
```

12.28.3 Member Function Documentation

12.28.3.1 void KbBindWidget::newLayout ( ) [private],[slot]

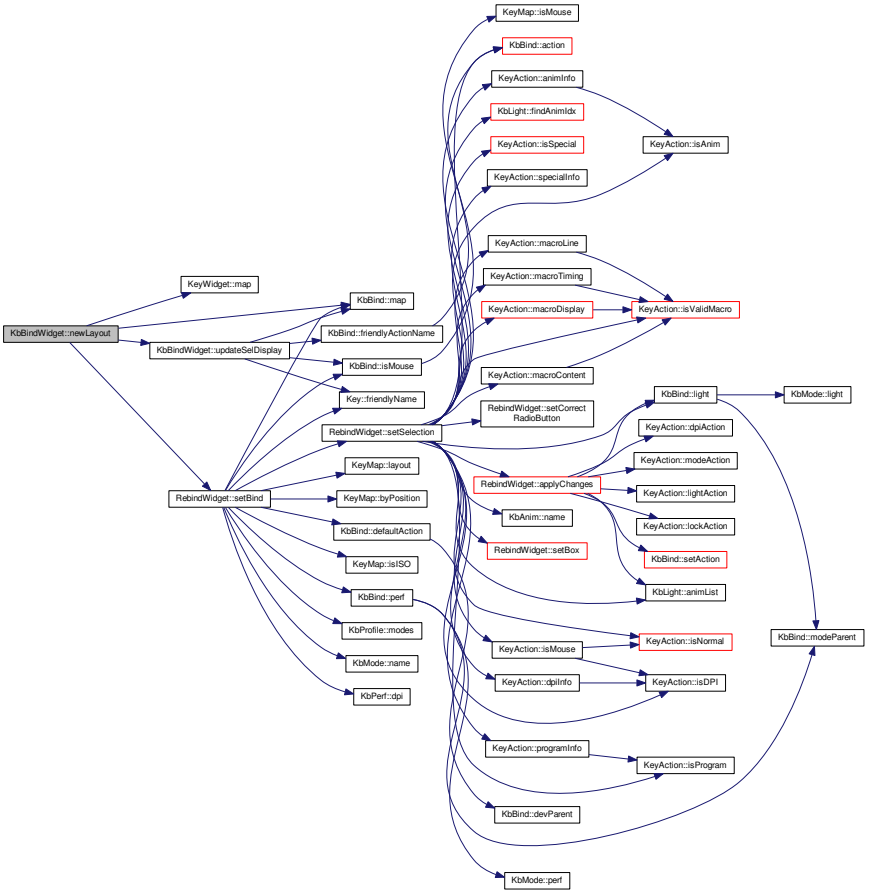
Definition at line 50 of file kbbindwidget.cpp.

References bind, Ui_KbBindWidget::keyWidget, KeyWidget::map(), KbBind::map(), profile, Ui_KbBindWidget::rbWidget, RebindWidget::setBind(), ui, and updateSelDisplay().

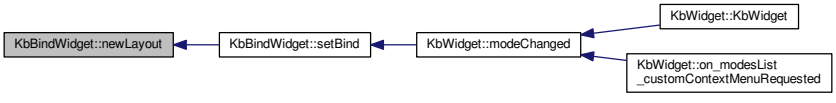
Referenced by setBind().

```
50                              {
51     ui->keyWidget->map(bind->map());
52     ui->rbWidget->setBind(bind, profile);
53     updateSelDisplay();
54 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.28.3.2 void KbBindWidget::newSelection (QStringList *selection*) [private], [slot]

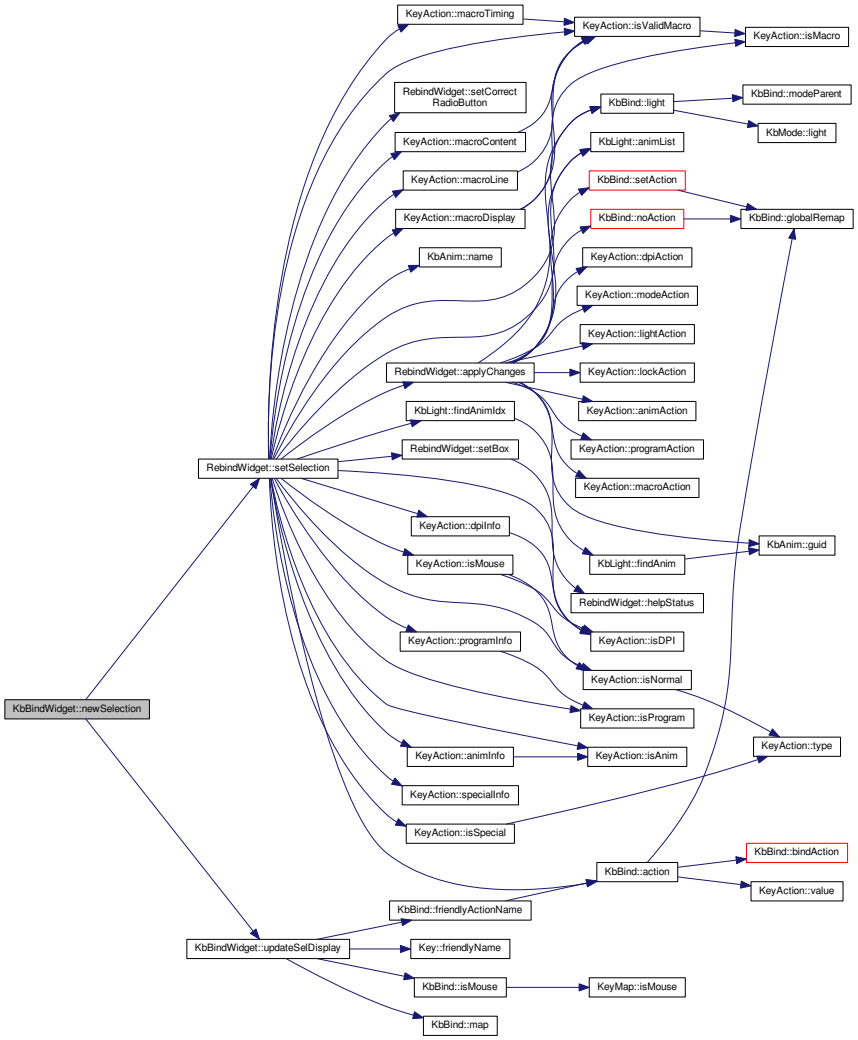
Definition at line 56 of file kbbindwidget.cpp.

References currentSelection, Ui_KbBindWidget::rbWidget, RebindWidget::setSelection(), ui, and updateSelDisplay().

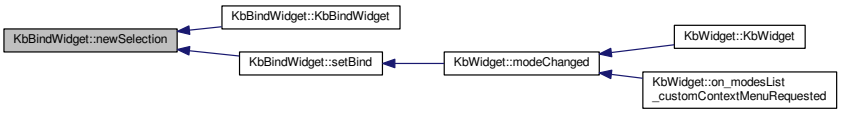
Referenced by KbBindWidget(), and setBind().

```
56                                                  {
57     currentSelection = selection;
58     ui->rbWidget->setSelection(selection, true);
59     updateSelDisplay();
60 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



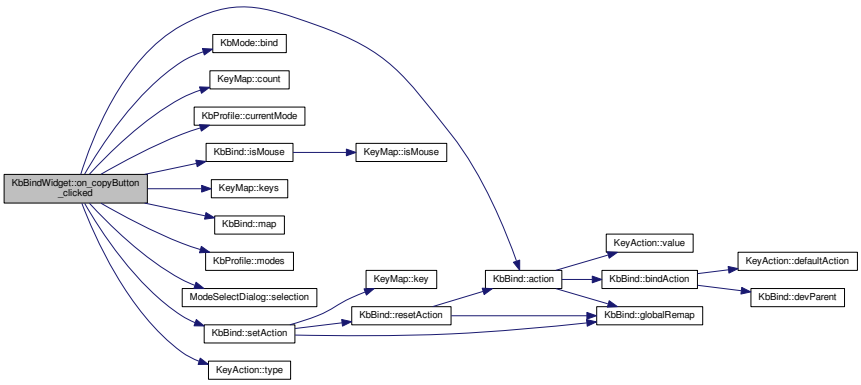
12.28.3.3 void KbBindWidget::on_copyButton_clicked ( ) [private],[slot]

Definition at line 103 of file kbbindwidget.cpp.

References KbBind::action(), bind, KbMode::bind(), KeyMap::count(), KbProfile::currentMode(), currentSelection, KbBind::isMouse(), KeyMap::keys(), KbBind::map(), KbProfile::modes(), profile, ModeSelectDialog::selection(), KbBind::setAction(), and KeyAction::type().

```
103                                         {
104     QStringList selection = currentSelection;
105     const KeyMap& map = bind->map();
106     int count = selection.count();
107     QString type = bind->isMouse() ? "button" : "key";
108     QString text = tr("%1 %2").arg(count).arg(type) + (count == 1 ? "" : "s");
109     if(count == 0){
110         // Copy all keys if none selected
111         selection = map.keys();
112         text = tr("all %1s").arg(type);
113     }
114     text = "Copy binding for " + text + " to:";
115     // Display popup
116     ModeSelectDialog dialog(this, profile->currentMode(),
      profile->modes(), text);
117     if(dialog.exec() != QDialog::Accepted)
118         return;
119     // Copy selected keys to selected modes
120     QList<KbMode*> selectedModes = dialog.selection();
121     foreach(KbMode* mode, selectedModes){
122         KbBind* modeBind = mode->bind();
123         foreach(const QString& key, selection){
124             modeBind->setAction(key, bind->action(key));
125         }
126     }
127 }
```

Here is the call graph for this function:



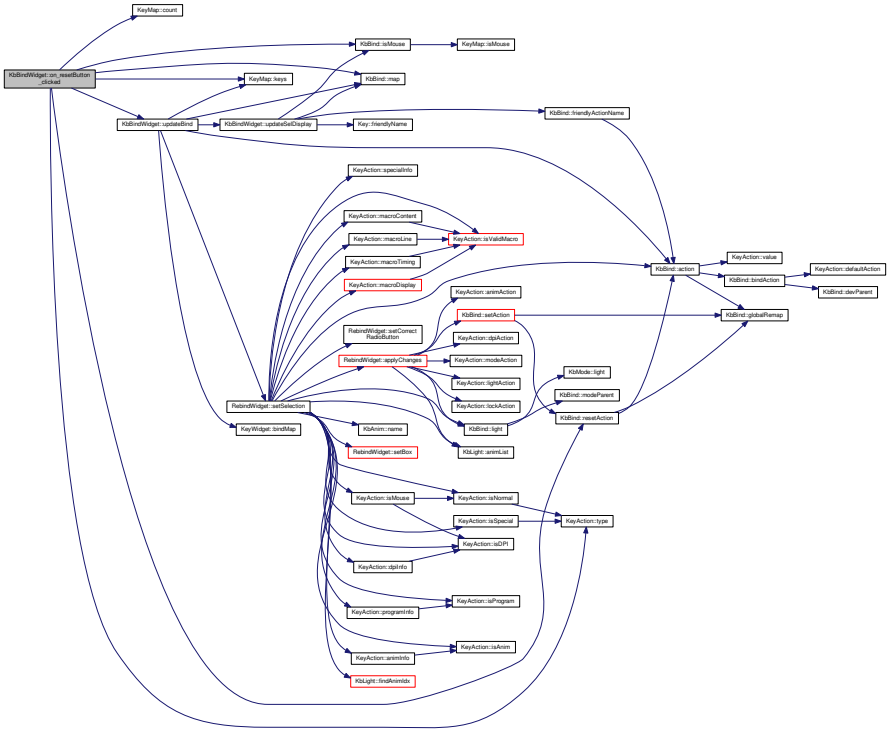
12.28.3.4 void KbBindWidget::on_resetButton_clicked ( ) [private], [slot]

Definition at line 82 of file kbbindwidget.cpp.

References bind, KeyMap::count(), currentSelection, KbBind::isMouse(), KeyMap::keys(), KbBind::map(), KbBind::resetAction(), KeyAction::type(), and updateBind().

```
82                                          {
83     QStringList selection = currentSelection;
84     const KeyMap& map = bind->map();
85     if(selection.isEmpty())
86         // Reset all keys if none selected
87         selection = map.keys();
88     uint count = selection.count();
89     QString type = bind->isMouse() ? "button" : "key";
90     QString text;
91     if(count == map.count())
92         text = "<center>Reset all %1s to default?</center>";
93     else if(count == 1)
94         text = "<center>Reset this %1 to default?</center>";
95     else
96         text = tr("<center>Reset %1 %2s to default?</center>").arg(count);
97     if(QMessageBox(QMessageBox::NoIcon, "Confirm action", text.arg(type), QMessageBox::Yes |
      QMessageBox::No, this).exec() != QMessageBox::Yes)
98         return;
99     bind->resetAction(selection);
100     updateBind();
101 }
```

Here is the call graph for this function:



12.28.3.5 void KbBindWidget::setBind (KbBind * newBind, KbProfile * newProfile)

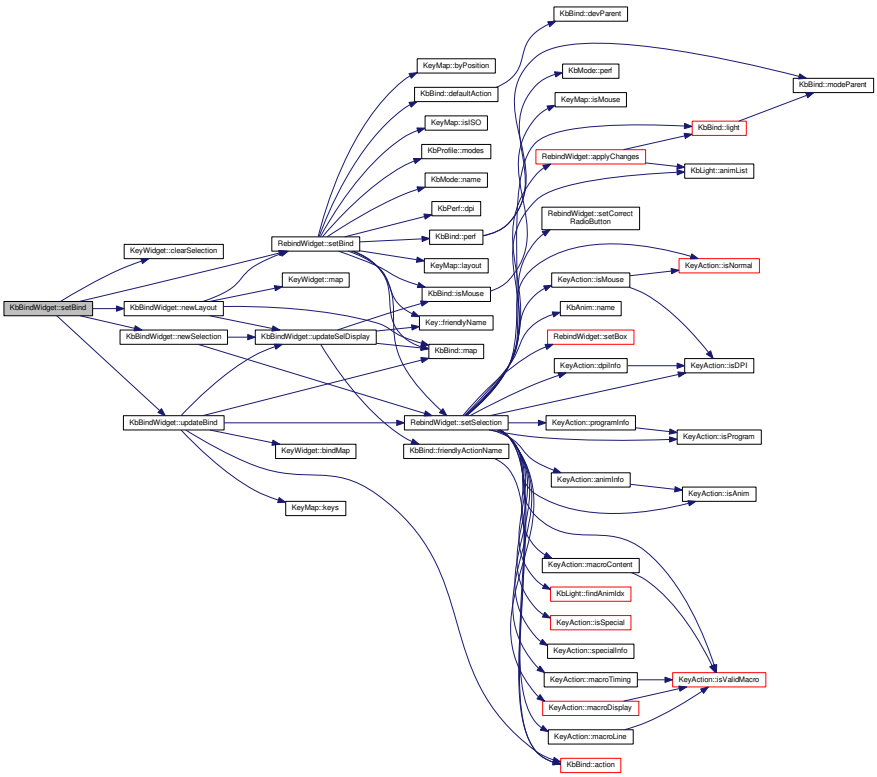
Definition at line 20 of file kbbindwidget.cpp.

References bind, KeyWidget::clearSelection(), Ui_KbBindWidget::keyWidget, newLayout(), newSelection(), profile, Ui_KbBindWidget::rbWidget, RebindWidget::setBind(), ui, and updateBind().

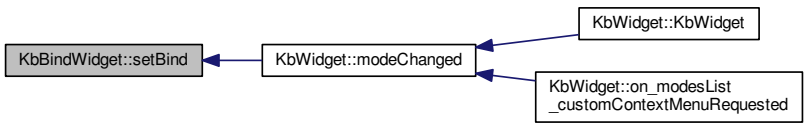
Referenced by KbWidget::modeChanged().

```
20                                                                  {
21     ui->keyWidget->clearSelection();
22     ui->rbWidget->setBind(newBind, newProfile);
23     newSelection(QStringList());
24     if(bind == newBind)
25         return;
26     if(bind){
27         disconnect(bind, SIGNAL(layoutChanged()), this, SLOT(newLayout()));
28         disconnect(bind, SIGNAL(updated()), this, SLOT(updateBind()));
29     }
30     connect(newBind, SIGNAL(layoutChanged()), this, SLOT(newLayout()));
31     connect(newBind, SIGNAL(updated()), this, SLOT(updateBind()));
32     bind = newBind;
33     profile = newProfile;
34     newLayout();
35     ui->rbWidget->setBind(bind, profile);
36     updateBind();
37 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.28.3.6 void KbBindWidget::updateBind ( ) [private],[slot]

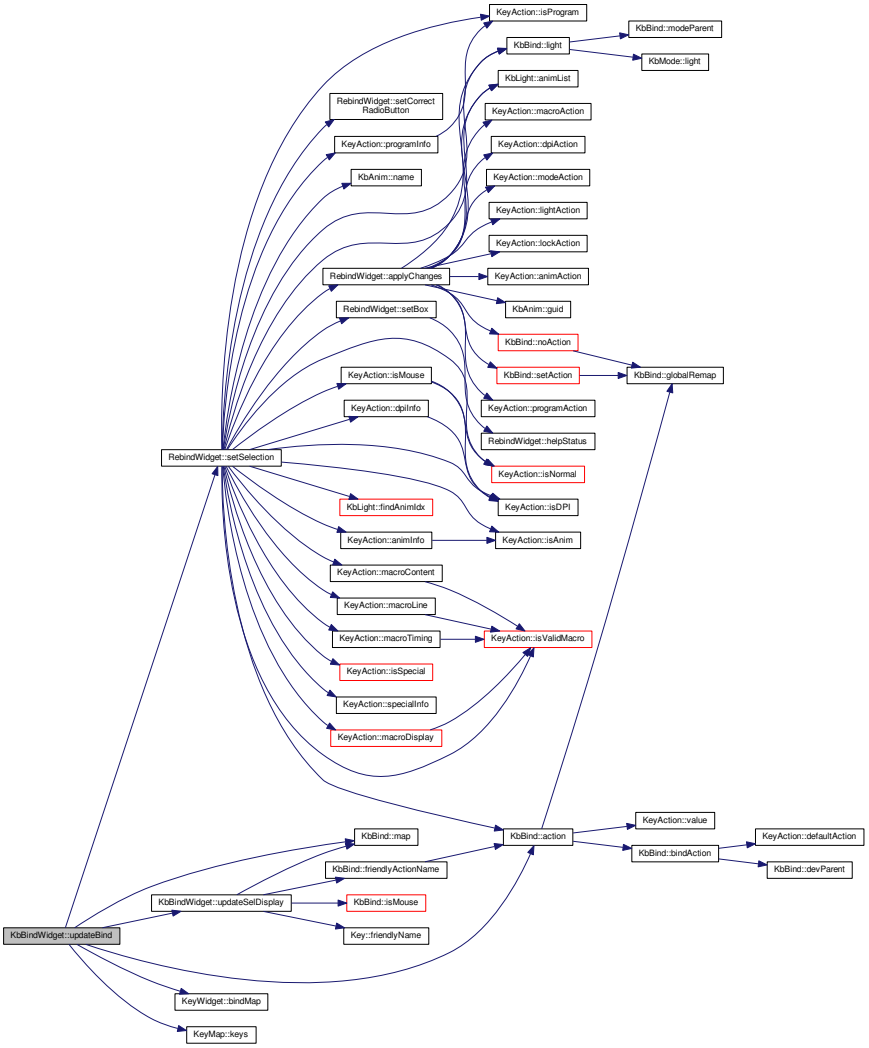
Definition at line 39 of file kbbindwidget.cpp.

References KbBind::action(), bind, KeyWidget::bindMap(), currentSelection, KeyMap::keys(), Ui_KbBindWidget::keyWidget, KbBind::map(), Ui_KbBindWidget::rbWidget, RebindWidget::setSelection(), ui, and updateSelDisplay().

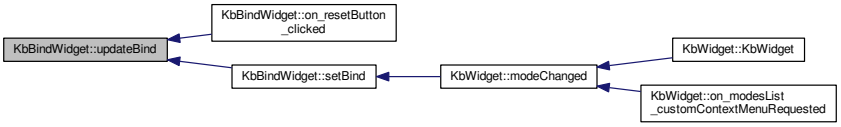
Referenced by on_resetButton_clicked(), and setBind().

```
39                                {
40     const KeyMap& map = bind->map();
41     // Build the action list based on the keymap
42     QHash<QString, QString> actions;
43     foreach(const QString& key, map.keys())
44         actions[key] = bind->action(key);
45     ui->keyWidget->bindMap(actions);
46     ui->rbWidget->setSelection(currentSelection);
47     updateSelDisplay();
48 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.28.3.7 void KbBindWidget::updateSelDisplay ( ) [private], [slot]

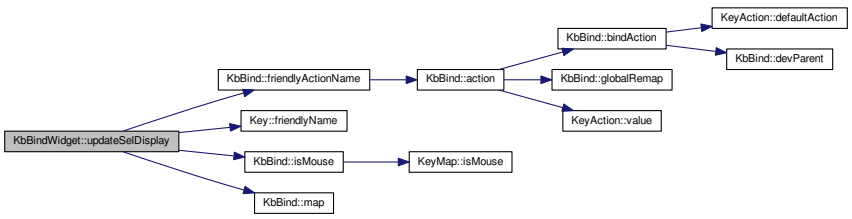
Definition at line 62 of file kbbindwidget.cpp.

References bind, currentSelection, KbBind::friendlyActionName(), Key::friendlyName(), KbBind::isMouse(), KbBind::map(), Ui_KbBindWidget::selectLabel, and ui.

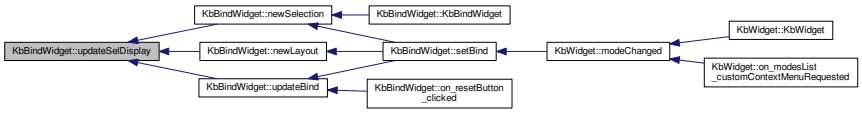
Referenced by newLayout(), newSelection(), and updateBind().

```
62                                      {
63     int count = currentSelection.count();
64     if(count == 0){
65         // No keys selected
66         ui->selectLabel->setText("Click to select");
67         return;
68     }
69     if(count == 1){
70         // Single key selected: show key name and binding
71         QString key = currentSelection[0];
72         const Key& pos = bind->map()[key];
73         if(!pos)
74             ui->selectLabel->setText("(Unknown)");
75         else
76             ui->selectLabel->setText(pos.friendlyName(false).split("\n")[0] + " →
      " + bind->friendlyActionName(key).split("\n")[0]);
77         return;
78     }
79     ui->selectLabel->setText(QString("%1 %2 selected").arg(count).arg(
      bind->isMouse() ? "buttons" : "keys"));
80 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.28.4 Field Documentation

12.28.4.1 KbBind* KbBindWidget::bind [private]

Definition at line 34 of file kbbindwidget.h.

Referenced by newLayout(), on_copyButton_clicked(), on_resetButton_clicked(), setBind(), updateBind(), and updateSelDisplay().

12.28.4.2 QStringList KbBindWidget::currentSelection [private]

Definition at line 36 of file kbbindwidget.h.

Referenced by newSelection(), on_copyButton_clicked(), on_resetButton_clicked(), updateBind(), and updateSelDisplay().

12.28.4.3 KbProfile* KbBindWidget::profile [private]

Definition at line 35 of file kbbindwidget.h.

Referenced by newLayout(), on_copyButton_clicked(), and setBind().

12.28.4.4 Ui::KbBindWidget* KbBindWidget::ui [private]

Definition at line 32 of file kbbindwidget.h.

Referenced by KbBindWidget(), newLayout(), newSelection(), setBind(), updateBind(), updateSelDisplay(), and ~KbBindWidget().

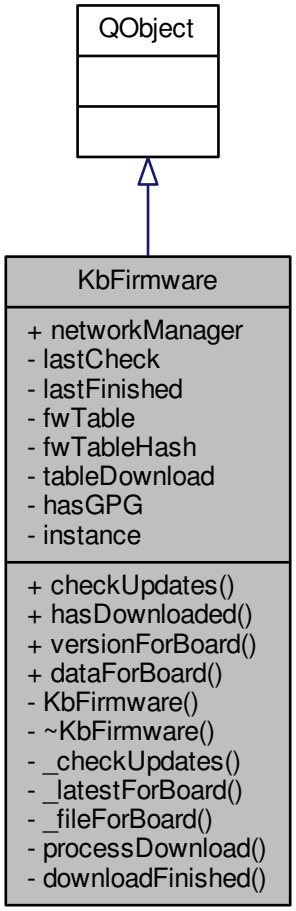
The documentation for this class was generated from the following files:

- src/ckb/kbbindwidget.h
- src/ckb/kbbindwidget.cpp

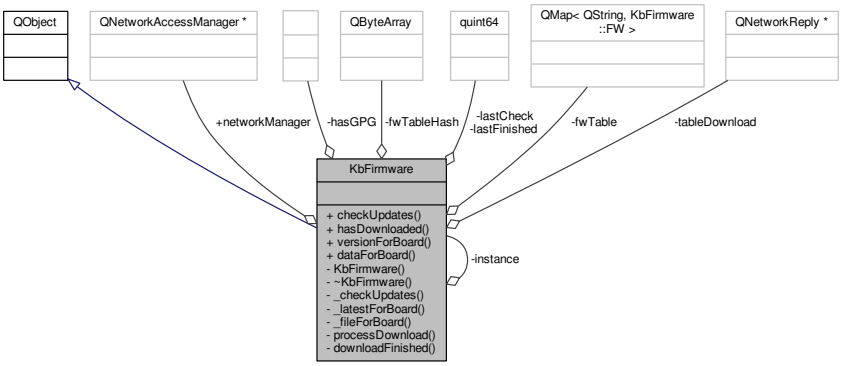
12.29 KbFirmware Class Reference

`#include <src/ckb/kbfirmware.h>`

Inheritance diagram for KbFirmware:



Collaboration diagram for KbFirmware:



Data Structures

- struct FW

Signals

- void downloaded ()

Static Public Member Functions

- static bool checkUpdates ()
- static bool hasDownloaded ()
- static float versionForBoard (const QString &features, bool waitForComplete=false)
- static QByteArray dataForBoard (const QString &features)

Data Fields

- QNetworkAccessManager * networkManager

Private Types

- enum { UNKNOWN = -1, NO, YES }

Private Slots

- void processDownload (QNetworkReply *reply)
- void downloadFinished ()

Private Member Functions

- KbFirmware ()
- ~KbFirmware ()
- bool _checkUpdates ()
- float _latestForBoard (const QString &features, bool waitForComplete)
- QByteArray _fileForBoard (const QString &features)

Private Attributes

- quint64 lastCheck
- quint64 lastFinished
- QMap< QString, FW > fwTable
- QByteArray fwTableHash
- QNetworkReply * tableDownload
- enum KbFirmware:: { ... } hasGPG

Static Private Attributes

- static KbFirmware instance

12.29.1 Detailed Description

Definition at line 10 of file kbfirmware.h.

12.29.2 Member Enumeration Documentation

12.29.2.1 anonymous enum [private]

Enumerator

UNKNOWN

NO

YES

Definition at line 52 of file kbfirmware.h.

```
52 { UNKNOWN = -1, NO, YES } hasGPG :2;
```

12.29.3 Constructor & Destructor Documentation

12.29.3.1 KbFirmware::KbFirmware ( ) [private]

Definition at line 14 of file kbfirmware.cpp.

References networkManager.

```
14                          :
15     lastCheck(0), lastFinished(0), networkManager(0),
       tableDownload(0), hasGPG(UNKNOWN)
16 {
17     // Disable bearer polling. This corrects an issue with latency spikes when
18     // using WiFi. The problem and workaround are described here:
19     // https://lostdomain.org/2017/06/17/qt-qnetworkaccessmanager-causing-latency-spikes-on-wifi/
20     qputenv("QT_BEARER_POLL_TIMEOUT", QByteArray::number(-1));
21     networkManager = new QNetworkAccessManager();
22 }
```

12.29.3.2 KbFirmware::~KbFirmware ( ) [private]

Definition at line 24 of file kbfirmware.cpp.

References networkManager.

```
24                          {
25     delete networkManager;
26 }
```

12.29.4 Member Function Documentation

12.29.4.1 bool KbFirmware::_checkUpdates ( ) [private]

Definition at line 28 of file kbfirmware.cpp.

References AUTO_CHECK_TIME, downloadFinished(), lastCheck, networkManager, and tableDownload.

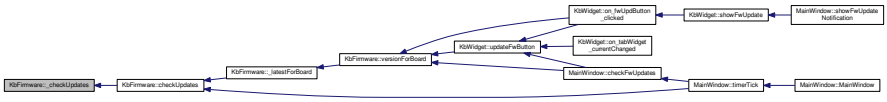
Referenced by checkUpdates().

```
28                                  {
29     quint64 now = QDateTime::currentMSecsSinceEpoch();
30     if(now < lastCheck + AUTO_CHECK_TIME)
31         return false;
```

```
32     // First location is for debugging only.
33     // tableDownload =
       networkManager->get(QNetworkRequest(QUrl("https://raw.githubusercontent.com/frickler24/ckb-next/issues-26-Firmware-Inc
34     // This one is the production one.
35     tableDownload = networkManager->get(QNetworkRequest(QUrl("
      https://raw.githubusercontent.com/mattanger/ckb-next/master/FIRMWARE")));
36     connect(tableDownload, SIGNAL(finished()), this, SLOT(
      downloadFinished()));
37     lastCheck = now;
38     return true;
39 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.29.4.2 QByteArray KbFirmware::_fileForBoard (const QString & *features*) [private]

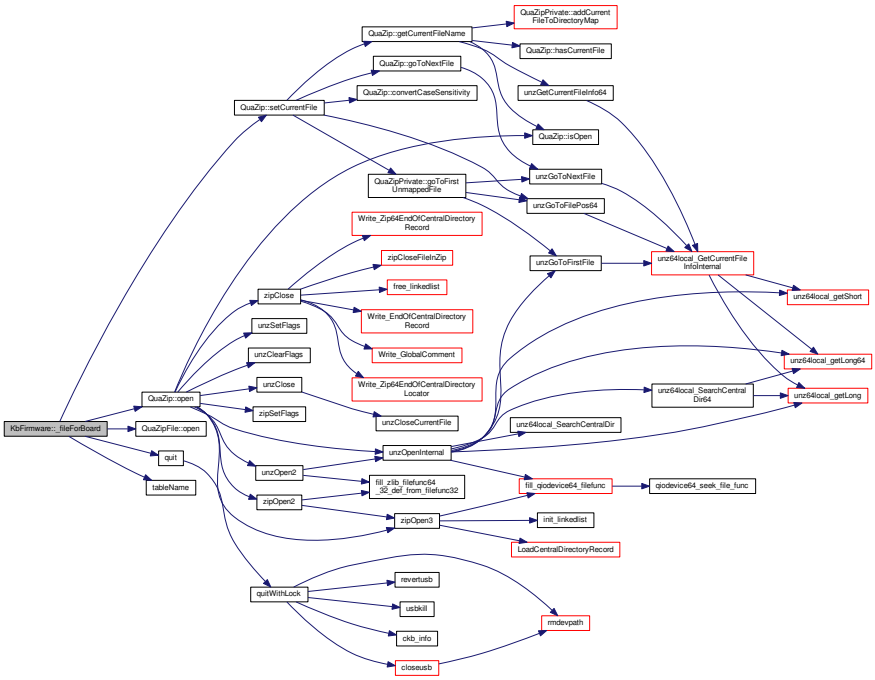
Definition at line 180 of file kbfirmware.cpp.

References QuaZip::csInsensitive, KbFirmware::FW::fileName, fwTable, KbFirmware::FW::hash, QuaZip::mdUnzip, networkManager, QuaZip::open(), QuaZipFile::open(), quit(), QuaZip::setCurrentFile(), tableName(), and KbFirmware::FW::url.

Referenced by dataForBoard().

```
180                                                     {
181     QString name = tableName(features);
182     FW info = fwTable.value(name);
183     if(info.hash.isEmpty())
184         return "";
185     // Download zip from URL. Wait for it to finish.
186     QNetworkReply* reply = networkManager->get(QNetworkRequest(QUrl(info.url)));
187     QEventLoop loop(this);
188     connect(reply, SIGNAL(finished()), &loop, SLOT(quit()));
189     loop.exec();
190     // Download finished, process data
191     if(reply->error() != QNetworkReply::NoError)
192         return "";
193     QByteArray zipData = reply->readAll();
194     QBuffer buffer(&zipData);
195     // Open zip archive
196     QuaZip zip(&buffer);
197     if(!zip.open(QuaZip::mdUnzip))
198         return "";
199     // Find the desired file
200     if(!zip.setCurrentFile(info.fileName, QuaZip::csInsensitive))
201         return "";
202     QuaZipFile binFile(&zip);
203     if(!binFile.open(QIODevice::ReadOnly))
204         return "";
205     QByteArray binary = binFile.readAll();
206     // Check the hash
207     if(QCryptographicHash::hash(binary, QCryptographicHash::Sha256) != info.hash)
208         return "";
209     return binary;
210 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.29.4.3 float KbFirmware::_latestForBoard (const QString & *features*, bool *waitForComplete*) [private]

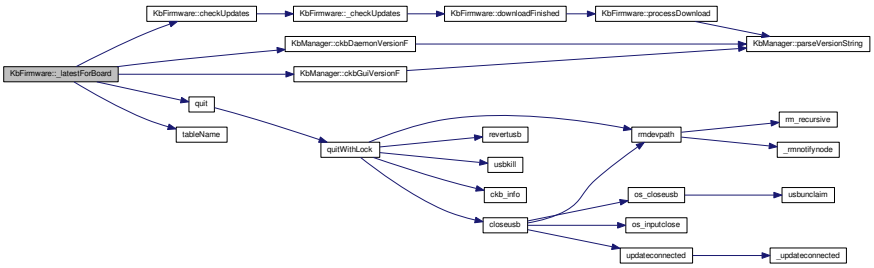
Definition at line 162 of file kbfirmware.cpp.

References checkUpdates(), KbManager::ckbDaemonVersionF(), KbManager::ckbGuiVersionF(), KbFirmware::FW::ckbVersion, downloaded(), fwTable, KbFirmware::FW::fwVersion, KbFirmware::FW::hash, quit(), tableDownload, and tableName().

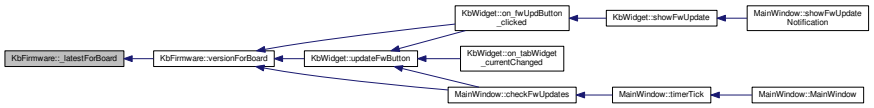
Referenced by versionForBoard().

```
162                                                                                 {
163     if((tableDownload || checkUpdates()) && waitForComplete){
164         // If waiting is desired, enter an event loop and stay here until the download is finished
165         QEventLoop loop(this);
166         connect(this, SIGNAL(downloaded()), &loop, SLOT(quit()));
167         loop.exec();
168     }
169     // Find this board
170     QString name = tableName(features);
171     FW info = fwTable.value(name);
172     if(info.hash.isEmpty())
173         return 0.f;
174     // Don't return the new version if the current ckb doesn't support it
175     if(info.ckbVersion > KbManager::ckbGuiVersionF() || info.ckbVersion >
    KbManager::ckbDaemonVersionF())
176         return -1.f;
177     return info.fwVersion;
178 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.29.4.4 static bool KbFirmware::checkUpdates ( ) [inline], [static]

Definition at line 15 of file kbfirmware.h.

References _checkUpdates(), and instance.

Referenced by _latestForBoard(), and MainWindow::timerTick().

```
15 { return instance._checkUpdates(); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.29.4.5 static QByteArray KbFirmware::dataForBoard (const QString & *features*) [inline], [static]

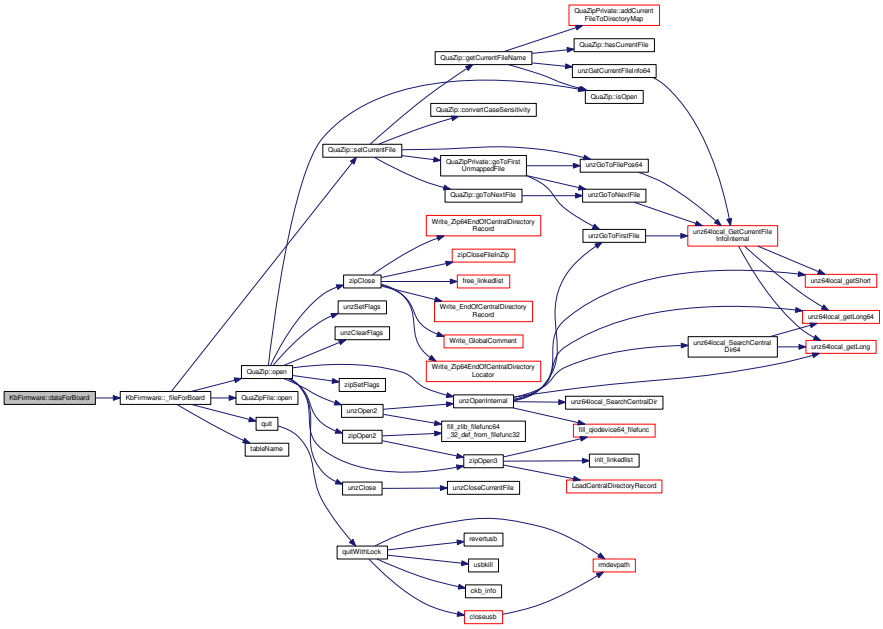
Definition at line 25 of file kbfirmware.h.

References _fileForBoard(), and instance.

Referenced by FwUpgradeDialog::exec().

```
25 { return instance._fileForBoard(features); }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.29.4.6 void KbFirmware::downloaded ( ) [signal]

Definition at line 148 of file moc_kbfirmware.cpp.

Referenced by _latestForBoard(), and downloadFinished().

```
149 {
150     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
151 }
```

Here is the caller graph for this function:

12.29.4.7 void KbFirmware::downloadFinished ( ) [private], [slot]

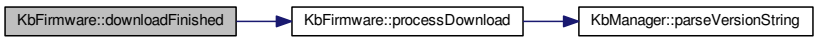
Definition at line 141 of file kbfirmware.cpp.

References downloaded(), processDownload(), and tableDownload.

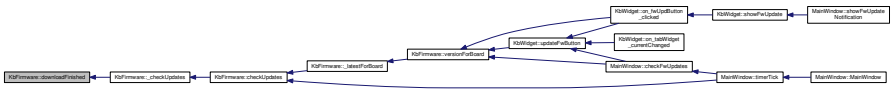
Referenced by _checkUpdates().

```
141                                 {
142     if(!tableDownload)
143         return;
144     processDownload(tableDownload);
145     tableDownload->deleteLater();
146     tableDownload = 0;
147     emit downloaded();
148 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.29.4.8 static bool KbFirmware::hasDownloaded ( ) [inline], [static]

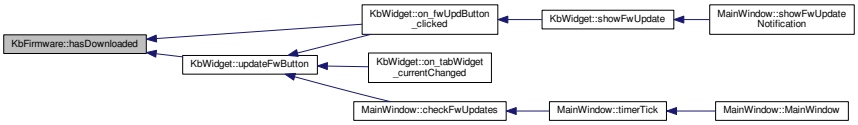
Definition at line 18 of file kbfirmware.h.

References instance, and lastFinished.

Referenced by KbWidget::on_fwUpdButton_clicked(), and KbWidget::updateFwButton().

```
18 { return instance.lastFinished != 0; }
```

Here is the caller graph for this function:



12.29.4.9 void KbFirmware::processDownload (QNetworkReply * *reply*) [private], [slot]

Definition at line 41 of file kbfirmware.cpp.

References KbFirmware::FW::ckbVersion, KbFirmware::FW::fileName, fwTable, fwTableHash, KbFirmware::FW::fwVersion, hasGPG, KbFirmware::FW::hash, lastCheck, lastFinished, NO, KbManager::parseVersionString(), UNKNOWN, KbFirmware::FW::url, and YES.

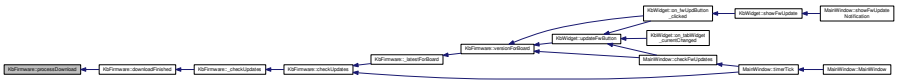
Referenced by downloadFinished().

```
41                                                  {
42      if(reply->error() != QNetworkReply::NoError)
43          return;
44      // Update last check
45      lastCheck = lastFinished = QDateTime::currentMSecsSinceEpoch();
46      QByteArray data = reply->readAll();
47      // Don't do anything if this is the same as the last version downloaded
48      QByteArray hash = QCryptographicHash::hash(data, QCryptographicHash::Sha256);
49      if(hash == fwTableHash)
50          return;
51      fwTableHash = hash;
52      if(hasGPG == UNKNOWN){
53          // Check for a GPG installation
54          QProcess gpg;
55          gpg.start("gpg", QStringList("--version"));
56          gpg.waitForFinished();
57          if(gpg.error() == QProcess::FailedToStart)
58              // No GPG install
59              hasGPG = NO;
60          else {
61              QString output = QString::fromUtf8(gpg.readAll());
62              // Must support RSA keys and SHA256
63              if(output.contains("RSA", Qt::CaseInsensitive) && output.contains("SHA256", Qt::CaseInsensitive
    ))
64                  hasGPG = YES;
65              else
66                  hasGPG = NO;
67          }
68          if(!hasGPG)
69              qDebug() << "No GPG detected, signature verification disabled";
70      }
71      if(hasGPG){
72          // If GPG is available, check the signature on the file before proceeding.
73          QDir tmp = QDir::temp();
74          // Save file to a temporary path. Include PID to avoid conflicts
75          qint64 pid = QCoreApplication::applicationPid();
76          QString fwPath = tmp.absoluteFilePath(QString("ckb-%1-firmware").arg(pid));
77          QFile firmware(fwPath);
78          if(!firmware.open(QIODevice::WriteOnly)
79                  || firmware.write(data) != data.length()){
80              qDebug() << "Failed to write firmware file to temporary location, aborting firmware check";
81              return;
82          }
83          firmware.close();
84          // Write GPG key
85          QString keyPath = tmp.absoluteFilePath(QString("ckb-%1-key.gpg").arg(pid));
86          if(!QFile::copy(":/bin/ckb-next-key.gpg", keyPath)){
87              firmware.remove();
88              qDebug() << "Failed to write GPG key to temporary location, aborting firmware check";
89              return;
90          }
91          // Check signature
92          QProcess gpg;
93          gpg.start("gpg", QStringList("--no-default-keyring") << "--keyring" << keyPath << "--verify" <<
    fwPath);
94          gpg.waitForFinished();
95          // Clean up temp files
96          tmp.remove(fwPath);
97          tmp.remove(keyPath);
98          if(gpg.error() != QProcess::UnknownError || gpg.exitCode() != 0){
99              qDebug() << "GPG couldn't verify firmware signature:";
100              qDebug() << gpg.readAllStandardOutput();
101              qDebug() << gpg.readAllStandardError();
102              return;
103          }
104          // Signature good, proceed to update database
105      }
106      fwTable.clear();
107      QStringList lines = QString::fromUtf8(data).split("\n");
108      bool scan = false;
109      foreach(QString line, lines){
110          // Collapse whitespace
111          line.replace(QRegExp("\\s+"), " ").remove(QRegExp("^\\s")).remove(QRegExp("\\s$"));
112          // Skip empty or commented-out lines
113          if(line.length() == 0 || line.at(0) == '#')
114              continue;
115          // Don't read anything until the entries begin and don't read anything after they end
116          if(!scan){
117              if(line == "!BEGIN FW ENTRIES")
118                  scan = true;
119              else
120                  continue;
121          }
122          if(line == "!END FW ENTRIES")
123              break;
124          QStringList components = line.split(" ");
125          if(components.length() != 7)
```

```
126             continue;
127         // "VENDOR-PRODUCT"
128         QString device = components[0].toUpper() + "-" + components[1].toUpper();
129         FW fw;
130         fw.fwVersion = components[2].toFloat();                              // Firmware blob version
131         fw.url = QUrl::fromPercentEncoding(components[3].toLatin1());        // URL to zip file
132         fw.ckbVersion = KbManager::parseVersionString(components[4]);        //
       Minimum ckb version
133         fw.fileName = QUrl::fromPercentEncoding(components[5].toLatin1());   // Name of file inside zip
134         fw.hash = QByteArray::fromHex(components[6].toLatin1());             // SHA256 of file inside zip
135         // Update entry
136         fwTable[device] = fw;
137     }
138     qDebug() << "Downloaded new firmware list." << fwTable.count() << "entries found.";
139 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.29.4.10 static float KbFirmware::versionForBoard (const QString & *features*, bool *waitForComplete* = `false`) [inline], [static]

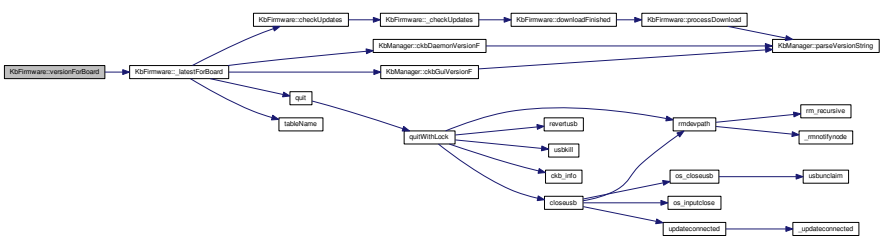
Definition at line 22 of file kbfirmware.h.

References _latestForBoard(), and instance.

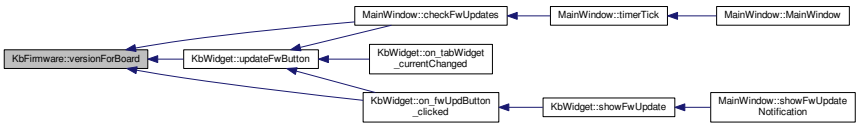
Referenced by MainWindow::checkFwUpdates(), KbWidget::on_fwUpdButton_clicked(), and KbWidget::updateFwButton().

```
22 { return instance._latestForBoard(features, waitForComplete); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.29.5 Field Documentation

12.29.5.1 QMap<QString, FW> KbFirmware::fwTable [private]

Definition at line 44 of file kbfirmware.h.

Referenced by _fileForBoard(), _latestForBoard(), and processDownload().

12.29.5.2 QByteArray KbFirmware::fwTableHash [private]

Definition at line 46 of file kbfirmware.h.

Referenced by processDownload().

12.29.5.3 enum { ... } KbFirmware::hasGPG [private]

Referenced by processDownload().

12.29.5.4 KbFirmware KbFirmware::instance [static], [private]

Definition at line 58 of file kbfirmware.h.

Referenced by checkUpdates(), dataForBoard(), hasDownloaded(), and versionForBoard().

12.29.5.5 quint64 KbFirmware::lastCheck [private]

Definition at line 35 of file kbfirmware.h.

Referenced by _checkUpdates(), and processDownload().

12.29.5.6 quint64 KbFirmware::lastFinished [private]

Definition at line 35 of file kbfirmware.h.

Referenced by hasDownloaded(), and processDownload().

12.29.5.7 QNetworkAccessManager* KbFirmware::networkManager

Definition at line 28 of file kbfirmware.h.

Referenced by _checkUpdates(), _fileForBoard(), KbFirmware(), and ~KbFirmware().

12.29.5.8 QNetworkReply* KbFirmware::tableDownload [private]

Definition at line 49 of file kbfirmware.h.

Referenced by _checkUpdates(), _latestForBoard(), and downloadFinished().

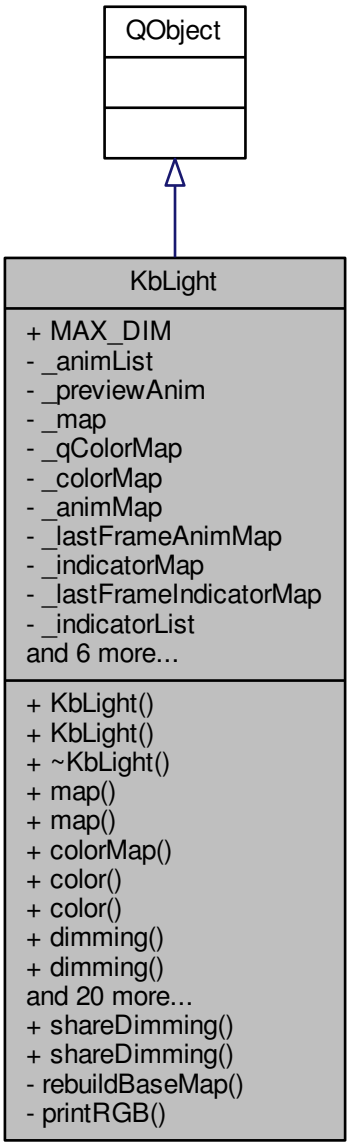
The documentation for this class was generated from the following files:

- src/ckb/kbfirmware.h
- src/ckb/kbfirmware.cpp
- src/ckb/release/moc_kbfirmware.cpp

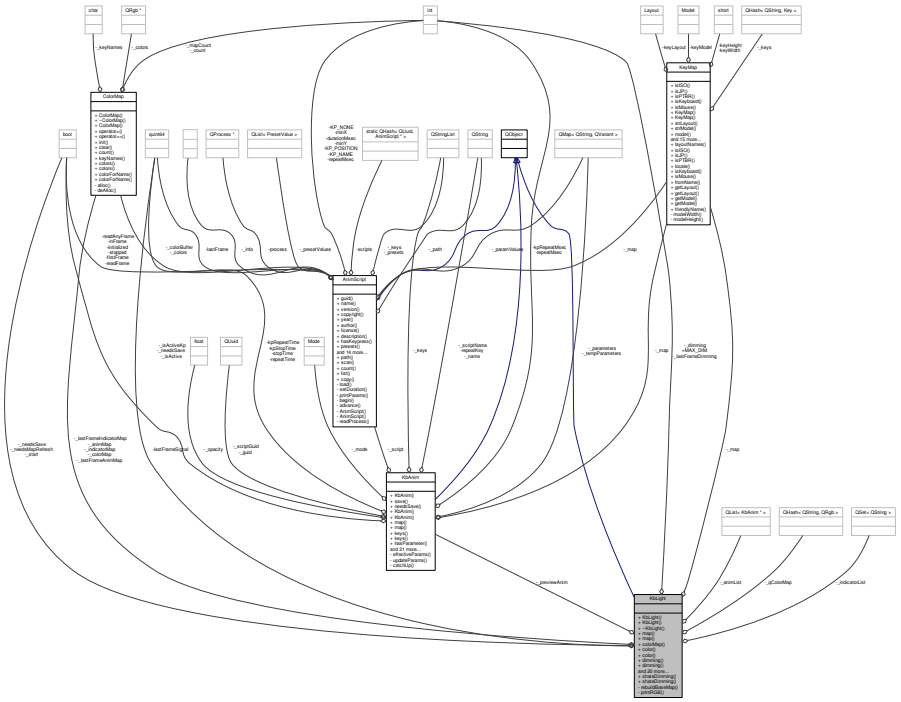
12.30 KbLight Class Reference

`#include <src/ckb/kblight.h>`

Inheritance diagram for KbLight:



Collaboration diagram for KbLight:



Public Types

- typedef QList< KbAnim * > AnimList

Signals

- void didLoad ()
- void updated ()
- void frameDisplayed (const ColorMap &animatedColors, const QSet< QString > &indicatorList)

Public Member Functions

- KbLight (KbMode *parent, const KeyMap &keyMap)
- KbLight (KbMode *parent, const KeyMap &keyMap, const KbLight &other)
- ~KbLight ()
- const KeyMap & map ()
- void map (const KeyMap &map)
- const QColorMap & colorMap ()
- void color (const QString &key, const QColor &newColor)
- void color (const QColor &newColor)
- int dimming ()
- void dimming (int newDimming)
- KbAnim * addAnim (const AnimScript *base, const QStringList &keys, const QString &name, const QMap< QString, QVariant > &preset)
- KbAnim * duplicateAnim (KbAnim *oldAnim)
- const AnimList & animList ()
- void animList (const AnimList &newAnimList)
- KbAnim * findAnim (const QUuid &guid) const

- int findAnimIdx (const QUuid &guid) const
- void previewAnim (const AnimScript *base, const QStringList &keys, const QMap< QString, QVariant > &preset)
- void stopPreview ()
- void restartAnimation ()
- void animKeypress (const QString &key, bool down)
- void open ()
- bool isStarted ()
- void close ()
- void resetIndicators ()
- void setIndicator (const char *name, QRgb argb)
- void frameUpdate (QFile &cmd, bool monochrome=false)
- void base (QFile &cmd, bool ignoreDim=false, bool monochrome=false)
- void load (CkbSettings &settings)
- void save (CkbSettings &settings)
- bool needsSave () const

Static Public Member Functions

- static int shareDimming ()
- static void shareDimming (int newShareDimming)

Static Public Attributes

- static const int MAX_DIM = 3

Private Member Functions

- void rebuildBaseMap ()
- void printRGB (QFile &cmd, const ColorMap &animMap)

Private Attributes

- AnimList _animList
- KbAnim * _previewAnim
- KeyMap _map
- QColorMap _qColorMap
- ColorMap _colorMap
- ColorMap _animMap
- ColorMap _lastFrameAnimMap
- ColorMap _indicatorMap
- ColorMap _lastFrameIndicatorMap
- QSet< QString > _indicatorList
- quint64 lastFrameSignal
- int _dimming
- int _lastFrameDimming
- bool _start
- bool _needsSave
- bool _needsMapRefresh

12.30.1 Detailed Description

Definition at line 17 of file kblight.h.

12.30.2 Member Typedef Documentation

12.30.2.1 typedef QList<KbAnim*> KbLight::AnimList

Definition at line 46 of file kblight.h.

12.30.3 Constructor & Destructor Documentation

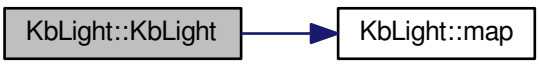
12.30.3.1 KbLight::KbLight (KbMode * *parent*, const KeyMap & *keyMap*)

Definition at line 10 of file kblight.cpp.

References map().

```
10                                                          :
11     QObject(parent), _previewAnim(0), lastFrameSignal(0),
      _dimming(0), _lastFrameDimming(0), _start(false),
      _needsSave(true), _needsMapRefresh(true)
12 {
13     map(keyMap);
14 }
```

Here is the call graph for this function:



12.30.3.2 KbLight::KbLight (KbMode * *parent*, const KeyMap & *keyMap*, const KbLight & *other*)

Definition at line 16 of file kblight.cpp.

References _animList, and map().

```
16                                                                                  :
17     QObject(parent), _previewAnim(0), _map(other._map),
      _qColorMap(other._qColorMap), lastFrameSignal(0),
      _dimming(other._dimming), _lastFrameDimming(other.
      _lastFrameDimming), _start(false), _needsSave(true),
      _needsMapRefresh(true)
18 {
19     map(keyMap);
20     // Duplicate animations
21     foreach(KbAnim* animation, other._animList)
22         _animList.append(new KbAnim(this, keyMap, *animation));
23 }
```

Here is the call graph for this function:

12.30.3.3 KbLight::~KbLight ( )

Definition at line 45 of file kblight.cpp.

References activeLights.

```
45                  {
46     activeLights.remove(this);
47 }
```

12.30.4 Member Function Documentation

12.30.4.1 KbAnim * KbLight::addAnim (const AnimScript * *base,* const QStringList & *keys,* const QString & *name,* const QMap< QString, QVariant > & *preset*)

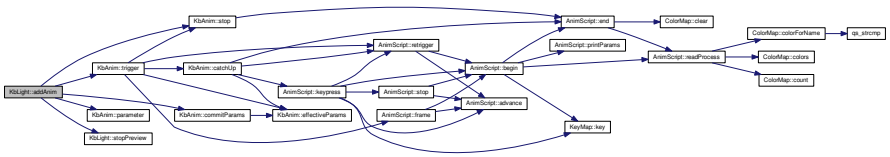
Definition at line 100 of file kblight.cpp.

References _animList, _map, _needsSave, _start, anim, KbAnim::commitParams(), KbAnim::parameter(), KbAnim::stop(), stopPreview(), and KbAnim::trigger().

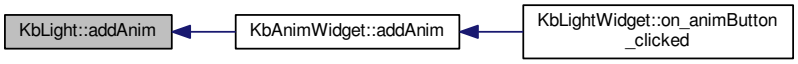
Referenced by KbAnimWidget::addAnim().

```
100
                            {
101     // Stop and restart all existing animations
102     stopPreview();
103     quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
104     foreach(KbAnim* anim, _animList){
105         anim->stop();
106         anim->trigger(timestamp);
107     }
108     // Load the new animation and set preset parameters
109     KbAnim* anim = new KbAnim(this, _map, name, keys, base);
110     QMapIterator<QString, QVariant> i(preset);
111     while(i.hasNext()){
112         i.next();
113         anim->parameter(i.key(), i.value());
114     }
115     anim->commitParams();
116     // Add the animation and start it
117     _animList.append(anim);
118     anim->trigger(timestamp);
119     _start = true;
120     _needsSave = true;
121     return anim;
122 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.2 void KbLight::animKeypress (const QString & *key*, bool *down*)

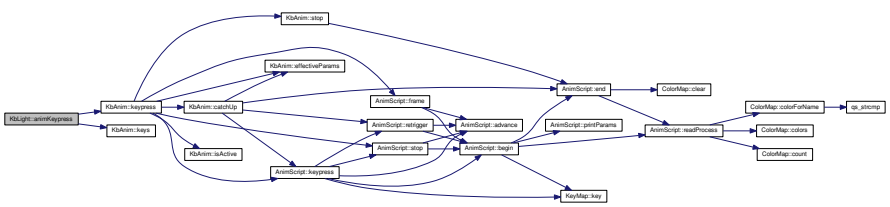
Definition at line 189 of file kblight.cpp.

References _animList, _previewAnim, anim, KbAnim::keypress(), and KbAnim::keys().

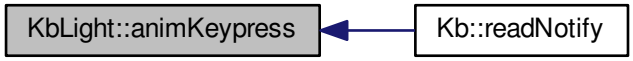
Referenced by Kb::readNotify().

```
189                                                         {
190     foreach(KbAnim* anim, _animList){
191         if(anim->keys().contains(key))
192             anim->keypress(key, down, QDateTime::currentMSecsSinceEpoch());
193     }
194     if(_previewAnim){
195         if(_previewAnim->keys().contains(key))
196             _previewAnim->keypress(key, down, QDateTime::currentMSecsSinceEpoch());
197     }
198 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.3 const AnimList& KbLight::animList ( ) [inline]

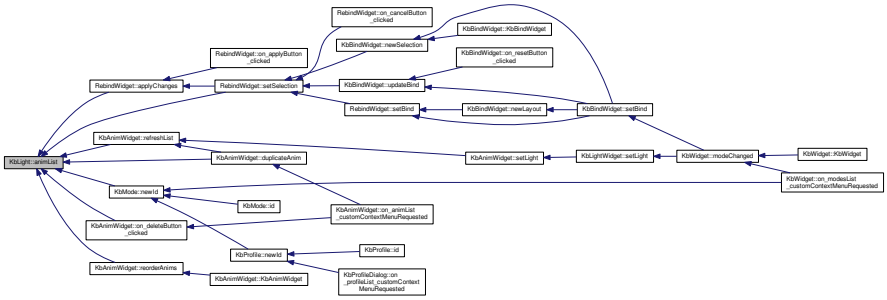
Definition at line 49 of file kblight.h.

References _animList.

Referenced by RebindWidget::applyChanges(), KbAnimWidget::duplicateAnim(), KbMode::newId(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::refreshList(), KbAnimWidget::reorderAnims(), and RebindWidget::setSelection().

```
49 { return _animList; }
```

Here is the caller graph for this function:



12.30.4.4 void KbLight::animList (const AnimList & *newAnimList*) [inline]

Definition at line 50 of file kblight.h.

References _animList, and _needsSave.

```
50 { _needsSave = true; _animList = newAnimList; }
```

12.30.4.5 void KbLight::base (QFile & *cmd*, bool *ignoreDim* = false, bool *monochrome* = false)

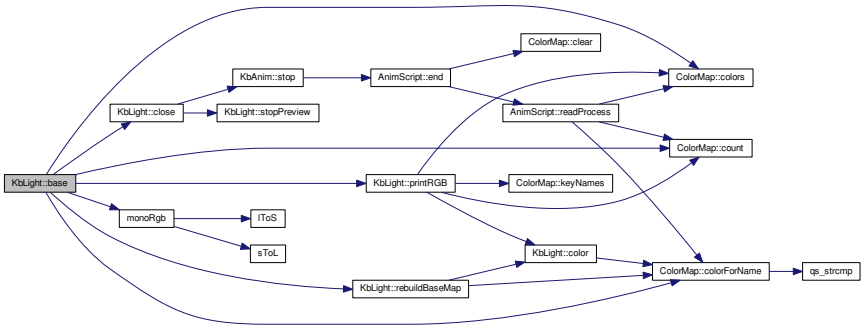
Definition at line 378 of file kblight.cpp.

References _animMap, _colorMap, _dimming, close(), ColorMap::colorForName(), ColorMap::colors(), ColorMap::count(), MAX_DIM, monoRgb(), printRGB(), and rebuildBaseMap().

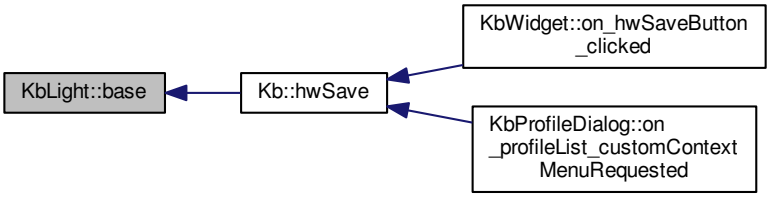
Referenced by Kb::hwSave().

```
378                                                                 {
379     close();
380     if(_dimming == MAX_DIM && !ignoreDim){
381         cmd.write(QString().sprintf("rgb 000000").toLatin1());
382         return;
383     }
384     // Set just the background color, ignoring any animation
385     rebuildBaseMap();
386     _animMap = _colorMap;
387     // If monochrome is active, create grayscale
388     if(monochrome){
389         int count = _animMap.count();
390         QRgb* colors = _animMap.colors();
391         for(int i = 0; i < count; i++){
392             QRgb& rgb = colors[i];
393             rgb = monoRgb(qRed(rgb), qGreen(rgb), qBlue(rgb));
394         }
395     }
396     // Set a few indicators to black as the hardware handles them differently
397     QRgb* mr = _animMap.colorForName("mr"), *m1 = _animMap.
      colorForName("m1"), *m2 = _animMap.colorForName("m2"), *m3 =
      _animMap.colorForName("m3"), *lock = _animMap.
      colorForName("lock");
398     if(mr) *mr = 0;
399     if(m1) *m1 = 0;
400     if(m2) *m2 = 0;
401     if(m3) *m3 = 0;
402     if(lock) *lock = 0;
403     // Send to driver
404     cmd.write("rgb");
405     printRGB(cmd, _animMap);
406 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.6 void KbLight::close ( )

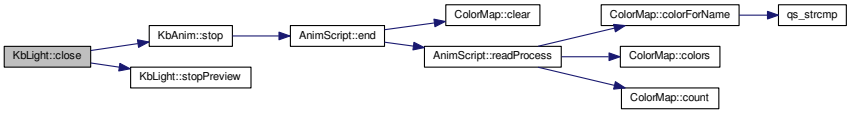
Definition at line 215 of file kblight.cpp.

References _animList, _start, activeLights, anim, KbAnim::stop(), and stopPreview().

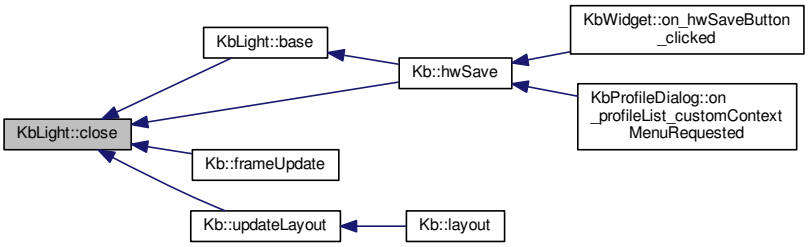
Referenced by base(), Kb::frameUpdate(), Kb::hwSave(), and Kb::updateLayout().

```
215                     {
216     activeLights.remove(this);
217     foreach(KbAnim* anim, _animList)
218         anim->stop();
219     stopPreview();
220     _start = false;
221 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.7 void KbLight::color (const QString & *key*, const QColor & *newColor*)

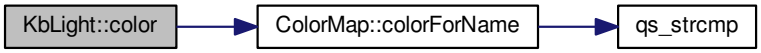
Definition at line 49 of file kblight.cpp.

References _colorMap, _needsMapRefresh, _needsSave, _qColorMap, and ColorMap::colorForName().

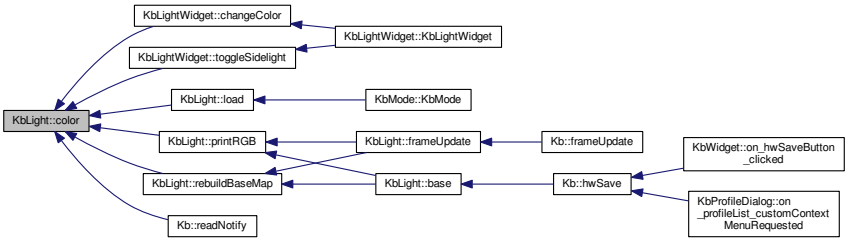
Referenced by KbLightWidget::changeColor(), load(), printRGB(), Kb::readNotify(), rebuildBaseMap(), and KbLightWidget::toggleSidelight().

```
49                                                      {
50     QRgb newRgb = newColor.rgb();
51     _qColorMap[key] = newRgb;
52     _needsSave = true;
53     if(!_needsMapRefresh){
54         // Update flat map if we're not scheduled to rebuild it
55         QByteArray rawName = key.toLatin1();
56         QRgb* rawRgb = _colorMap.colorForName(rawName.data());
57         if(rawRgb)
58             *rawRgb = newRgb;
59     }
60 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



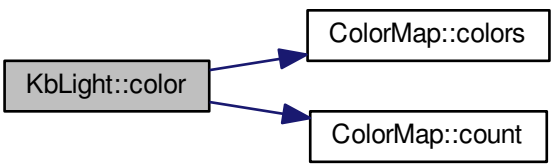
12.30.4.8 void KbLight::color (const QColor & *newColor*)

Definition at line 62 of file kblight.cpp.

References _colorMap, _needsMapRefresh, _needsSave, _qColorMap, ColorMap::colors(), and ColorMap::count().

```
62                                          {
63     QRgb newRgb = newColor.rgb();
64     QMutableColorMapIterator i(_qColorMap);
65     while(i.hasNext()){
66         i.next();
67         i.value() = newRgb;
68     }
69     _needsSave = true;
70     // Reset flat map
71     _needsMapRefresh = false;
72     int mapCount = _colorMap.count();
73     QRgb* flat = _colorMap.colors();
74     for(int i = 0; i < mapCount; i++)
75         flat[i] = mapCount;
76 }
```

Here is the call graph for this function:



12.30.4.9 const QColorMap& KbLight::colorMap ( ) [inline]

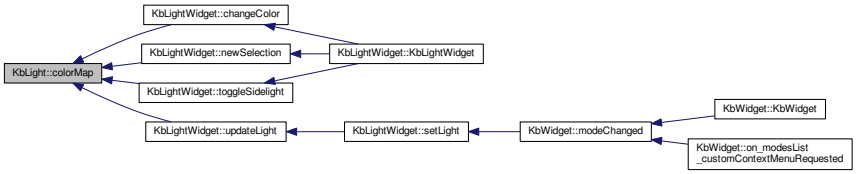
Definition at line 31 of file kblight.h.

References _qColorMap.

Referenced by KbLightWidget::changeColor(), KbLightWidget::newSelection(), KbLightWidget::toggleSidelight(), and KbLightWidget::updateLight().

```
31 { return _qColorMap; }
```

Here is the caller graph for this function:



12.30.4.10 void KbLight::didLoad ( ) [signal]

Definition at line 162 of file moc_kblight.cpp.

Referenced by load().

```
163 {
164     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
165 }
```

Here is the caller graph for this function:

12.30.4.11 int KbLight::dimming ( ) [inline]

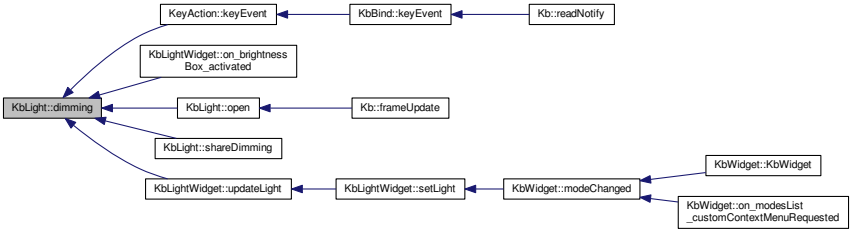
Definition at line 39 of file kblight.h.

References _dimming.

Referenced by KeyAction::keyEvent(), KbLightWidget::on_brightnessBox_activated(), open(), shareDimming(), and KbLightWidget::updateLight().

```
39 { return _dimming; }
```

Here is the caller graph for this function:



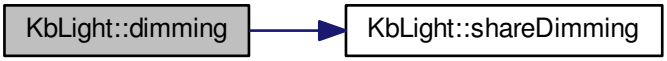
12.30.4.12 void KbLight::dimming (int *newDimming*)

Definition at line 92 of file kblight.cpp.

References _dimming, _needsSave, _shareDimming, shareDimming(), and updated().

```
92                                  {
93     if(_shareDimming != -1)
94         shareDimming(newDimming);
95     _needsSave = true;
96     _dimming = newDimming;
97     emit updated();
98 }
```

Here is the call graph for this function:



12.30.4.13 KbAnim * KbLight::duplicateAnim (KbAnim * *oldAnim*)

Definition at line 148 of file kblight.cpp.

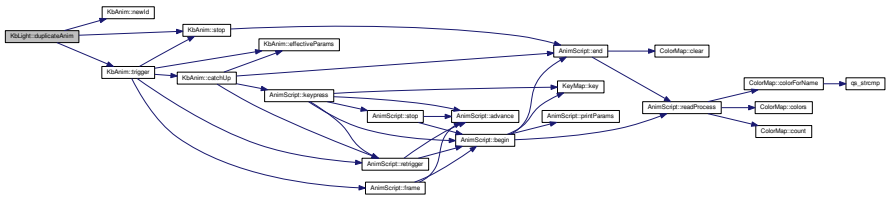
References _animList, _map, _needsSave, _start, anim, KbAnim::newId(), KbAnim::stop(), and KbAnim::trigger().

Referenced by KbAnimWidget::duplicateAnim().

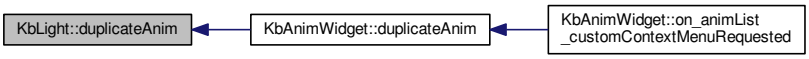
```
148                                          {
149     // Stop and restart all existing animations
150     quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
151     foreach(KbAnim* anim, _animList){
152         anim->stop();
153         anim->trigger(timestamp);
154     }
155     // Same as addAnim, just duplicate the existing one
156     KbAnim* anim = new KbAnim(this, _map, *oldAnim);
157     anim->newId();
158     int index = _animList.indexOf(oldAnim);
159     if(index < 0)
160         _animList.append(anim);
161     else
162         _animList.insert(index + 1, anim);
```

```
163     anim->trigger(timestamp);
164     _start = true;
165     _needsSave = true;
166     return anim;
167 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.14 KbAnim* KbLight::findAnim (const QUuid & *guid*) const [inline]

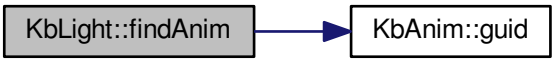
Definition at line 51 of file kblight.h.

References _animList, anim, and KbAnim::guid().

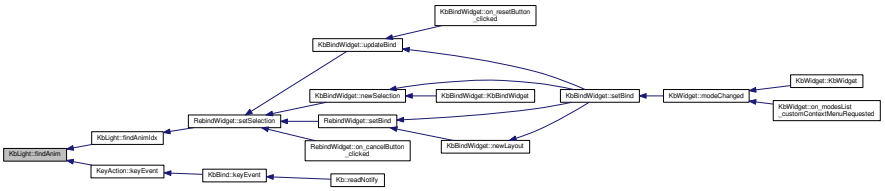
Referenced by findAnimIdx(), and KeyAction::keyEvent().

```
51 { foreach(KbAnim* anim, _animList) { if(anim->guid() == guid) return anim; } return
       0; }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.15 int KbLight::findAnimIdx (const QUuid & *guid*) const [inline]

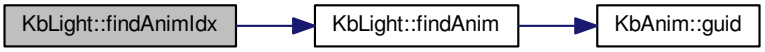
Definition at line 52 of file kblight.h.

References _animList, and findAnim().

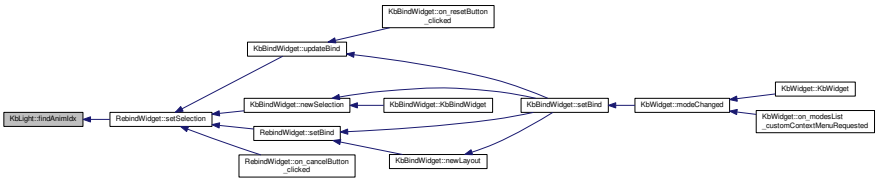
Referenced by RebindWidget::setSelection().

```
52 { return _animList.indexOf(findAnim(guid)); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.16 void KbLight::frameDisplayed (const ColorMap & *animatedColors*, const QSet< QString > & *indicatorList*) [signal]

Definition at line 174 of file moc_kblight.cpp.

Referenced by frameUpdate().

```
175 {
176     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)), const_cast<void*>(
      reinterpret_cast<const void*>(&_t2)) };
177     QMetaObject::activate(this, &staticMetaObject, 2, _a);
178 }
```

Here is the caller graph for this function:

12.30.4.17 void KbLight::frameUpdate (QFile & *cmd,* bool *monochrome* = `false`)

Definition at line 296 of file kblight.cpp.

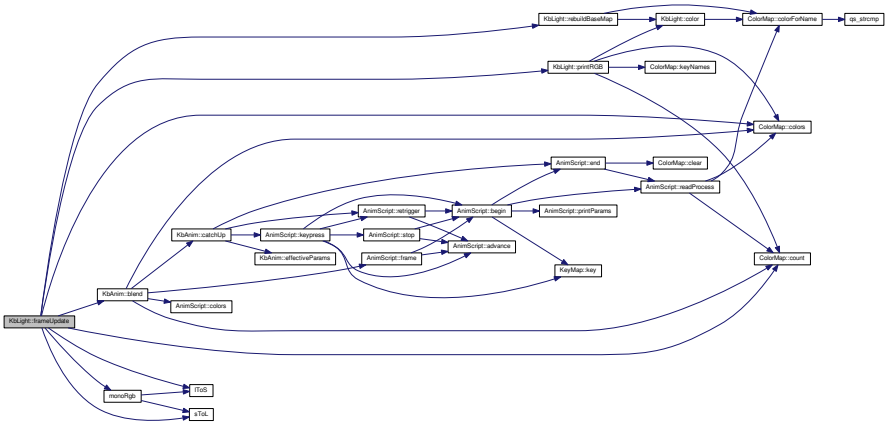
References _animList, _animMap, _colorMap, _dimming, _indicatorList, _indicatorMap, _lastFrameAnimMap, _lastFrameDimming, _lastFrameIndicatorMap, _previewAnim, anim, KbAnim::blend(), ColorMap::colors(), ColorMap::count(), frameDisplayed(), lastFrameSignal, IToS(), monoRgb(), printRGB(), rebuildBaseMap(), and sToL().

Referenced by Kb::frameUpdate().

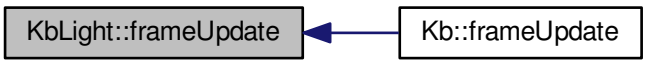
```
296                                                 {
297     rebuildBaseMap();
298     _animMap = _colorMap;
299     // Advance animations
300     quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
301     foreach(KbAnim* anim, _animList)
302         anim->blend(_animMap, timestamp);
303     if(_previewAnim)
304         _previewAnim->blend(_animMap, timestamp);
305 
306     // Avoid expensive processing if nothing has changed from the last frame.
307     if(_animMap == _lastFrameAnimMap && _indicatorMap ==
      _lastFrameIndicatorMap && _lastFrameDimming ==
      _dimming)
308         return;
309     _lastFrameAnimMap = _animMap;
310     _lastFrameIndicatorMap = _indicatorMap;
311     _lastFrameDimming = _dimming;
312 
313     int count = _animMap.count();
314     QRgb* colors = _animMap.colors();
315     // Apply active indicators and/or perform monochrome conversion
316     if(monochrome || !_indicatorList.isEmpty()){
317         QRgb* indicators = _indicatorMap.colors();
318         for(int i = 0; i < count; i++){
319             QRgb& rgb = colors[i];
320             float r = qRed(rgb);
321             float g = qGreen(rgb);
322             float b = qBlue(rgb);
323             // Apply indicators
324             QRgb rgb2 = indicators[i];
325             if(qAlpha(rgb2) != 0){
326                 float r2 = qRed(rgb2);
327                 float g2 = qGreen(rgb2);
328                 float b2 = qBlue(rgb2);
329                 float a2 = qAlpha(rgb2) / 255.f;
330                 r = round(r2 * a2 + r * (1.f - a2));
331                 g = round(g2 * a2 + g * (1.f - a2));
332                 b = round(b2 * a2 + b * (1.f - a2));
333             }
334             // If monochrome mode is active, average the channels to get a grayscale image
335             if(monochrome)
336                 rgb = monoRgb(r, g, b);
337             else
338                 rgb = qRgb(r, g, b);
339         }
340     }
341 
342     // Emit signals for the animation (only do this every 50ms - it can cause a lot of CPU usage)
343     if(timestamp >= lastFrameSignal + 50){
344         emit frameDisplayed(_animMap, _indicatorList);
345         lastFrameSignal = timestamp;
346     }
347 
348     // If brightness is at 0%, turn off lighting entirely
349     if(_dimming == 3){
```

```
350         cmd.write("rgb 000000");
351         return;
352     }
353
354     float light = (3 - _dimming) / 3.f;
355     // Apply global dimming
356     if(light != 1.f || monochrome){
357         for(int i = 0; i < count; i++){
358             QRgb& rgb = colors[i];
359             // Like the monochrome conversion, this should be done in a linear colorspace
360             float r = sToL(qRed(rgb));
361             float g = sToL(qGreen(rgb));
362             float b = sToL(qBlue(rgb));
363             r *= light;
364             g *= light;
365             b *= light;
366             r = round(lToS(r));
367             g = round(lToS(g));
368             b = round(lToS(b));
369             rgb = qRgb(r, g, b);
370         }
371     }
372
373     // Apply light
374     cmd.write("rgb");
375     printRGB(cmd, _animMap);
376 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.18 bool KbLight::isStarted ( )

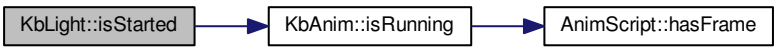
Definition at line 169 of file kblight.cpp.

References _animList, _start, and KbAnim::isRunning().

Referenced by Kb::frameUpdate().

```
169                         {
170     if(!_start)
171         return false;
172     foreach(KbAnim* animation, _animList){
173         if(!animation->isRunning())
174             return false;
175     }
176     return true;
177 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.30.4.19 void KbLight::load (CkbSettings & *settings*)

Definition at line 408 of file kblight.cpp.

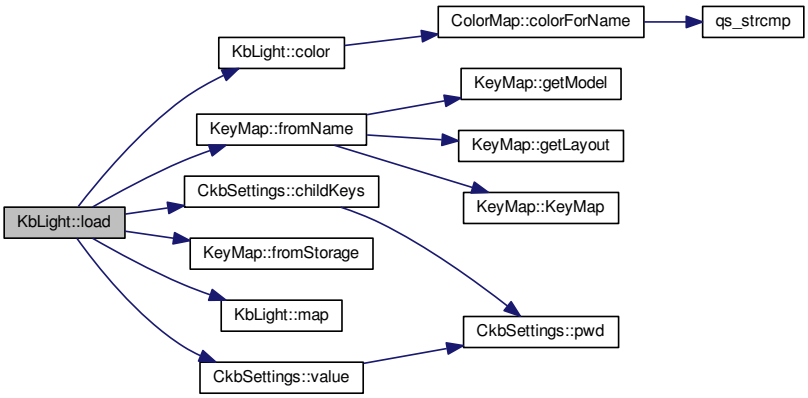
References _animList, _dimming, _map, _needsMapRefresh, _needsSave, _qColorMap, anim, CkbSettings::childKeys(), color(), didLoad(), KeyMap::fromName(), KeyMap::fromStorage(), map(), MAX_DIM, and CkbSettings::value().

Referenced by KbMode::KbMode().

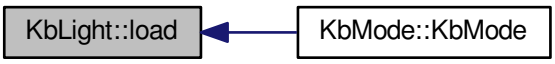
```
408                                     {
409     // Load light settings
410     _needsSave = false;
411     SGroup group(settings, "Lighting");
412     KeyMap currentMap = _map;
413     _map = KeyMap::fromName(settings.value("KeyMap").toString());
414     _dimming = settings.value("Brightness").toUInt();
415     if(_dimming > MAX_DIM)
416         _dimming = MAX_DIM;
417     // Load RGB settings
418     bool useReal = settings.value("UseRealNames").toBool();
419     {
420         SGroup group(settings, "Keys");
421         foreach(QString key, settings.childKeys()){
422             QString name = key.toLower();
423             if(!useReal)
424                 name = _map.fromStorage(name);
425             QColor color = settings.value(key).toString();
426             if(!color.isValid())
427                 color = QColor(255, 255, 255);
428             _qColorMap[name] = color.rgb();
429         }
430         _needsMapRefresh = true;
431     }
432     // Load animations
```

```
433     foreach(KbAnim* anim, _animList)
434         anim->deleteLater();
435     _animList.clear();
436     {
437         SGroup group(settings, "Animations");
438         foreach(QString anim, settings.value("List").toStringList()){
439             QUuid id = anim;
440             _animList.append(new KbAnim(this, _map, id, settings));
441         }
442     }
443     emit didLoad();
444     map(currentMap);
445 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.20 const KeyMap& KbLight::map ( ) [inline]

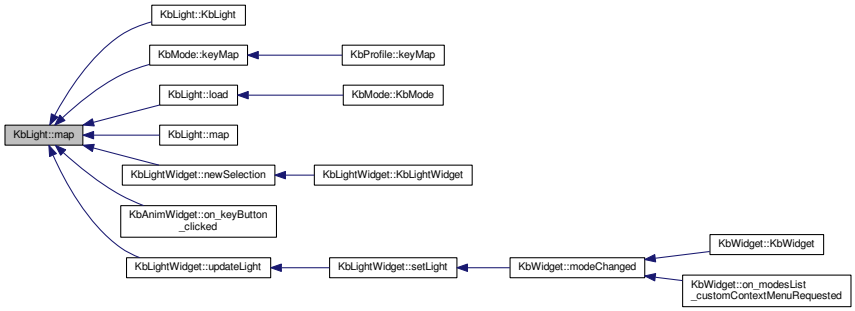
Definition at line 28 of file kblight.h.

References _map.

Referenced by KbLight(), KbMode::keyMap(), load(), map(), KbLightWidget::newSelection(), KbAnimWidget::on_keyButton_clicked(), and KbLightWidget::updateLight().

```
28 { return _map; }
```

Here is the caller graph for this function:



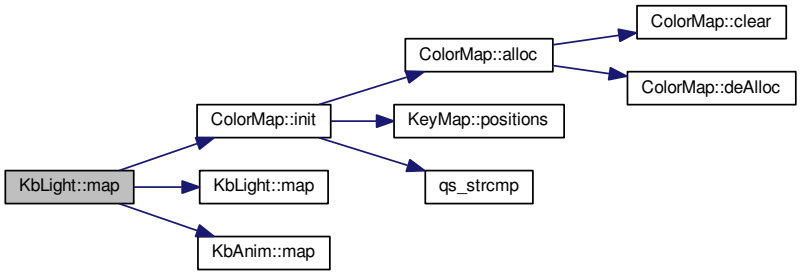
12.30.4.21 void KbLight::map (const KeyMap & *map*)

Definition at line 25 of file kblight.cpp.

References _animList, _animMap, _colorMap, _indicatorMap, _map, _needsMapRefresh, _needsSave, _qColorMap, anim, ColorMap::init(), map(), KbAnim::map(), and updated().

```
25                                  {
26     // If any of the keys are missing from the color map, set them to white
27     QHashIterator<QString, Key> i(map);
28     while(i.hasNext()){
29         i.next();
30         const QString& key = i.key();
31         if(!_qColorMap.contains(key))
32             _qColorMap[key] = 0xFFFFFFFF;
33     }
34     // Set the new map
35     _map = map;
36     foreach(KbAnim* anim, _animList)
37         anim->map(map);
38     _colorMap.init(_map);
39     _animMap.init(_map);
40     _indicatorMap.init(_map);
41     _needsSave = _needsMapRefresh = true;
42     emit updated();
43 }
```

Here is the call graph for this function:



12.30.4.22 bool KbLight::needsSave ( ) const

Definition at line 474 of file kblight.cpp.

References _animList, _needsSave, anim, and KbAnim::needsSave().

Referenced by KbMode::needsSave().

```
474                                {
475     if(_needsSave)
476         return true;
477     foreach(KbAnim* anim, _animList){
478         if(anim->needsSave())
479             return true;
480     }
481     return false;
482 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.30.4.23 void KbLight::open ( )

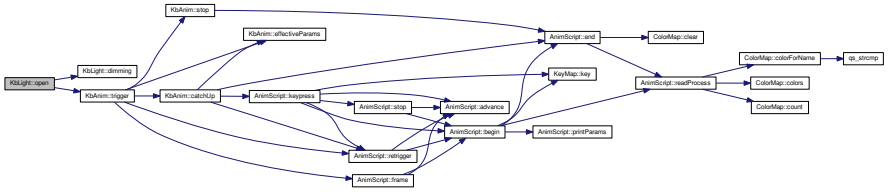
Definition at line 200 of file kblight.cpp.

References _animList, _dimming, _previewAnim, _shareDimming, _start, activeLights, anim, dimming(), and KbAnim::trigger().

Referenced by Kb::frameUpdate().

```
200                 {
201     // Apply shared dimming if needed
202     if(_shareDimming != -1 && _shareDimming !=
    _dimming)
203         dimming(_shareDimming);
204     activeLights.insert(this);
205     if(_start)
206         return;
207     quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
208     foreach(KbAnim* anim, _animList)
209         anim->trigger(timestamp);
210     if(_previewAnim)
211         _previewAnim->trigger(timestamp);
212     _start = true;
213 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.30.4.24 void KbLight::previewAnim (const AnimScript * *base*, const QStringList & *keys*, const QMap< QString, QVariant > & *preset*)

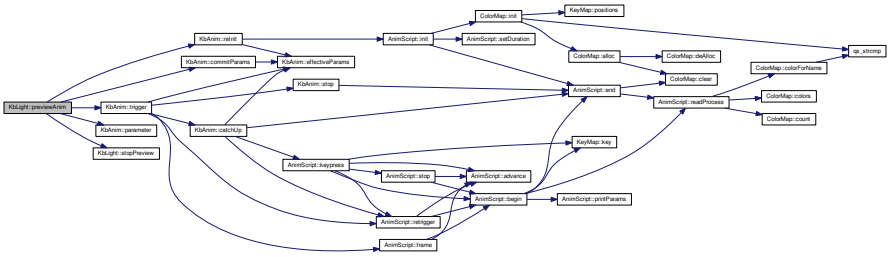
Definition at line 124 of file kblight.cpp.

References _map, _previewAnim, _start, anim, KbAnim::commitParams(), KbAnim::parameter(), KbAnim::reInit(), stopPreview(), and KbAnim::trigger().

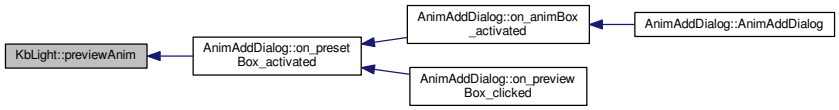
Referenced by AnimAddDialog::on_presetBox_activated().

```
124
          {
125     if(_previewAnim)
126         stopPreview();
127     quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
128     // Load the new animation and set preset parameters
129     KbAnim* anim = new KbAnim(this, _map, "", keys, base);
130     QMapIterator<QString, QVariant> i(preset);
131     while(i.hasNext()){
132         i.next();
133         anim->parameter(i.key(), i.value());
134     }
135     anim->commitParams();
136     anim->reInit();
137     // Add the animation and start it
138     _previewAnim = anim;
139     anim->trigger(timestamp);
140     _start = true;
141 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.25 void KbLight::printRGB (QFile & *cmd,* const ColorMap & *animMap*) [private]

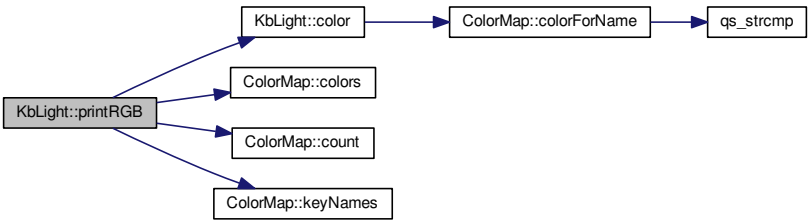
Definition at line 223 of file kblight.cpp.

References color(), ColorMap::colors(), ColorMap::count(), and ColorMap::keyNames().

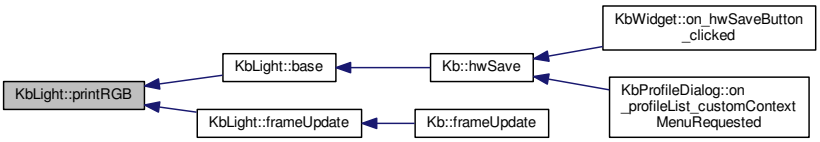
Referenced by base(), and frameUpdate().

```
223                                                         {
224     int count = animMap.count();
225     const char* const* names = animMap.keyNames();
226     const QRgb* colors = animMap.colors();
227     // Print each color and the corresponding RGB value
228     for(int i = 0; i < count; i++){
229         cmd.write(" ");
230         cmd.write(names[i]);
231         char output[8];
232         QRgb color = colors[i];
233         snprintf(output, sizeof(output), ":%02x%02x%02x", qRed(color), qGreen(color), qBlue(color));
234         cmd.write(output);
235     }
236 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.26 void KbLight::rebuildBaseMap ( ) [private]

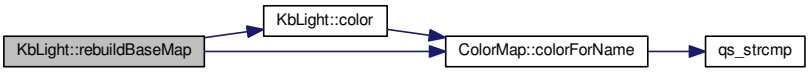
Definition at line 238 of file kblight.cpp.

References _colorMap, _needsMapRefresh, _qColorMap, color(), and ColorMap::colorForName().

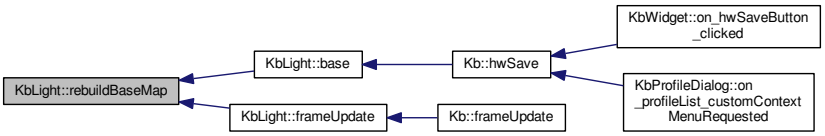
Referenced by base(), and frameUpdate().

```
238                             {
239     if(!_needsMapRefresh)
240         return;
241     _needsMapRefresh = false;
242     // Copy RGB values from QColorMap to ColorMap
243     QColorMapIterator i(_qColorMap);
244     while(i.hasNext()){
245         i.next();
246         QByteArray rawName = i.key().toLatin1();
247         QRgb color = i.value();
248         QRgb* rawColor = _colorMap.colorForName(rawName.data());
249         if(rawColor)
250             *rawColor = color;
251     }
252 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.27 void KbLight::resetIndicators ( )

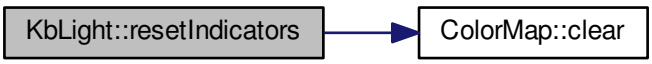
Definition at line 254 of file kblight.cpp.

References _indicatorList, _indicatorMap, and ColorMap::clear().

Referenced by KbPerf::applyIndicators().

```
254                                {
255     _indicatorMap.clear();
256     _indicatorList.clear();
257 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.30.4.28 void KbLight::restartAnimation ( )

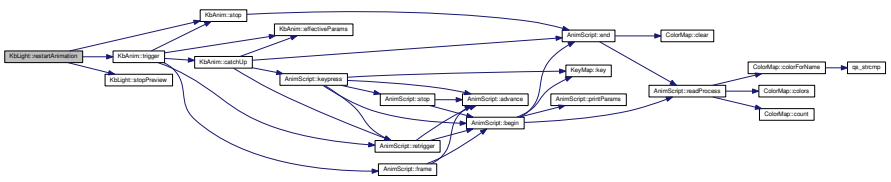
Definition at line 179 of file kblight.cpp.

References _animList, _start, anim, KbAnim::stop(), stopPreview(), and KbAnim::trigger().

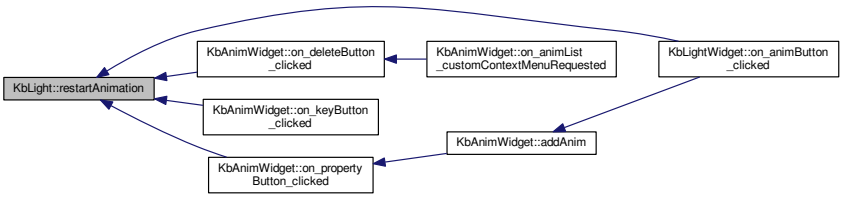
Referenced by KbLightWidget::on_animButton_clicked(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::on_keyButton_clicked(), and KbAnimWidget::on_propertyButton_clicked().

```
179                                 {
180     quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
181     foreach(KbAnim* anim, _animList){
182         anim->stop();
183         anim->trigger(timestamp);
184     }
185     stopPreview();
186     _start = true;
187 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.29 void KbLight::save (CkbSettings & *settings*)

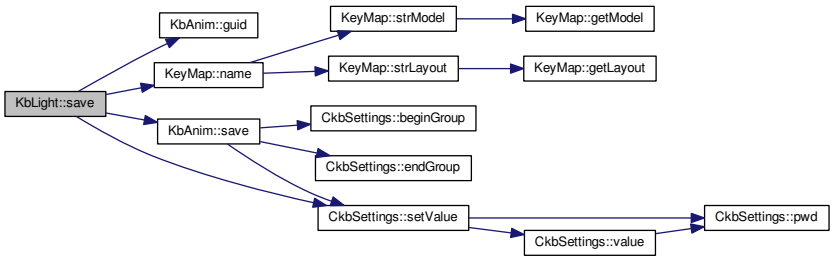
Definition at line 447 of file kblight.cpp.

References _animList, _dimming, _map, _needsSave, _qColorMap, anim, KbAnim::guid(), KeyMap::name(), KbAnim::save(), and CkbSettings::setValue().

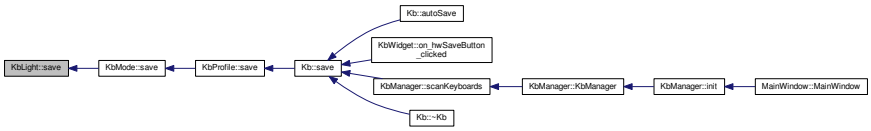
Referenced by KbMode::save().

```
447                                     {
448     _needsSave = false;
449     SGroup group(settings, "Lighting");
450     settings.setValue("KeyMap", _map.name());
451     settings.setValue("Brightness", _dimming);
452     settings.setValue("UseRealNames", true);
453     {
454         // Save RGB settings
455         SGroup group(settings, "Keys");
456         QMutableColorMapIterator i(_qColorMap);
457         while(i.hasNext()){
458             i.next();
459             settings.setValue(i.key(), QColor(i.value()).name());
460         }
461     }
462     {
463         // Save animations
464         SGroup group(settings, "Animations");
465         QStringList aList;
466         foreach(KbAnim* anim, _animList){
467             aList << anim->guid().toString().toUpper();
468             anim->save(settings);
469         }
470         settings.setValue("List", aList);
471     }
472 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.30.4.30 void KbLight::setIndicator (const char * *name*, QRgb *argb*)

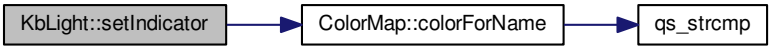
Definition at line 259 of file kblight.cpp.

References _indicatorList, _indicatorMap, and ColorMap::colorForName().

Referenced by KbPerf::lightIndicator().

```
259                                                     {
260     QRgb* dest = _indicatorMap.colorForName(name);
261     if(dest){
262         *dest = argb;
263         _indicatorList.insert(name);
264     }
265 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.30.4.31 int KbLight::shareDimming ( ) [static]

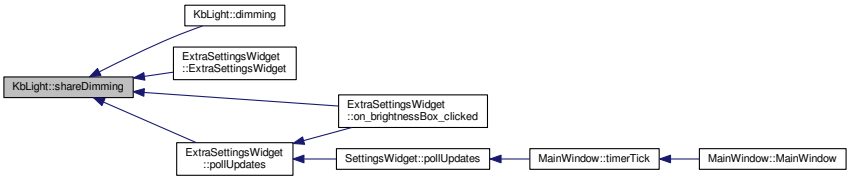
Definition at line 78 of file kblight.cpp.

References _shareDimming.

Referenced by dimming(), ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_brightnessBox_clicked(), and ExtraSettingsWidget::pollUpdates().

```
78                          {
79     return _shareDimming;
80 }
```

Here is the caller graph for this function:



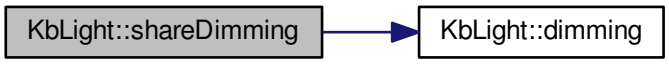
12.30.4.32 void KbLight::shareDimming (int *newShareDimming*) [static]

Definition at line 82 of file kblight.cpp.

References _shareDimming, activeLights, and dimming().

```
82                                              {
83     if(_shareDimming == newShareDimming)
84         return;
85     _shareDimming = newShareDimming;
86     if(newShareDimming != -1){
87         foreach(KbLight* light, activeLights)
88             light->dimming(newShareDimming);
89     }
90 }
```

Here is the call graph for this function:



12.30.4.33 void KbLight::stopPreview ( )

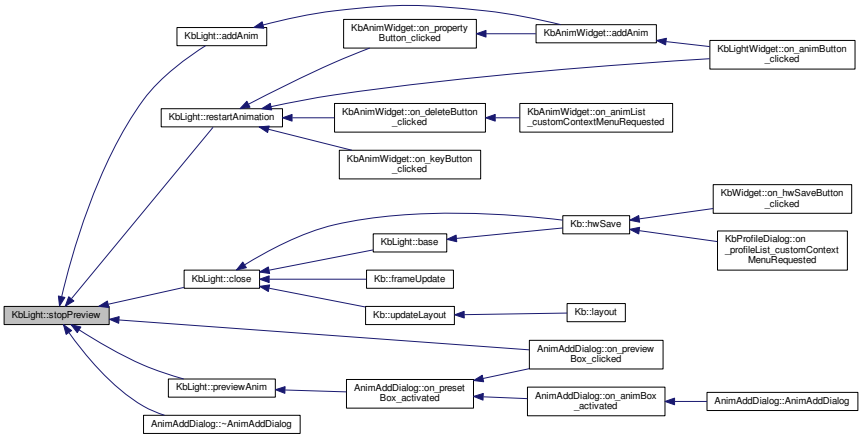
Definition at line 143 of file kblight.cpp.

References _previewAnim.

Referenced by addAnim(), close(), AnimAddDialog::on_previewBox_clicked(), previewAnim(), restartAnimation(), and AnimAddDialog::~AnimAddDialog().

```
143                         {
144     delete _previewAnim;
145     _previewAnim = 0;
146 }
```

Here is the caller graph for this function:



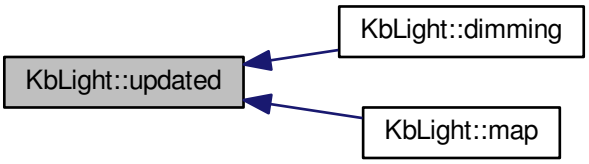
12.30.4.34 void KbLight::updated ( ) [signal]

Definition at line 168 of file moc_kblight.cpp.

Referenced by dimming(), and map().

```
169 {
170     QMetaObject::activate(this, &staticMetaObject, 1, Q_NULLPTR);
171 }
```

Here is the caller graph for this function:



12.30.5 Field Documentation

12.30.5.1 AnimList KbLight::_animList [private]

Definition at line 89 of file kblight.h.

Referenced by addAnim(), animKeypress(), animList(), close(), duplicateAnim(), findAnim(), findAnimIdx(), frameUpdate(), isStarted(), KbLight(), load(), map(), needsSave(), open(), restartAnimation(), and save().

12.30.5.2 ColorMap KbLight::_animMap [private]

Definition at line 93 of file kblight.h.

Referenced by base(), frameUpdate(), and map().

12.30.5.3 ColorMap KbLight::_colorMap [private]

Definition at line 93 of file kblight.h.

Referenced by base(), color(), frameUpdate(), map(), and rebuildBaseMap().

12.30.5.4 int KbLight::_dimming [private]

Definition at line 96 of file kblight.h.

Referenced by base(), dimming(), frameUpdate(), load(), open(), and save().

12.30.5.5 QSet<QString> KbLight::_indicatorList [private]

Definition at line 94 of file kblight.h.

Referenced by frameUpdate(), resetIndicators(), and setIndicator().

12.30.5.6 ColorMap KbLight::_indicatorMap [private]

Definition at line 93 of file kblight.h.

Referenced by frameUpdate(), map(), resetIndicators(), and setIndicator().

12.30.5.7 ColorMap KbLight::_lastFrameAnimMap [private]

Definition at line 93 of file kblight.h.

Referenced by frameUpdate().

12.30.5.8 int KbLight::_lastFrameDimming [private]

Definition at line 96 of file kblight.h.

Referenced by frameUpdate().

12.30.5.9 ColorMap KbLight::_lastFrameIndicatorMap [private]

Definition at line 93 of file kblight.h.

Referenced by frameUpdate().

12.30.5.10 KeyMap KbLight::_map [private]

Definition at line 91 of file kblight.h.

Referenced by addAnim(), duplicateAnim(), load(), map(), previewAnim(), and save().

12.30.5.11 bool KbLight::_needsMapRefresh [private]

Definition at line 98 of file kblight.h.

Referenced by color(), load(), map(), and rebuildBaseMap().

12.30.5.12 bool KbLight::_needsSave [private]

Definition at line 98 of file kblight.h.

Referenced by addAnim(), animList(), color(), dimming(), duplicateAnim(), load(), map(), needsSave(), and save().

12.30.5.13 KbAnim* KbLight::_previewAnim [private]

Definition at line 90 of file kblight.h.

Referenced by animKeypress(), frameUpdate(), open(), previewAnim(), and stopPreview().

12.30.5.14 QColorMap KbLight::_qColorMap [private]

Definition at line 92 of file kblight.h.

Referenced by color(), colorMap(), load(), map(), rebuildBaseMap(), and save().

12.30.5.15 bool KbLight::_start [private]

Definition at line 97 of file kblight.h.

Referenced by addAnim(), close(), duplicateAnim(), isStarted(), open(), previewAnim(), and restartAnimation().

12.30.5.16 quint64 KbLight::lastFrameSignal [private]

Definition at line 95 of file kblight.h.

Referenced by frameUpdate().

12.30.5.17 const int KbLight::MAX_DIM = 3 [static]

Definition at line 38 of file kblight.h.

Referenced by base(), ExtraSettingsWidget::ExtraSettingsWidget(), KeyAction::keyEvent(), and load().

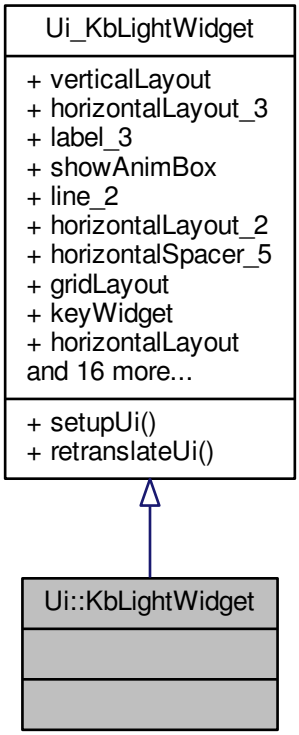
The documentation for this class was generated from the following files:

- src/ckb/kblight.h
- src/ckb/kblight.cpp
- src/ckb/release/moc_kblight.cpp

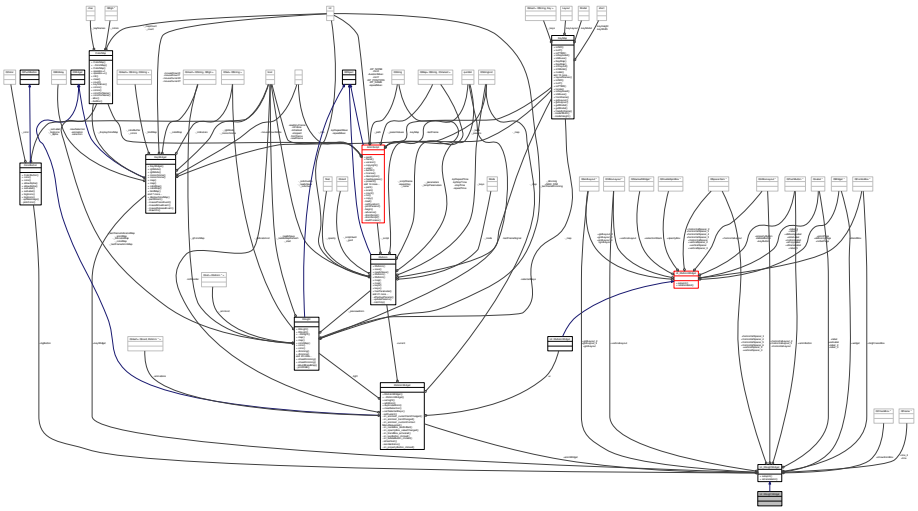
12.31 Ui::KbLightWidget Class Reference

`#include <src/ckb/ui_kblightwidget.h>`

Inheritance diagram for Ui::KbLightWidget:



Collaboration diagram for Ui::KbLightWidget:



Public Member Functions

- void setupUi (QWidget *KbLightWidget)

- void retranslateUi (QWidget *KbLightWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QHBoxLayout * horizontalLayout_3
- QLabel * label_3
- QCheckBox * showAnimBox
- QFrame * line_2
- QHBoxLayout * horizontalLayout_2
- QSpacerItem * horizontalSpacer_5
- QGridLayout * gridLayout
- KeyWidget * keyWidget
- QHBoxLayout * horizontalLayout
- ColorButton * bgButton
- QLabel * selLabel
- QPushButton * animButton
- QSpacerItem * horizontalSpacer_6
- QSpacerItem * verticalSpacer_2
- QGridLayout * gridLayout_2
- QComboBox * brightnessBox
- QLabel * label
- QSpacerItem * horizontalSpacer_3
- QSpacerItem * horizontalSpacer_4
- QSpacerItem * verticalSpacer_3
- QLabel * label_2
- QFrame * line
- QWidget * widget
- QGridLayout * gridLayout_4
- KbAnimWidget * animWidget

12.31.1 Detailed Description

Definition at line 266 of file ui_kblightwidget.h.

12.31.2 Member Function Documentation

12.31.2.1 void Ui_KbLightWidget::retranslateUi (QWidget * *KbLightWidget*) [inline],[inherited]

Definition at line 241 of file ui_kblightwidget.h.

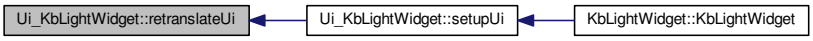
References Ui_KbLightWidget::animButton, Ui_KbLightWidget::bgButton, Ui_KbLightWidget::brightnessBox, Ui_KbLightWidget::label, Ui_KbLightWidget::label_2, Ui_KbLightWidget::label_3, Ui_KbLightWidget::selLabel, and Ui_KbLightWidget::showAnimBox.

Referenced by Ui_KbLightWidget::setupUi().

```
242     {
243         KbLightWidget->setWindowTitle(QApplication::translate("KbLightWidget", "Form", Q_NULLPTR));
244         label_3->setText(QApplication::translate("KbLightWidget", "Colors", Q_NULLPTR));
245 #ifndef QT_NO_TOOLTIP
246         showAnimBox->setToolTip(QApplication::translate("KbLightWidget", "If enabled, the same
       colors will be displayed here as on the keyboard. If disabled, the base colors will be shown.", Q_NULLPTR));
247 #endif // QT_NO_TOOLTIP
248         showAnimBox->setText(QApplication::translate("KbLightWidget", "Show animated", Q_NULLPTR
     ));
249         bgButton->setText(QApplication::translate("KbLightWidget", "Change color...", Q_NULLPTR));
250         selLabel->setText(QApplication::translate("KbLightWidget", "Click to select", Q_NULLPTR));
251         animButton->setText(QApplication::translate("KbLightWidget", "New animation...",
```

```
    Q_NULLPTR));
252         brightnessBox->clear();
253         brightnessBox->insertItems(0, QStringList()
254          << QApplication::translate("KbLightWidget", "100%", Q_NULLPTR)
255          << QApplication::translate("KbLightWidget", "67%", Q_NULLPTR)
256          << QApplication::translate("KbLightWidget", "33%", Q_NULLPTR)
257          << QApplication::translate("KbLightWidget", "Off", Q_NULLPTR)
258         );
259         label->setText(QApplication::translate("KbLightWidget", "\342\230\274", Q_NULLPTR));
260         label_2->setText(QApplication::translate("KbLightWidget", "Animations", Q_NULLPTR));
261     } // retranslateUi
```

Here is the caller graph for this function:



12.31.2.2 void Ui_KbLightWidget::setupUi (QWidget * *KbLightWidget*) [inline], [inherited]

Definition at line 63 of file ui_kblightwidget.h.

References Ui_KbLightWidget::animButton, Ui_KbLightWidget::animWidget, Ui_KbLightWidget::bgButton, Ui_KbLightWidget::brightnessBox, Ui_KbLightWidget::gridLayout, Ui_KbLightWidget::gridLayout_2, Ui_KbLightWidget::gridLayout_4, Ui_KbLightWidget::horizontalLayout, Ui_KbLightWidget::horizontalLayout_2, Ui_KbLightWidget::horizontalLayout_3, Ui_KbLightWidget::horizontalSpacer_3, Ui_KbLightWidget::horizontalSpacer_4, Ui_KbLightWidget::horizontalSpacer_5, Ui_KbLightWidget::horizontalSpacer_6, Ui_KbLightWidget::keyWidget, Ui_KbLightWidget::label, Ui_KbLightWidget::label_2, Ui_KbLightWidget::label_3, Ui_KbLightWidget::line, Ui_KbLightWidget::line_2, Ui_KbLightWidget::retranslateUi(), Ui_KbLightWidget::selLabel, Ui_KbLightWidget::showAnimBox, Ui_KbLightWidget::verticalLayout, Ui_KbLightWidget::verticalSpacer_2, Ui_KbLightWidget::verticalSpacer_3, and Ui_KbLightWidget::widget.

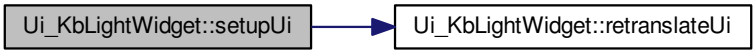
Referenced by KbLightWidget::KbLightWidget().

```
64     {
65         if (KbLightWidget->objectName().isEmpty())
66             KbLightWidget->setObjectName(QStringLiteral("KbLightWidget"));
67         KbLightWidget->resize(632, 540);
68         KbLightWidget->setFocusPolicy(Qt::ClickFocus);
69         verticalLayout = new QVBoxLayout(KbLightWidget);
70         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
71         verticalLayout->setContentsMargins(0, 0, 0, -1);
72         horizontalLayout_3 = new QHBoxLayout();
73         horizontalLayout_3->setObjectName(QStringLiteral("horizontalLayout_3"));
74         label_3 = new QLabel(KbLightWidget);
75         label_3->setObjectName(QStringLiteral("label_3"));
76         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
77         sizePolicy.setHorizontalStretch(0);
78         sizePolicy.setVerticalStretch(0);
79         sizePolicy.setHeightForWidth(label_3->sizePolicy().hasHeightForWidth());
80         label_3->setSizePolicy(sizePolicy);
81         QFont font;
82         font.setBold(true);
83         font.setWeight(75);
84         label_3->setFont(font);
85
86         horizontalLayout_3->addWidget(label_3);
87
88         showAnimBox = new QCheckBox(KbLightWidget);
89         showAnimBox->setObjectName(QStringLiteral("showAnimBox"));
90         showAnimBox->setChecked(true);
91
92         horizontalLayout_3->addWidget(showAnimBox);
93
94
95         verticalLayout->addLayout(horizontalLayout_3);
96
97         line_2 = new QFrame(KbLightWidget);
98         line_2->setObjectName(QStringLiteral("line_2"));
99         line_2->setFrameShape(QFrame::HLine);
```

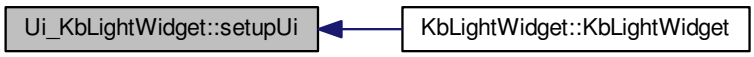
```
100         line_2->setFrameShadow(QFrame::Sunken);
101 
102         verticalLayout->addWidget(line_2);
103 
104         horizontalLayout_2 = new QHBoxLayout();
105         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
106         horizontalSpacer_5 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
107 
108         horizontalLayout_2->addItem(horizontalSpacer_5);
109 
110         gridLayout = new QGridLayout();
111         gridLayout->setObjectName(QStringLiteral("gridLayout"));
112         keyWidget = new KeyWidget(KbLightWidget);
113         keyWidget->setObjectName(QStringLiteral("keyWidget"));
114         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Expanding);
115         sizePolicy1.setHorizontalStretch(0);
116         sizePolicy1.setVerticalStretch(0);
117         sizePolicy1.setHeightForWidth(keyWidget->sizePolicy().hasHeightForWidth());
118         keyWidget->setSizePolicy(sizePolicy1);
119         keyWidget->setMinimumSize(QSize(100, 100));
120         keyWidget->setFocusPolicy(Qt::ClickFocus);
121 
122         gridLayout->addWidget(keyWidget, 0, 0, 1, 3);
123 
124         horizontalLayout = new QHBoxLayout();
125         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
126         bgButton = new ColorButton(KbLightWidget);
127         bgButton->setObjectName(QStringLiteral("bgButton"));
128         QSizePolicy sizePolicy2(QSizePolicy::Fixed, QSizePolicy::Fixed);
129         sizePolicy2.setHorizontalStretch(0);
130         sizePolicy2.setVerticalStretch(0);
131         sizePolicy2.setHeightForWidth(bgButton->sizePolicy().hasHeightForWidth());
132         bgButton->setSizePolicy(sizePolicy2);
133         bgButton->setMinimumSize(QSize(140, 0));
134         bgButton->setMaximumSize(QSize(140, 16777215));
135 
136         horizontalLayout->addWidget(bgButton);
137 
138         selLabel = new QLabel(KbLightWidget);
139         selLabel->setObjectName(QStringLiteral("selLabel"));
140         sizePolicy.setHeightForWidth(selLabel->sizePolicy().hasHeightForWidth());
141         selLabel->setSizePolicy(sizePolicy);
142         selLabel->setAlignment(Qt::AlignCenter);
143 
144         horizontalLayout->addWidget(selLabel);
145 
146         animButton = new QPushButton(KbLightWidget);
147         animButton->setObjectName(QStringLiteral("animButton"));
148         sizePolicy2.setHeightForWidth(animButton->sizePolicy().hasHeightForWidth());
149         animButton->setSizePolicy(sizePolicy2);
150         animButton->setMinimumSize(QSize(140, 0));
151         animButton->setMaximumSize(QSize(140, 16777215));
152 
153         horizontalLayout->addWidget(animButton);
154 
155 
156         gridLayout->addLayout(horizontalLayout, 1, 0, 1, 3);
157 
158 
159         horizontalLayout_2->addLayout(gridLayout);
160 
161         horizontalSpacer_6 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
162 
163         horizontalLayout_2->addItem(horizontalSpacer_6);
164 
165 
166         verticalLayout->addLayout(horizontalLayout_2);
167 
168         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
169 
170         verticalLayout->addItem(verticalSpacer_2);
171 
172         gridLayout_2 = new QGridLayout();
173         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
174         brightnessBox = new QComboBox(KbLightWidget);
175         brightnessBox->setObjectName(QStringLiteral("brightnessBox"));
176         sizePolicy2.setHeightForWidth(brightnessBox->sizePolicy().hasHeightForWidth());
177         brightnessBox->setSizePolicy(sizePolicy2);
178 
179         gridLayout_2->addWidget(brightnessBox, 1, 3, 1, 1);
180 
181         label = new QLabel(KbLightWidget);
182         label->setObjectName(QStringLiteral("label"));
183         QFont font1;
```

```
184         font1.setPointSize(20);
185         label->setFont(font1);
186         label->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
187 
188         gridLayout_2->addWidget(label, 1, 2, 1, 1);
189 
190         horizontalSpacer_3 = new QSpacerItem(0, 0, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
191 
192         gridLayout_2->addItem(horizontalSpacer_3, 0, 4, 1, 1);
193 
194         horizontalSpacer_4 = new QSpacerItem(0, 0, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
195 
196         gridLayout_2->addItem(horizontalSpacer_4, 0, 2, 1, 1);
197 
198 
199         verticalLayout->addLayout(gridLayout_2);
200 
201         verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
202 
203         verticalLayout->addItem(verticalSpacer_3);
204 
205         label_2 = new QLabel(KbLightWidget);
206         label_2->setObjectName(QStringLiteral("label_2"));
207         label_2->setFont(font);
208 
209         verticalLayout->addWidget(label_2);
210 
211         line = new QFrame(KbLightWidget);
212         line->setObjectName(QStringLiteral("line"));
213         line->setFrameShape(QFrame::HLine);
214         line->setFrameShadow(QFrame::Sunken);
215 
216         verticalLayout->addWidget(line);
217 
218         widget = new QWidget(KbLightWidget);
219         widget->setObjectName(QStringLiteral("widget"));
220         widget->setMinimumSize(QSize(0, 170));
221         gridLayout_4 = new QGridLayout(widget);
222         gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
223         gridLayout_4->setContentsMargins(-1, 0, -1, 0);
224         animWidget = new KbAnimWidget(widget);
225         animWidget->setObjectName(QStringLiteral("animWidget"));
226 
227         gridLayout_4->addWidget(animWidget, 0, 0, 1, 1);
228 
229 
230         verticalLayout->addWidget(widget);
231 
232         QWidget::setTabOrder(showAnimBox, bgButton);
233         QWidget::setTabOrder(bgButton, animButton);
234         QWidget::setTabOrder(animButton, brightnessBox);
235 
236         retranslateUi(KbLightWidget);
237 
238         QMetaObject::connectSlotsByName(KbLightWidget);
239     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.31.3 Field Documentation

12.31.3.1 QPushButton* Ui_KbLightWidget::animButton [inherited]

Definition at line 48 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), Ui_KbLightWidget::retranslateUi(), and Ui_KbLightWidget::setupUi().

12.31.3.2 KbAnimWidget* Ui_KbLightWidget::animWidget [inherited]

Definition at line 61 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), KbLightWidget::newSelection(), KbLightWidget::on_animButton_clicked(), KbLightWidget::setLight(), and Ui_KbLightWidget::setupUi().

12.31.3.3 ColorButton* Ui_KbLightWidget::bgButton [inherited]

Definition at line 46 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), KbLightWidget::newSelection(), Ui_KbLightWidget::retranslateUi(), and Ui_KbLightWidget::setupUi().

12.31.3.4 QComboBox* Ui_KbLightWidget::brightnessBox [inherited]

Definition at line 52 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::retranslateUi(), Ui_KbLightWidget::setupUi(), and KbLightWidget::updateLight().

12.31.3.5 QGridLayout* Ui_KbLightWidget::gridLayout [inherited]

Definition at line 43 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.6 QGridLayout* Ui_KbLightWidget::gridLayout_2 [inherited]

Definition at line 51 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.7 QGridLayout* Ui_KbLightWidget::gridLayout_4 [inherited]

Definition at line 60 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.8 QHBoxLayout* Ui_KbLightWidget::horizontalLayout [inherited]

Definition at line 45 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.9 QHBoxLayout* Ui_KbLightWidget::horizontalLayout_2 [inherited]

Definition at line 41 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.10 QHBoxLayout* Ui_KbLightWidget::horizontalLayout_3 [inherited]

Definition at line 37 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.11 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_3 [inherited]

Definition at line 54 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.12 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_4 [inherited]

Definition at line 55 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.13 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_5 [inherited]

Definition at line 42 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.14 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_6 [inherited]

Definition at line 49 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.15 KeyWidget* Ui_KbLightWidget::keyWidget [inherited]

Definition at line 44 of file ui_kblightwidget.h.

Referenced by KbLightWidget::changeAnim(), KbLightWidget::changeAnimKeys(), KbLightWidget::changeColor(), KbLightWidget::KbLightWidget(), KbLightWidget::on_animButton_clicked(), KbLightWidget::on_bgButton_clicked(), KbLightWidget::on_showAnimBox_clicked(), KbLightWidget::setLight(), KbLightWidget::setMonochrome(), Ui_KbLightWidget::setupUi(), KbLightWidget::toggleSidelight(), and KbLightWidget::updateLight().

12.31.3.16 QLabel* Ui_KbLightWidget::label [inherited]

Definition at line 53 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::retranslateUi(), and Ui_KbLightWidget::setupUi().

12.31.3.17 QLabel* Ui_KbLightWidget::label_2 [inherited]

Definition at line 57 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::retranslateUi(), and Ui_KbLightWidget::setupUi().

12.31.3.18 QLabel* Ui_KbLightWidget::label_3 [inherited]

Definition at line 38 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::retranslateUi(), and Ui_KbLightWidget::setupUi().

12.31.3.19 QFrame* Ui_KbLightWidget::line [inherited]

Definition at line 58 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.20 QFrame* Ui_KbLightWidget::line_2 [inherited]

Definition at line 40 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.21 QLabel* Ui_KbLightWidget::selLabel [inherited]

Definition at line 47 of file ui_kblightwidget.h.

Referenced by KbLightWidget::newSelection(), Ui_KbLightWidget::retranslateUi(), and Ui_KbLightWidget::setupUi().

12.31.3.22 QCheckBox* Ui_KbLightWidget::showAnimBox [inherited]

Definition at line 39 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), Ui_KbLightWidget::retranslateUi(), KbLightWidget::setLight(), and Ui_KbLightWidget::setupUi().

12.31.3.23 QVBoxLayout* Ui_KbLightWidget::verticalLayout [inherited]

Definition at line 36 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.24 QSpacerItem* Ui_KbLightWidget::verticalSpacer_2 [inherited]

Definition at line 50 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.25 QSpacerItem* Ui_KbLightWidget::verticalSpacer_3 [inherited]

Definition at line 56 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

12.31.3.26 QWidget* Ui_KbLightWidget::widget [inherited]

Definition at line 59 of file ui_kblightwidget.h.

Referenced by Ui_KbLightWidget::setupUi().

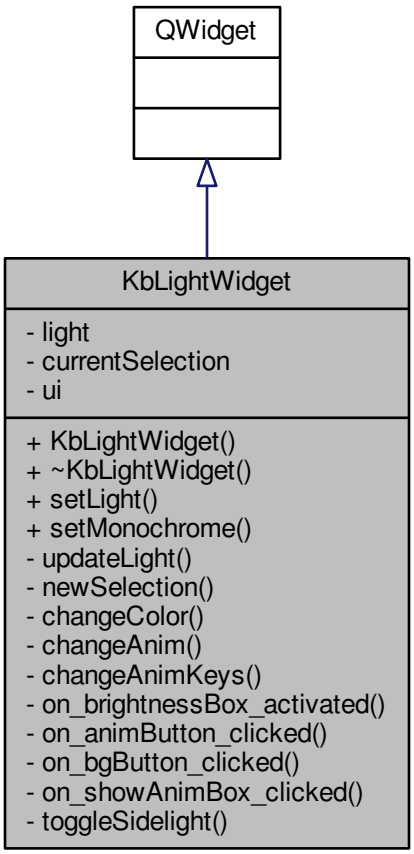
The documentation for this class was generated from the following file:

- src/ckb/ui_kblightwidget.h

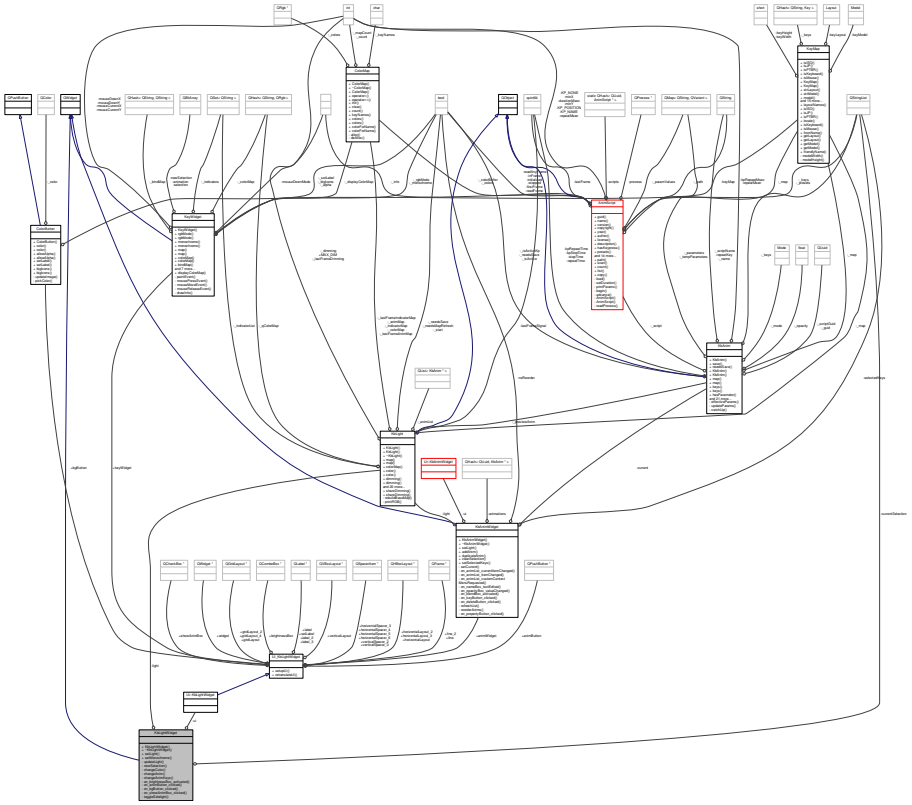
12.32 KbLightWidget Class Reference

`#include <src/ckb/kblightwidget.h>`

Inheritance diagram for KbLightWidget:



Collaboration diagram for KbLightWidget:



Public Member Functions

- KbLightWidget (QWidget ∗parent=0)
- ∼KbLightWidget ()
- void setLight (KbLight ∗newLight)
- void setMonochrome ()

Private Slots

- void updateLight ()
- void newSelection (QStringList selection)
- void changeColor (QColor newColor)
- void changeAnim (KbAnim ∗newAnim)
- void changeAnimKeys (QStringList keys)
- void on_brightnessBox_activated (int index)
- void on_animButton_clicked ()
- void on_bgButton_clicked ()
- void on_showAnimBox_clicked (bool checked)
- void toggleSidelight ()

Private Attributes

- KbLight ∗ light
- QStringList currentSelection
- Ui::KbLightWidget ∗ ui

12.32.1 Detailed Description

Definition at line 15 of file kblightwidget.h.

12.32.2 Constructor & Destructor Documentation

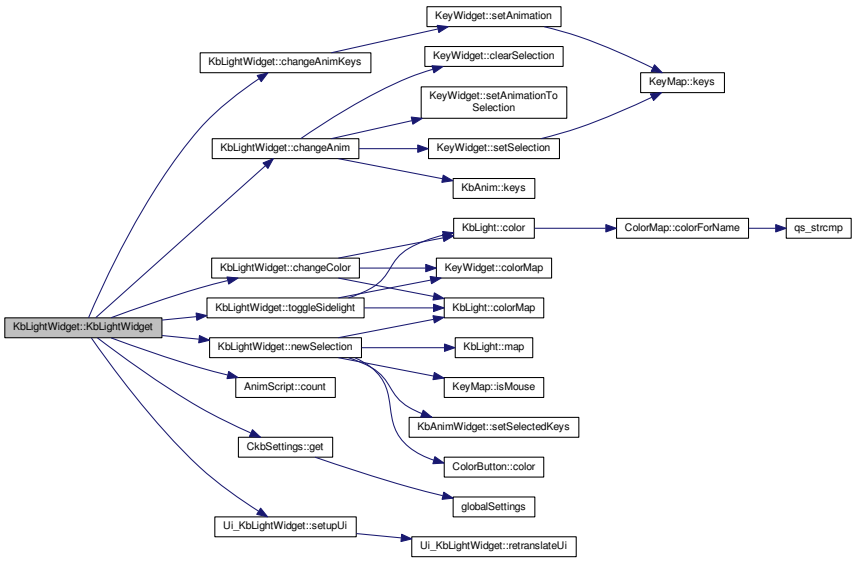
12.32.2.1 KbLightWidget::KbLightWidget (QWidget * *parent* = 0) [explicit]

Definition at line 6 of file kblightwidget.cpp.

References Ui_KbLightWidget::animButton, Ui_KbLightWidget::animWidget, Ui_KbLightWidget::bgButton, changeAnim(), changeAnimKeys(), changeColor(), AnimScript::count(), CkbSettings::get(), Ui_KbLightWidget::keyWidget, newSelection(), Ui_KbLightWidget::setupUi(), Ui_KbLightWidget::showAnimBox, toggleSidelight(), and ui.

```
6                                                :
7     QWidget(parent), light(0),
8     ui(new Ui::KbLightWidget)
9 {
10     ui->setupUi(this);
11     if(AnimScript::count() == 0)
12         ui->animButton->setVisible(false);
13
14     connect(ui->bgButton, SIGNAL(colorChanged(QColor)), this, SLOT(
      changeColor(QColor)));
15     connect(ui->keyWidget, SIGNAL(selectionChanged(QStringList)), this, SLOT(
      newSelection(QStringList)));
16     connect(ui->keyWidget, SIGNAL(sidelightToggled()), this, SLOT(
      toggleSidelight())); // click on a toggle button, like sidelight
17     connect(ui->animWidget, SIGNAL(animChanged(KbAnim*)), this, SLOT(
      changeAnim(KbAnim*)));
18     connect(ui->animWidget, SIGNAL(didUpdateSelection(QStringList)), this, SLOT(
      changeAnimKeys(QStringList)));
19
20     // Restore "show animated" setting
21     ui->showAnimBox->setChecked(!CkbSettings::get("UI/Light/ShowBaseOnly").
      toBool());
22 }
```

Here is the call graph for this function:



12.32.2.2 KbLightWidget::~KbLightWidget ( )

Definition at line 24 of file kblightwidget.cpp.

References ui.

```
24                                {
25     delete ui;
26 }
```

12.32.3 Member Function Documentation

12.32.3.1 void KbLightWidget::changeAnim (KbAnim * *newAnim*) [private], [slot]

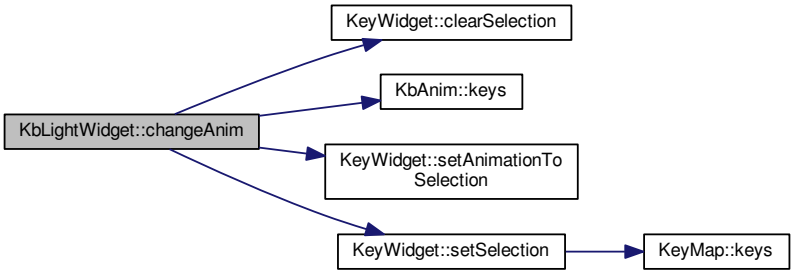
Definition at line 121 of file kblightwidget.cpp.

References KeyWidget::clearSelection(), KbAnim::keys(), Ui_KbLightWidget::keyWidget, KeyWidget::setAnimationToSelection(), KeyWidget::setSelection(), and ui.

Referenced by KbLightWidget().

```
121                                              {
122     if(newAnim)
123         ui->keyWidget->setSelection(newAnim->keys());
124     else
125         ui->keyWidget->clearSelection();
126     ui->keyWidget->setAnimationToSelection();
127 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.32.3.2 void KbLightWidget::changeAnimKeys (QStringList *keys*) [private], [slot]

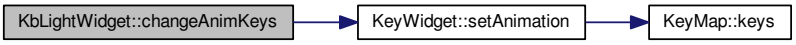
Definition at line 129 of file kblightwidget.cpp.

References Ui_KbLightWidget::keyWidget, KeyWidget::setAnimation(), and ui.

Referenced by KbLightWidget().

```
129                                                 {
130     ui->keyWidget->setAnimation(keys);
131 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.32.3.3 void KbLightWidget::changeColor (QColor *newColor*) [private], [slot]

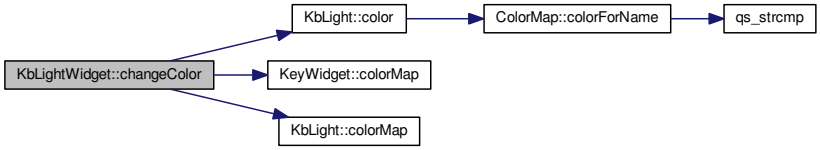
Definition at line 99 of file kblightwidget.cpp.

References KbLight::color(), KeyWidget::colorMap(), KbLight::colorMap(), currentSelection, Ui_KbLightWidget::keyWidget, light, and ui.

Referenced by KbLightWidget().

```
99                                                  {
100     if(light){
101         foreach(QString key, currentSelection)
102             light->color(key, newColor);
103         ui->keyWidget->colorMap(light->colorMap());
104     }
105 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.32.3.4 void KbLightWidget::newSelection (QStringList *selection*) [private], [slot]

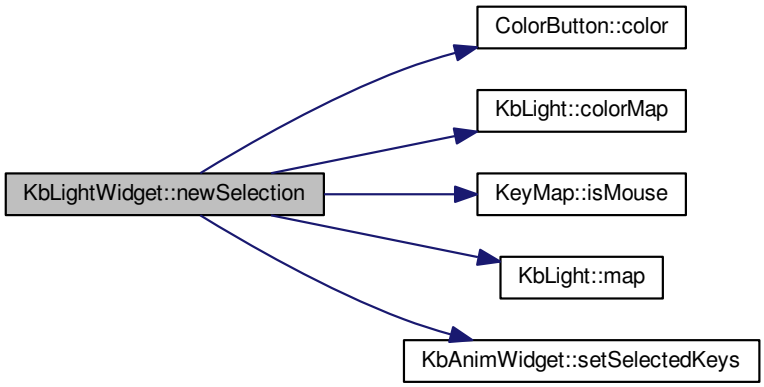
Definition at line 72 of file kblightwidget.cpp.

References Ui_KbLightWidget::animWidget, Ui_KbLightWidget::bgButton, ColorButton::color(), KbLight::colorMap(), currentSelection, KeyMap::isMouse(), light, KbLight::map(), Ui_KbLightWidget::selLabel, KbAnimWidget::setSelectedKeys(), and ui.

Referenced by KbLightWidget().

```
72                                                  {
73     // Determine selected color (invalid color if no selection or if they're not all the same)
74     QColor selectedColor;
75     const QColorMap& colorMap = light->colorMap();
76     foreach(const QString& key, selection){
77         QColor color = colorMap.value(key);
78         if(!selectedColor.isValid())
79             selectedColor = color;
80         else if(color != selectedColor){
81             selectedColor = QColor();
82             break;
83         }
84     }
85
86     currentSelection = selection;
87     ui->animWidget->setSelectedKeys(selection);
88     ui->bgButton->color(selectedColor);
89     int count = selection.count();
90     if(count == 0){
91         ui->selLabel->setText("Click to select");
92         return;
93     } else if(count == 1)
94         ui->selLabel->setText(tr("1 %1 selected").arg(light->map().
      isMouse() ? "zone" : "key"));
95     else
96         ui->selLabel->setText(tr("%1 %2 selected").arg(count).arg(
      light->map().isMouse() ? "zones" : "keys"));
97 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

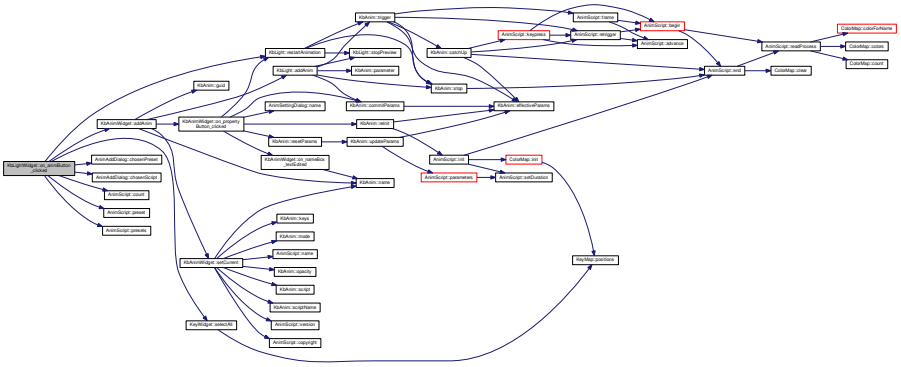
12.32.3.5 void KbLightWidget::on_animButton_clicked ( ) [private],[slot]

Definition at line 143 of file kblightwidget.cpp.

References KbAnimWidget::addAnim(), Ui_KbLightWidget::animWidget, AnimAddDialog::chosenPreset(), AnimAddDialog::chosenScript(), AnimScript::count(), currentSelection, Ui_KbLightWidget::keyWidget, light, AnimScript::preset(), AnimScript::presets(), KbLight::restartAnimation(), KeyWidget::selectAll(), and ui.

```
143                                         {
144     if(AnimScript::count() == 0)
145         return;
146     if(currentSelection.isEmpty())
147         ui->keyWidget->selectAll();
148     AnimAddDialog dialog(this, light, currentSelection);
149     dialog.exec();
150     if(dialog.result() != QDialog::Accepted)
151         return;
152     const AnimScript* script = dialog.chosenScript();
153     int presetId = dialog.chosenPreset();
154     ui->animWidget->addAnim(script, currentSelection, script->
      presets()[presetId], script->preset(presetId));
155     light->restartAnimation();
156 }
```

Here is the call graph for this function:



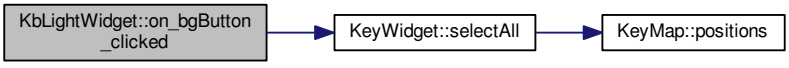
12.32.3.6 void KbLightWidget::on_bgButton_clicked ( ) [private], [slot]

Definition at line 138 of file kblightwidget.cpp.

References currentSelection, Ui_KbLightWidget::keyWidget, KeyWidget::selectAll(), and ui.

```
138                                         {
139     if(currentSelection.isEmpty())
140         ui->keyWidget->selectAll();
141 }
```

Here is the call graph for this function:



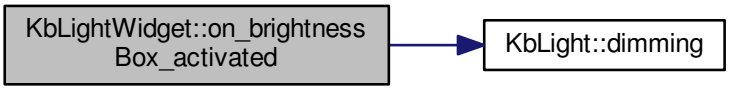
12.32.3.7 void KbLightWidget::on_brightnessBox_activated (int *index*) [private], [slot]

Definition at line 133 of file kblightwidget.cpp.

References KbLight::dimming(), and light.

```
133                                                     {
134     if(light)
135         light->dimming(index);
136 }
```

Here is the call graph for this function:



12.32.3.8 void KbLightWidget::on_showAnimBox_clicked (bool *checked*) [private],[slot]

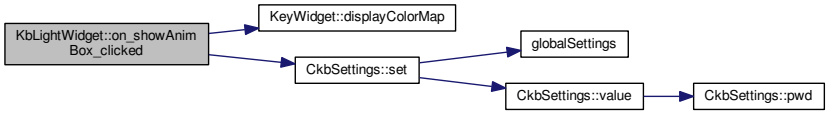
Definition at line 52 of file kblightwidget.cpp.

References KeyWidget::displayColorMap(), Ui_KbLightWidget::keyWidget, light, CkbSettings::set(), and ui.

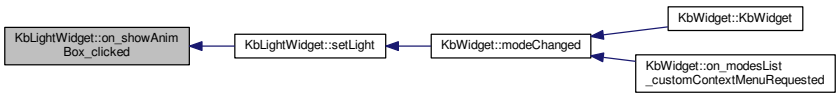
Referenced by setLight().

```
52                                                  {
53     // FIXME: set globally if there's more than one KbLightWidget active
54     // Connect/disconnect animation slot
55     if(checked){
56         if(light)
57             connect(light, SIGNAL(frameDisplayed(const ColorMap&,const QSet<QString>&)),
       ui->keyWidget, SLOT(displayColorMap(const ColorMap&,const QSet<QString>&)));
58     } else {
59         if(light)
60             disconnect(light, SIGNAL(frameDisplayed(const ColorMap&,const QSet<QString>&)),
       ui->keyWidget, SLOT(displayColorMap(const ColorMap&,const QSet<QString>&)));
61         ui->keyWidget->displayColorMap(ColorMap());
62     }
63     CkbSettings::set("UI/Light/ShowBaseOnly", !checked);
64 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.32.3.9 void KbLightWidget::setLight (KbLight * *newLight*)

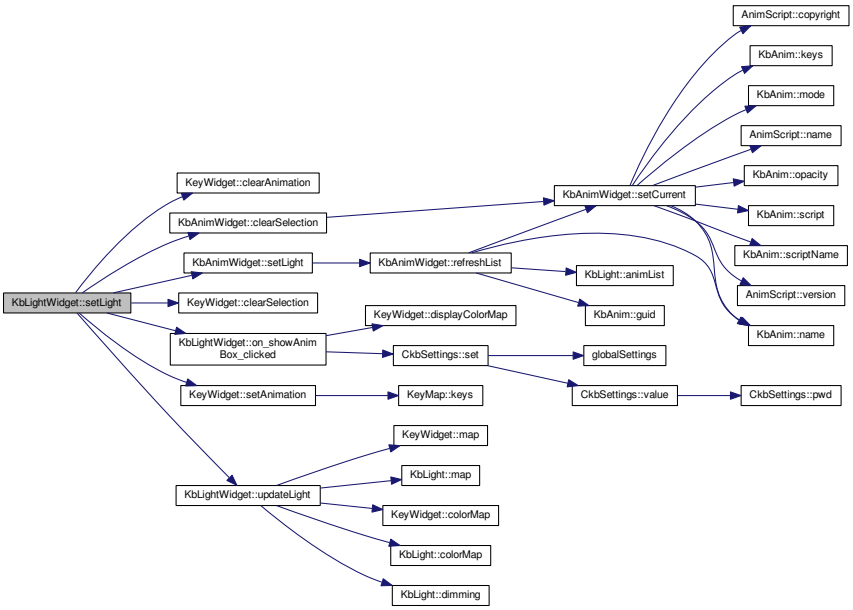
Definition at line 28 of file kblightwidget.cpp.

References Ui_KbLightWidget::animWidget, KeyWidget::clearAnimation(), KbAnimWidget::clearSelection(), KeyWidget::clearSelection(), Ui_KbLightWidget::keyWidget, light, on_showAnimBox_clicked(), KeyWidget::setAnimation(), KbAnimWidget::setLight(), Ui_KbLightWidget::showAnimBox, ui, and updateLight().

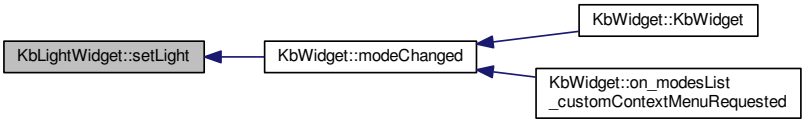
Referenced by KbWidget::modeChanged().

```
28                                          {
29     ui->keyWidget->clearSelection();
30     ui->keyWidget->clearAnimation();
31     ui->animWidget->clearSelection();
32     if(light == newLight)
33         return;
34     if(light){
35         disconnect(light, SIGNAL(updated()), this, SLOT(updateLight()));
36         // Disabled animated colors for previous light
37         on_showAnimBox_clicked(false);
38     }
39     connect(newLight, SIGNAL(updated()), this, SLOT(updateLight()));
40     light = newLight;
41     updateLight();
42     ui->keyWidget->setAnimation(QStringList());
43     ui->animWidget->setLight(newLight);
44     if(ui->showAnimBox->isChecked())
45         on_showAnimBox_clicked(true);
46 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.32.3.10 void KbLightWidget::setMonochrome ( )

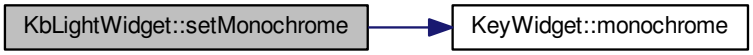
Definition at line 48 of file kblightwidget.cpp.

References Ui_KbLightWidget::keyWidget, KeyWidget::monochrome(), and ui.

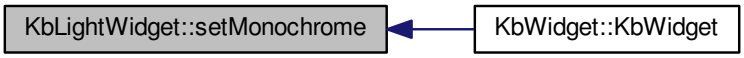
Referenced by KbWidget::KbWidget().

```
48                                  {
49     ui->keyWidget->monochrome(true);
50 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.32.3.11 void KbLightWidget::toggleSidelight ( ) [private], [slot]

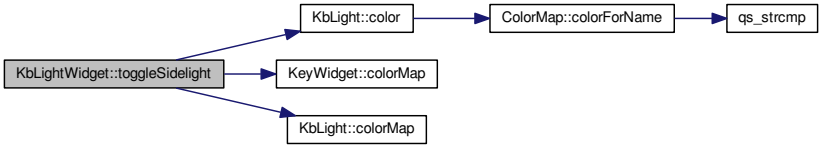
Definition at line 107 of file kblightwidget.cpp.

References KbLight::color(), KeyWidget::colorMap(), KbLight::colorMap(), Ui_KbLightWidget::keyWidget, light, and ui.

Referenced by KbLightWidget().

```
107                                     {
108     if(light){
109         if (light->colorMap()["lsidel"] == 0xFF000000){
110                 light->color("lsidel",QRgb(0xFFFFFFFF));
111                 light->color("rsidel",QRgb(0xFFFFFFFF));
112         } else {
113                 light->color("lsidel", QRgb(0xFF000000));
114                 light->color("rsidel", QRgb(0xFF000000));
115         }
116         ui->keyWidget->colorMap(light->colorMap());
117     }
118 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.32.3.12 void KbLightWidget::updateLight ( ) [private], [slot]

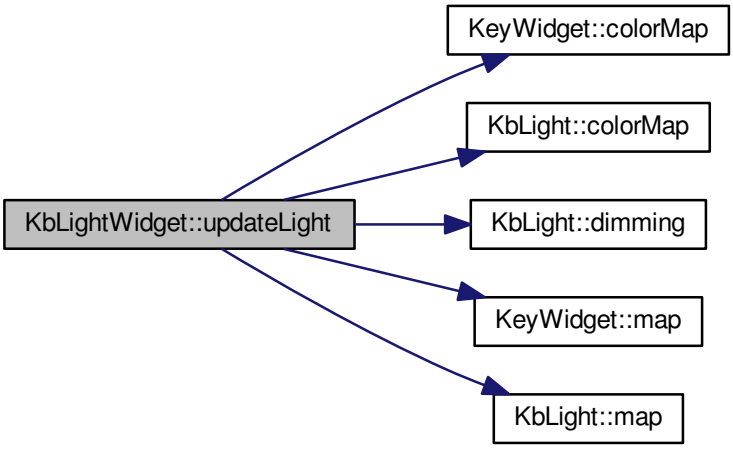
Definition at line 66 of file kblightwidget.cpp.

References Ui_KbLightWidget::brightnessBox, KeyWidget::colorMap(), KbLight::colorMap(), KbLight::dimming(), Ui_KbLightWidget::keyWidget, light, KeyWidget::map(), KbLight::map(), and ui.

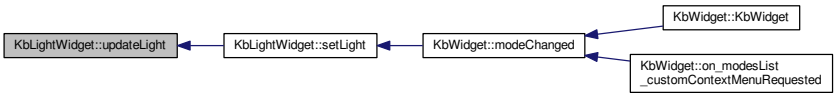
Referenced by setLight().

```
66                              {
67     ui->keyWidget->map(light->map());
68     ui->keyWidget->colorMap(light->colorMap());
69     ui->brightnessBox->setCurrentIndex(light->dimming());
70 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.32.4 Field Documentation

12.32.4.1 QStringList KbLightWidget::currentSelection [private]

Definition at line 44 of file kblightwidget.h.

Referenced by changeColor(), newSelection(), on_animButton_clicked(), and on_bgButton_clicked().

12.32.4.2 KbLight* KbLightWidget::light [private]

Definition at line 43 of file kblightwidget.h.

Referenced by changeColor(), newSelection(), on_animButton_clicked(), on_brightnessBox_activated(), on_showAnimBox_clicked(), setLight(), toggleSidelight(), and updateLight().

12.32.4.3 Ui::KbLightWidget* KbLightWidget::ui [private]

Definition at line 46 of file kblightwidget.h.

Referenced by changeAnim(), changeAnimKeys(), changeColor(), KbLightWidget(), newSelection(), on_animButton_clicked(), on_bgButton_clicked(), on_showAnimBox_clicked(), setLight(), setMonochrome(), toggleSidelight(), updateLight(), and ~KbLightWidget().

The documentation for this class was generated from the following files:

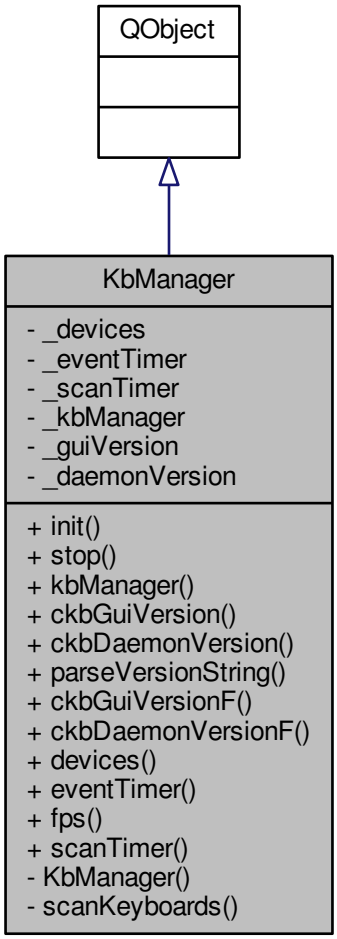
- src/ckb/kblightwidget.h

- src/ckb/kblightwidget.cpp

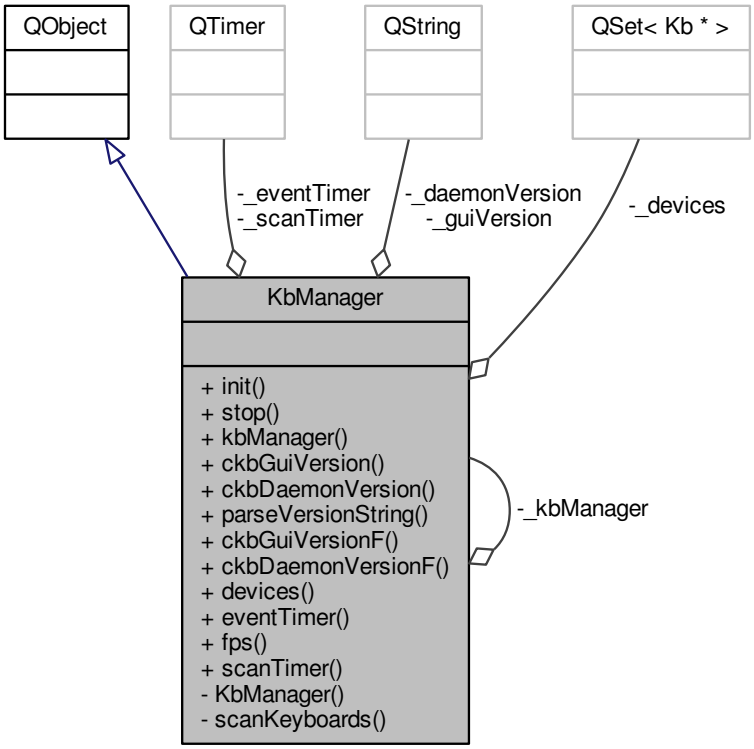
12.33 KbManager Class Reference

`#include <src/ckb/kbmanager.h>`

Inheritance diagram for KbManager:



Collaboration diagram for KbManager:



Signals

- void kbConnected (Kb ∗device)
- void kbDisconnected (Kb ∗device)
- void versionUpdated ()

Static Public Member Functions

- static void init (QString guiVersion)
- static void stop ()
- static KbManager ∗ kbManager ()
- static QString ckbGuiVersion ()
- static QString ckbDaemonVersion ()
- static float parseVersionString (QString version)
- static float ckbGuiVersionF ()
- static float ckbDaemonVersionF ()
- static const QSet< Kb ∗ > devices ()
- static QTimer ∗ eventTimer ()
- static void fps (int framerate)
- static QTimer ∗ scanTimer ()

Private Slots

- void scanKeyboards ()

Private Member Functions

- KbManager (QObject *parent=0)

Private Attributes

- QSet< Kb * > _devices
- QTimer * _eventTimer
- QTimer * _scanTimer

Static Private Attributes

- static KbManager * _kbManager = 0
- static QString _guiVersion
- static QString _daemonVersion = "<unavailable>"

12.33.1 Detailed Description

Definition at line 14 of file kbmanager.h.

12.33.2 Constructor & Destructor Documentation

12.33.2.1 KbManager::KbManager (QObject * *parent* = 0) [explicit], [private]

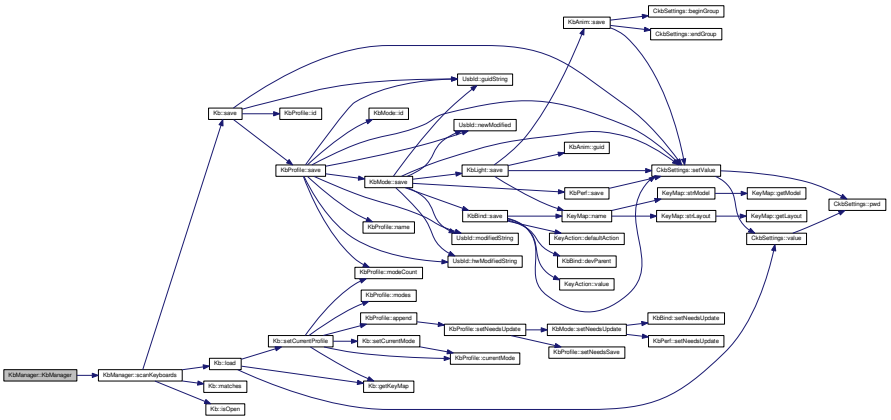
Definition at line 26 of file kbmanager.cpp.

References _eventTimer, _scanTimer, and scanKeyboards().

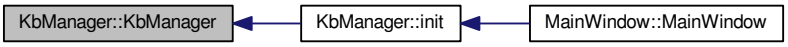
Referenced by init().

```
26                                  : QObject(parent){
27     // Set up the timers
28     _eventTimer = new QTimer(this);
29     _eventTimer->setTimerType(Qt::PreciseTimer);
30     _scanTimer = new QTimer(this);
31     _scanTimer->start(1000);
32     connect(_scanTimer, SIGNAL(timeout()), this, SLOT(scanKeyboards()));
33
34     // Scan for keyboards immediately so they show up as soon as the GUI appears.
35     QTimer::singleShot(0,this,SLOT(scanKeyboards()));
36 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.33.3 Member Function Documentation

12.33.3.1 static QString KbManager::ckbDaemonVersion ( ) [inline], [static]

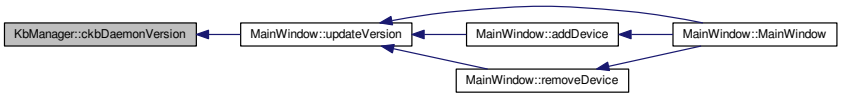
Definition at line 27 of file kbmanager.h.

References _daemonVersion.

Referenced by MainWindow::updateVersion().

```
27 { return _daemonVersion; }
```

Here is the caller graph for this function:



12.33.3.2 static float KbManager::ckbDaemonVersionF ( ) [inline], [static]

Definition at line 31 of file kbmanager.h.

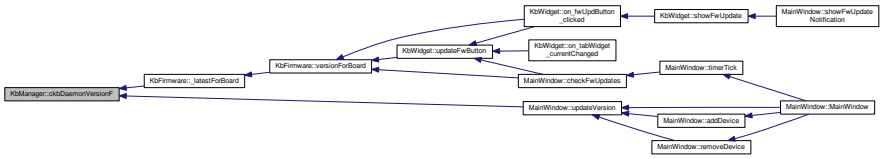
References _daemonVersion, DAEMON_UNAVAILABLE_STR, and parseVersionString().

Referenced by KbFirmware::_latestForBoard(), and MainWindow::updateVersion().

```
31 { return _daemonVersion == DAEMON_UNAVAILABLE_STR ? INFINITY :
      parseVersionString(_daemonVersion); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.33.3.3 static QString KbManager::ckbGuiVersion ( ) [inline], [static]

Definition at line 26 of file kbmanager.h.

References _guiVersion.

```
26 { return _guiVersion; }
```

12.33.3.4 static float KbManager::ckbGuiVersionF ( ) [inline], [static]

Definition at line 30 of file kbmanager.h.

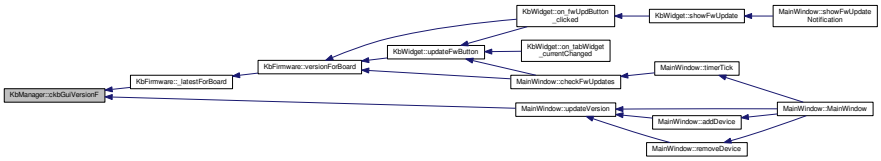
References _guiVersion, and parseVersionString().

Referenced by KbFirmware::_latestForBoard(), and MainWindow::updateVersion().

```
30 { return parseVersionString(_guiVersion); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.33.3.5 static const QSet<Kb*> KbManager::devices ( ) [inline], [static]

Definition at line 34 of file kbmanager.h.

References _devices, and _kbManager.

```
34 { return _kbManager ? _kbManager->_devices : QSet<Kb*>(); }
```

12.33.3.6 static QTimer* KbManager::eventTimer ( ) [inline], [static]

Definition at line 38 of file kbmanager.h.

References _eventTimer, and _kbManager.

Referenced by fps().

```
38 { return _kbManager ? _kbManager->_eventTimer : 0; }
```

Here is the caller graph for this function:

12.33.3.7 void KbManager::fps (int *framerate*) [static]

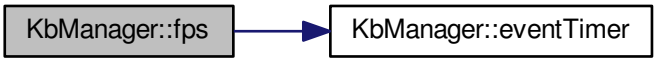
Definition at line 38 of file kbmanager.cpp.

References eventTimer().

Referenced by Kb::frameRate().

```
38                                {
39     QTimer* timer = eventTimer();
40     if(!timer)
41         return;
42     if(timer->isActive())
43         timer->setInterval(1000 / framerate);
44     else
45         timer->start(1000 / framerate);
46 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.33.3.8 void KbManager::init (QString *guiVersion*) [static]

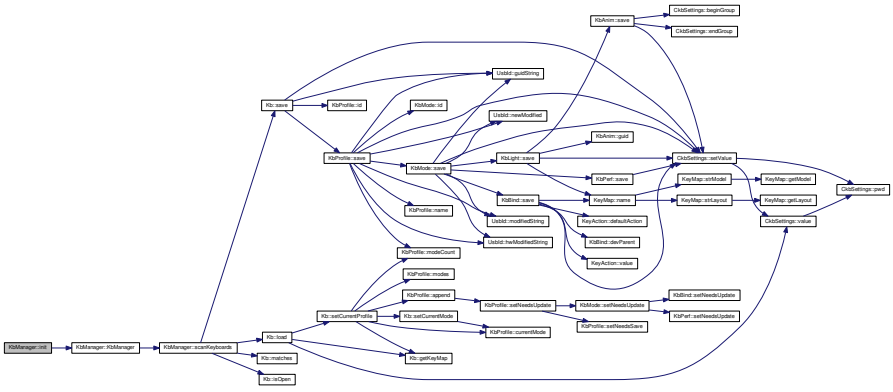
Definition at line 12 of file kbmanager.cpp.

References _guiVersion, _kbManager, and KbManager().

Referenced by MainWindow::MainWindow().

```
12                                          {
13     _guiVersion = guiVersion;
14     if(_kbManager)
15         return;
16     _kbManager = new KbManager();
17 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

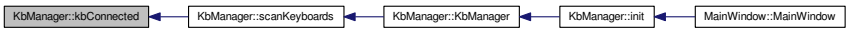
12.33.3.9 void KbManager::kbConnected (Kb * *device*) [signal]

Definition at line 174 of file moc_kbmanager.cpp.

Referenced by scanKeyboards().

```
175 {
176     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
177     QMetaObject::activate(this, &staticMetaObject, 0, _a);
178 }
```

Here is the caller graph for this function:



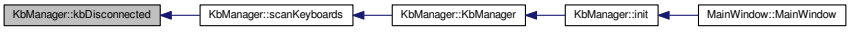
12.33.3.10 void KbManager::kbDisconnected (Kb * *device*) [signal]

Definition at line 181 of file moc_kbmanager.cpp.

Referenced by scanKeyboards().

```
182 {
183     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
184     QMetaObject::activate(this, &staticMetaObject, 1, _a);
185 }
```

Here is the caller graph for this function:



12.33.3.11 static KbManager* KbManager::kbManager ( ) [inline], [static]

Definition at line 23 of file kbmanager.h.

References _kbManager.

Referenced by MainWindow::MainWindow().

```
23 { return _kbManager; }
```

Here is the caller graph for this function:

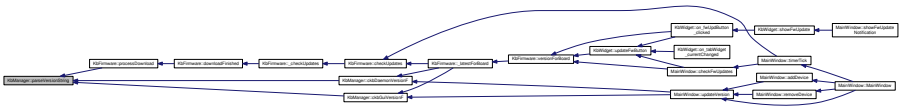
12.33.3.12 float KbManager::parseVersionString (QString *version*) [static]

Definition at line 48 of file kbmanager.cpp.

Referenced by ckbDaemonVersionF(), ckbGuiVersionF(), and KbFirmware::processDownload().

```
48                                                  {
49     // Remove extraneous info (anything after a +, anything non-numeric)
50     QStringList dots = version.replace(QRegExp("\\+.+"), "").replace(QRegExp("[^\\d\\.]"), "").split(".");
51     float base = 1.f;
52     float res = 0.f;
53     // A number like "1.2.3" will result in 1.0203
54     // NB: will fail if a point version goes over 99 or if using more than two dots. floats can only
       reliably encode 7 decimal digits.
55     foreach(const QString& dot, dots){
56         res += dot.toFloat() * base;
57         base /= 100.f;
58     }
59     return res;
60 }
```

Here is the caller graph for this function:



12.33.3.13 void KbManager::scanKeyboards ( ) [private], [slot]

Definition at line 62 of file kbmanager.cpp.

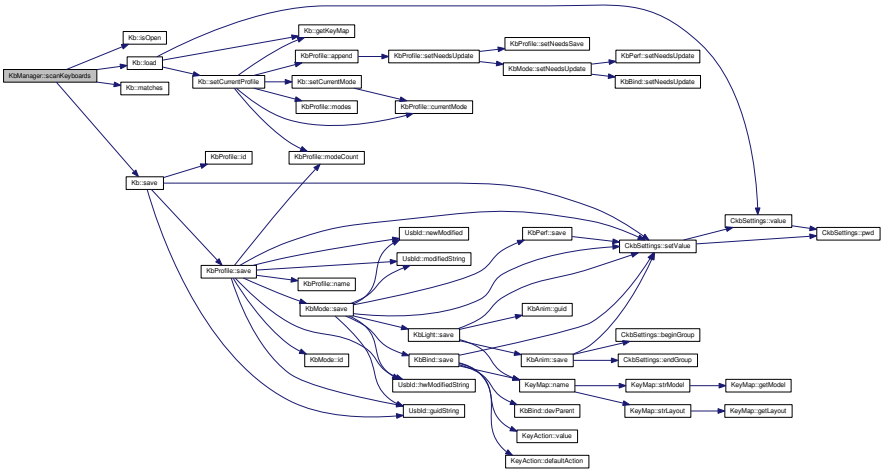
References _daemonVersion, _devices, _eventTimer, _scanTimer, DAEMON_UNAVAILABLE_STR, devpath, Kb::isOpen(), kbConnected(), kbDisconnected(), Kb::load(), Kb::matches(), Kb::save(), and versionUpdated().

Referenced by KbManager().

```
62                                {
63     QString rootdev = devpath.arg(0);
64     QFile connected(rootdev + "/connected");
65     if(!connected.open(QIODevice::ReadOnly)){
66         // No root controller - remove all keyboards
67         foreach(Kb* kb, _devices){
68             emit kbDisconnected(kb);
69             kb->save();
70             delete kb;
71         }
72         _devices.clear();
73         if(_daemonVersion != DAEMON_UNAVAILABLE_STR){
```

```
74              _daemonVersion = DAEMON_UNAVAILABLE_STR;
75              emit versionUpdated();
76          }
77          return;
78      }
79      // Check daemon version
80      QFile version(rootdev + "/version");
81      QString vString;
82      if(version.open(QIODevice::ReadOnly)){
83          vString = QString::fromUtf8(version.readLine()).trimmed();
84          version.close();
85      } else
86          vString = DAEMON_UNAVAILABLE_STR;
87      if(_daemonVersion != vString){
88          _daemonVersion = vString;
89          emit versionUpdated();
90      }
91
92      // Scan connected devices
93      QList<QStringList> lines;
94      while(1){
95          QString line = connected.readLine().trimmed();
96          if(line.isEmpty())
97              break;
98          QStringList components = line.split(" ");
99          if(components.length() < 2)             // "<path> <serial> <name>" (we're only interested in the
    first two)
100             continue;
101         lines.append(components);
102     }
103     connected.close();
104
105     // Remove any active devices not in the list
106     QMutableSetIterator<Kb*> i(_devices);
107     while(i.hasNext()){
108         Kb* kb = i.next();
109         bool matched = false;
110         foreach(const QStringList& line, lines){
111             if(kb->matches(line[0], line[1])){
112                 matched = true;
113                 break;
114             }
115         }
116         if(matched)
117             continue;
118         // Device not found, remove
119         i.remove();
120         emit kbDisconnected(kb);
121         kb->save();
122         delete kb;
123     }
124
125     // Add any new devices found in the list
126     foreach(const QStringList& line, lines){
127         bool matched = false;
128         foreach(Kb* kb, _devices){
129             if(kb->matches(line[0], line[1])){
130                 matched = true;
131                 break;
132             }
133         }
134         if(matched)
135             continue;
136         // Device not found, create new
137         Kb* kb = new Kb(this, line[0]);
138         if(!kb->isOpen()){
139             delete kb;
140             continue;
141         }
142         _devices.insert(kb);
143         // Load preferences and send signal
144         emit kbConnected(kb);
145         kb->load();
146         connect(_eventTimer, SIGNAL(timeout()), kb, SLOT(frameUpdate()));
147         connect(_scanTimer, SIGNAL(timeout()), kb, SLOT(autoSave()));
148     }
149 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.33.3.14 static QTimer* KbManager::scanTimer ( ) [inline], [static]

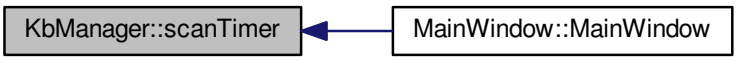
Definition at line 43 of file kbmanager.h.

References _kbManager, and _scanTimer.

Referenced by MainWindow::MainWindow().

```
43 { return _kbManager ? _kbManager->_scanTimer : 0; }
```

Here is the caller graph for this function:



12.33.3.15 void KbManager::stop ( ) [static]

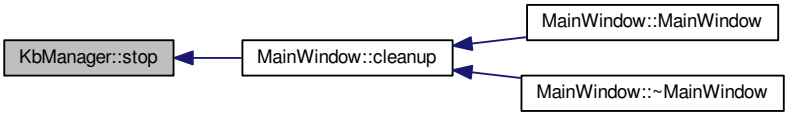
Definition at line 19 of file kbmanager.cpp.

References _kbManager.

Referenced by MainWindow::cleanup().

```
19                      {
20     if(!_kbManager)
21         return;
22     delete _kbManager;
23     _kbManager = 0;
24 }
```

Here is the caller graph for this function:



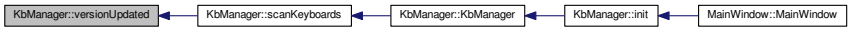
12.33.3.16 void KbManager::versionUpdated ( ) [signal]

Definition at line 188 of file moc_kbmanager.cpp.

Referenced by scanKeyboards().

```
189 {
190     QMetaObject::activate(this, &staticMetaObject, 2, Q_NULLPTR);
191 }
```

Here is the caller graph for this function:



12.33.4 Field Documentation

12.33.4.1 QString KbManager::_daemonVersion = "<unavailable>" [static], [private]

Definition at line 59 of file kbmanager.h.

Referenced by ckbDaemonVersion(), ckbDaemonVersionF(), and scanKeyboards().

12.33.4.2 QSet<Kb*> KbManager::_devices [private]

Definition at line 63 of file kbmanager.h.

Referenced by devices(), and scanKeyboards().

12.33.4.3 QTimer* KbManager::_eventTimer [private]

Definition at line 64 of file kbmanager.h.

Referenced by eventTimer(), KbManager(), and scanKeyboards().

12.33.4.4 QString KbManager::_guiVersion [static], [private]

Definition at line 59 of file kbmanager.h.

Referenced by ckbGuiVersion(), ckbGuiVersionF(), and init().

12.33.4.5 KbManager * KbManager::_kbManager = 0 [static], [private]

Definition at line 58 of file kbmanager.h.

Referenced by devices(), eventTimer(), init(), kbManager(), scanTimer(), and stop().

12.33.4.6 QTimer * KbManager::_scanTimer [private]

Definition at line 64 of file kbmanager.h.

Referenced by KbManager(), scanKeyboards(), and scanTimer().

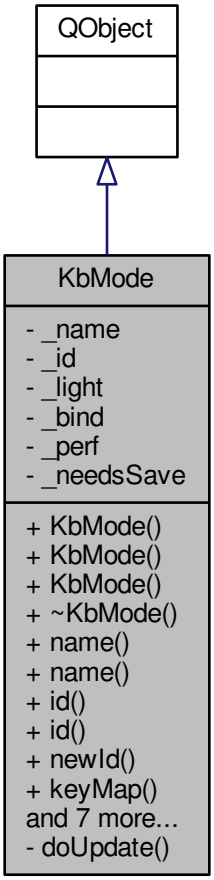
The documentation for this class was generated from the following files:

- src/ckb/kbmanager.h
- src/ckb/kbmanager.cpp
- src/ckb/release/moc_kbmanager.cpp

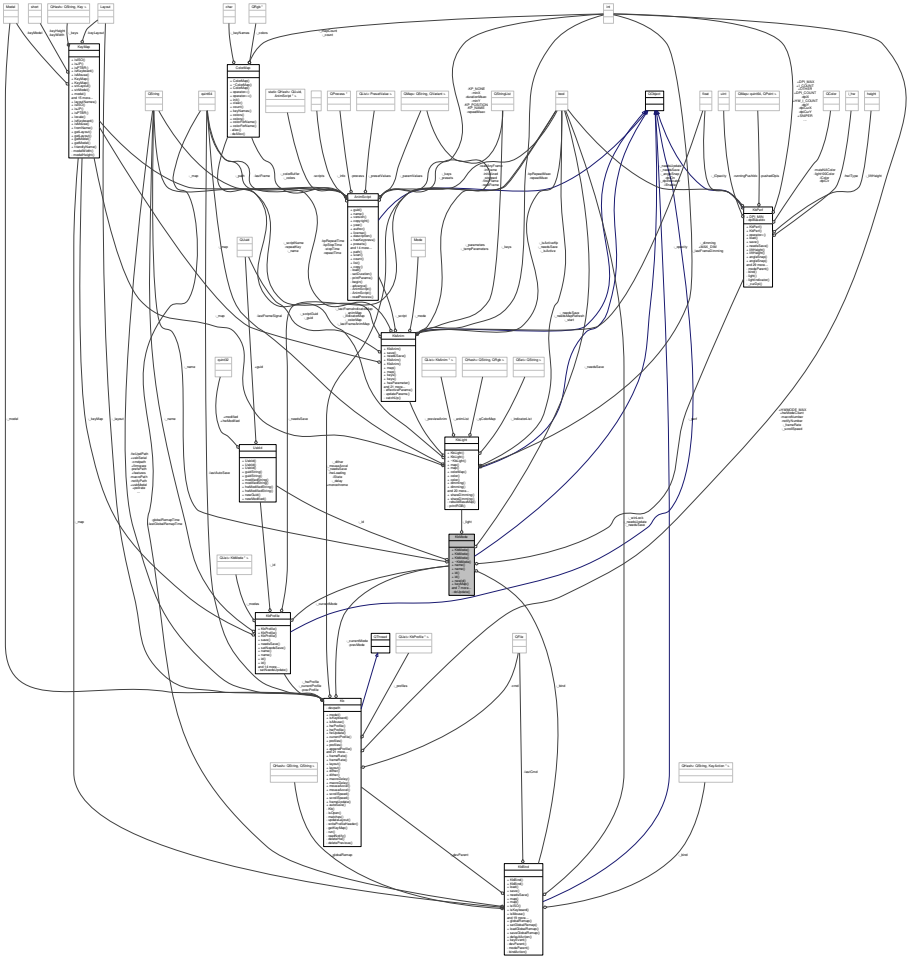
12.34 KbMode Class Reference

`#include <src/ckb/kbmode.h>`

Inheritance diagram for KbMode:



Collaboration diagram for KbMode:



Signals

- void updated ()

Public Member Functions

- KbMode (Kb *parent, const KeyMap &keyMap, const QString &guid="", const QString &modified="")
- KbMode (Kb *parent, const KeyMap &keyMap, CkbSettings &settings)
- KbMode (Kb *parent, const KeyMap &keyMap, const KbMode &other)

  KbMode::KbMode.
- ~KbMode ()

  KbMode::~KbMode Destructor may be used for Debugging (issue #38 with SIGSEGV). Insert qDebug-statement.
- const QString & name () const
- void name (const QString &newName)
- UsbId & id ()
- void id (const UsbId &newId)
- void newId ()
- void keyMap (const KeyMap &keyMap)
- KbLight * light ()
- KbBind * bind ()
- KbPerf * perf ()

- void save (CkbSettings &settings)
- bool needsSave () const
- void setNeedsSave ()
- void setNeedsUpdate ()

Private Slots

- void doUpdate ()

Private Attributes

- QString _name
- UsbId _id
- KbLight * _light
- KbBind * _bind
- KbPerf * _perf
- bool _needsSave

12.34.1 Detailed Description

Definition at line 36 of file kbmode.h.

12.34.2 Constructor & Destructor Documentation

12.34.2.1 KbMode::KbMode (Kb * *parent*, const KeyMap & *keyMap*, const QString & *guid* = " ", const QString & *modified* = " ")

Definition at line 4 of file kbmode.cpp.

References _id, _light, doUpdate(), UsbId::guid, and updated().

```
4                                                                                                    :
5     QObject(parent),
6     _name("Unnamed"), _id(guid, modified),
7     _light(new KbLight(this, keyMap)), _bind(new KbBind(this, parent, keyMap)),
      _perf(new KbPerf(this)),
8     _needsSave(true)
9 {
10     connect(_light, SIGNAL(updated()), this, SLOT(doUpdate()));
11     if(_id.guid.isNull())
12         _id.guid = QUuid::createUuid();
13 }
```

Here is the call graph for this function:

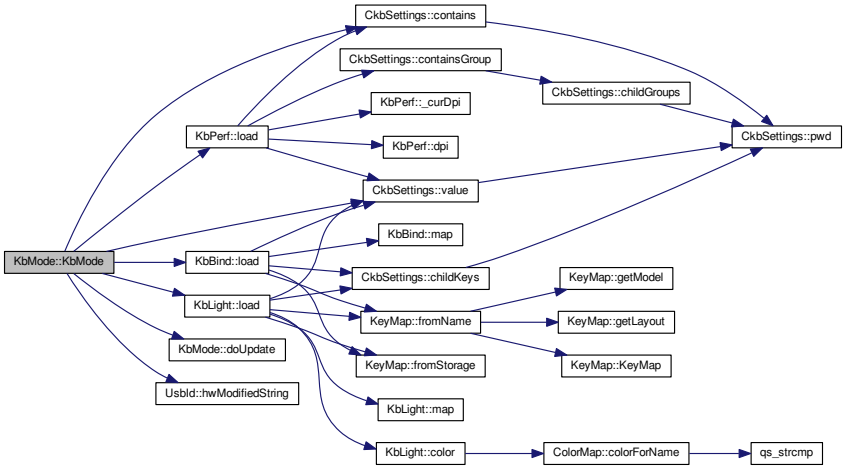
12.34.2.2 KbMode::KbMode (Kb * *parent*, const KeyMap & *keyMap*, CkbSettings & *settings*)

Definition at line 30 of file kbmode.cpp.

References _bind, _id, _light, _name, _perf, CkbSettings::contains(), doUpdate(), UsbId::guid, UsbId::hwModified, UsbId::hwModifiedString(), KbPerf::load(), KbBind::load(), KbLight::load(), UsbId::modified, updated(), and CkbSettings::value().

```
30                                                                    :
31     QObject(parent),
32     _name(settings.value("Name").toString().trimmed()),
33     _id(settings.value("GUID").toString().trimmed(), settings.value("Modified").toString().
       trimmed()),
34     _light(new KbLight(this, keyMap)), _bind(new KbBind(this, parent, keyMap)),
       _perf(new KbPerf(this)),
35     _needsSave(false)
36 {
37     if(settings.contains("HwModified"))
38         _id.hwModifiedString(settings.value("HwModified").toString());
39     else
40         _id.hwModified = _id.modified;
41
42     connect(_light, SIGNAL(updated()), this, SLOT(doUpdate()));
43     if(_id.guid.isNull())
44         _id.guid = QUuid::createUuid();
45     if(_name == "")
46         _name = "Unnamed";
47     _light->load(settings);
48     _bind->load(settings);
49     _perf->load(settings);
50 }
```

Here is the call graph for this function:



12.34.2.3 KbMode::KbMode (Kb * *parent*, const KeyMap & *keyMap*, const KbMode & *other*)

Parameters

parent	Kb as parent (append to the Keyboard list
keyMap	Map to copy from

other	Mode to copy from Constructor to copy an existing Keyboard-Mode KbMode &other

Definition at line 21 of file kbmode.cpp.

References _light, doUpdate(), and updated().

```
21                                                                    :
22     QObject(parent),
23     _name(other._name), _id(other._id),
24     _light(new KbLight(this, keyMap, *other._light)), _bind(new
       KbBind(this, parent, keyMap, *other._bind)), _perf(new KbPerf(this, *other.
       _perf)),
25     _needsSave(true)
26 {
27     connect(_light, SIGNAL(updated()), this, SLOT(doUpdate()));
28 }
```

Here is the call graph for this function:

12.34.2.4 KbMode::~KbMode ( )

Definition at line 89 of file kbmode.cpp.

```
89                  {
90 }
```

12.34.3 Member Function Documentation

12.34.3.1 KbBind* KbMode::bind ( ) [inline]

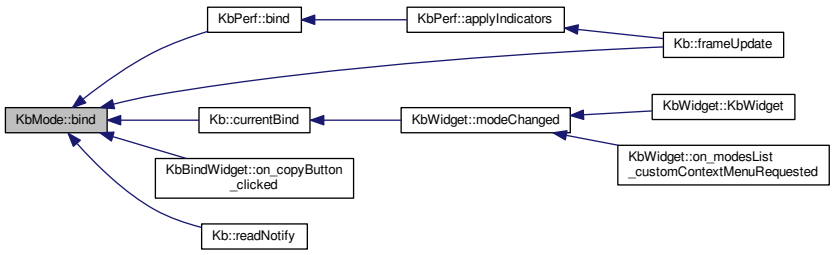
Definition at line 61 of file kbmode.h.

References _bind.

Referenced by KbPerf::bind(), Kb::currentBind(), Kb::frameUpdate(), KbBindWidget::on_copyButton_clicked(), and Kb::readNotify().

```
61 { return _bind; }
```

Here is the caller graph for this function:



12.34.3.2 void KbMode::doUpdate ( ) [private], [slot]

Definition at line 82 of file kbmode.cpp.

References updated().

Referenced by KbMode().

```
82                      {
83     emit updated();
84 }
```

Here is the caller graph for this function:

12.34.3.3 UsbId& KbMode::id ( ) [inline]

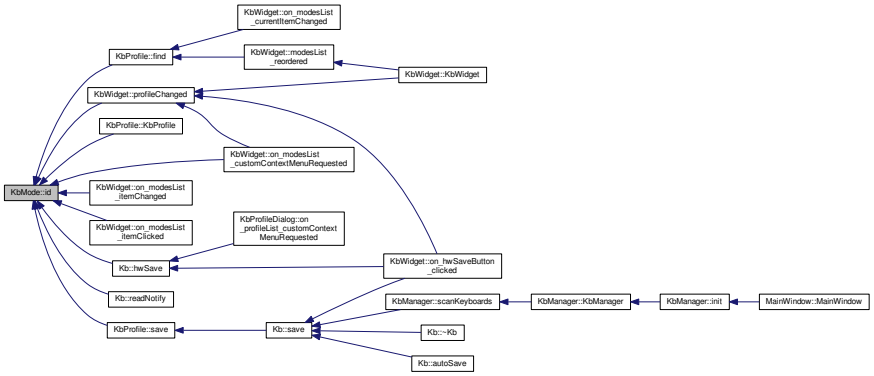
Definition at line 52 of file kbmode.h.

References _id.

Referenced by KbProfile::find(), Kb::hwSave(), KbProfile::KbProfile(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::on_modesList_itemChanged(), KbWidget::on_modesList_itemClicked(), KbWidget::profileChanged(), Kb::readNotify(), and KbProfile::save().

```
52 { return _id; }
```

Here is the caller graph for this function:



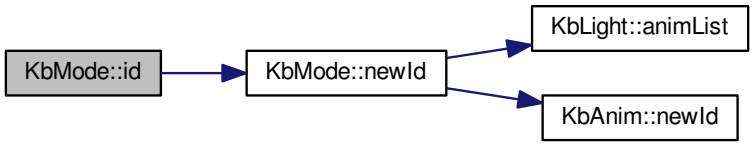
12.34.3.4 void KbMode::id (const UsbId & *newId*) [inline]

Definition at line 53 of file kbmode.h.

References _id, _needsSave, and newId().

```
53 { _needsSave = true; _id = newId; }
```

Here is the call graph for this function:



12.34.3.5 void KbMode::keyMap (const KeyMap & *keyMap*)

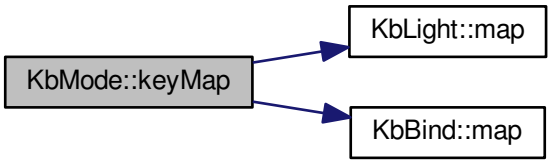
Definition at line 60 of file kbmode.cpp.

References _bind, _light, _needsSave, KbLight::map(), and KbBind::map().

Referenced by KbProfile::keyMap().

```
60                                      {
61     _needsSave = true;
62     _light->map(keyMap);
63     _bind->map(keyMap);
64 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.34.3.6 KbLight* KbMode::light ( ) [inline]

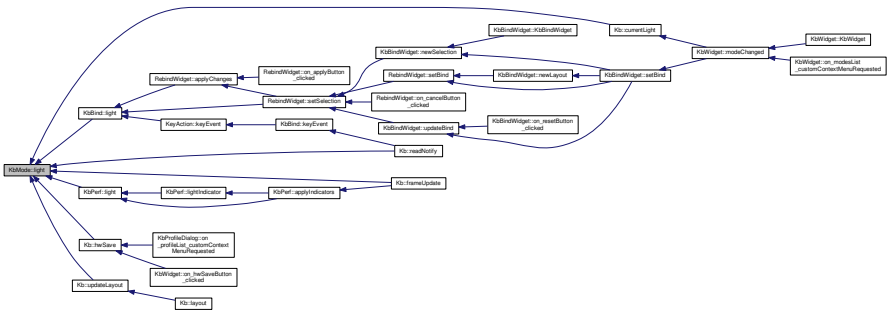
Definition at line 60 of file kbmode.h.

References _light.

Referenced by Kb::currentLight(), Kb::frameUpdate(), Kb::hwSave(), KbBind::light(), KbPerf::light(), Kb::readNotify(), and Kb::updateLayout().

```
60 { return _light; }
```

Here is the caller graph for this function:



12.34.3.7 const QString& KbMode::name ( ) const [inline]

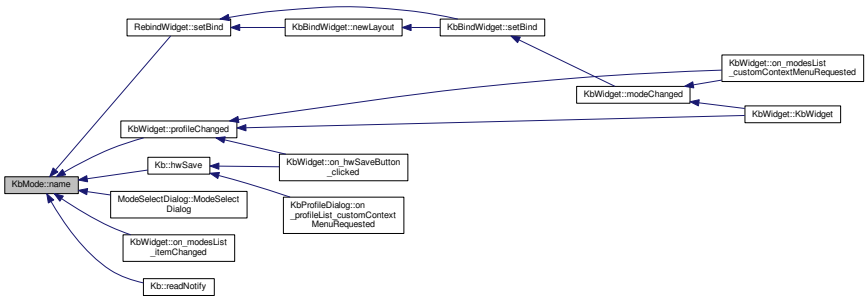
Definition at line 50 of file kbmode.h.

References _name.

Referenced by Kb::hwSave(), ModeSelectDialog::ModeSelectDialog(), KbWidget::on_modesList_itemChanged(), KbWidget::profileChanged(), Kb::readNotify(), and RebindWidget::setBind().

```
50 { return _name; }
```

Here is the caller graph for this function:



12.34.3.8 void KbMode::name (const QString & *newName*) [inline]

Definition at line 51 of file kbmode.h.

References _name, and _needsSave.

```
51 { _needsSave = true; _name = newName.trimmed(); if(_name == "")
      _name = "Unnamed"; }
```

12.34.3.9 bool KbMode::needsSave ( ) const

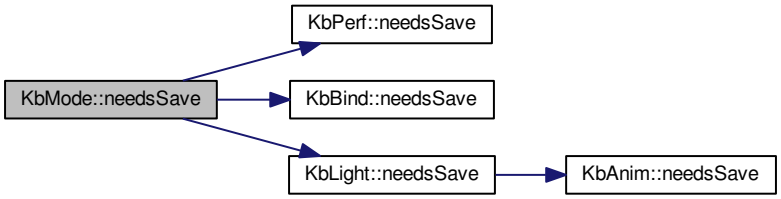
Definition at line 78 of file kbmode.cpp.

References _bind, _light, _needsSave, _perf, KbPerf::needsSave(), KbBind::needsSave(), and KbLight::needsSave().

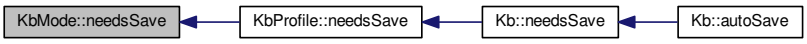
Referenced by KbProfile::needsSave().

```
78                          {
79     return _needsSave || _light->needsSave() || _bind->
      needsSave() || _perf->needsSave();
80 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.34.3.10 void KbMode::newId ( )

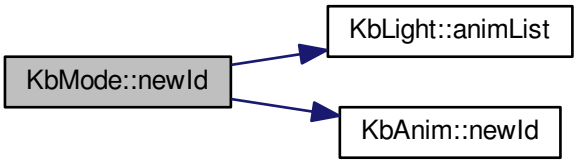
Definition at line 52 of file kbmode.cpp.

References _id, _light, _needsSave, anim, KbLight::animList(), and KbAnim::newId().

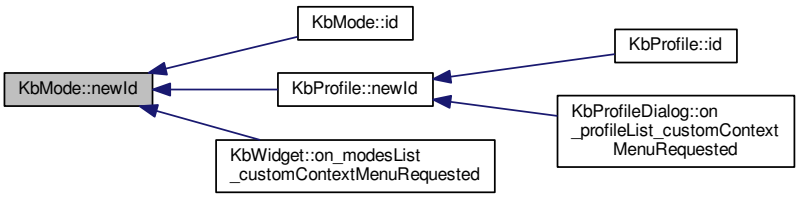
Referenced by id(), KbProfile::newId(), and KbWidget::on_modesList_customContextMenuRequested().

```
52                  {
53      _needsSave = true;
54      _id = UsbId();
55      // Create new IDs for animations
56      foreach(KbAnim* anim, _light->animList())
57          anim->newId();
58  }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.34.3.11 KbPerf* KbMode::perf ( ) [inline]

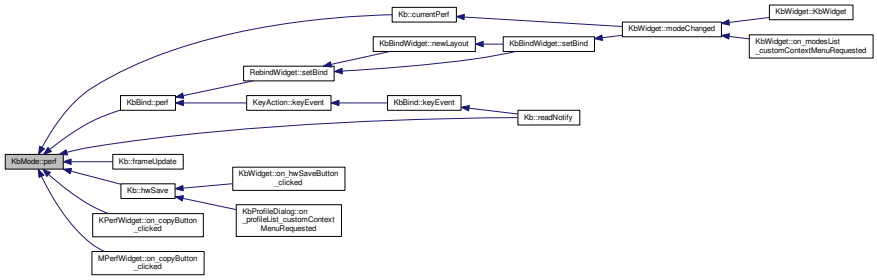
Definition at line 62 of file kbmode.h.

References _perf.

Referenced by Kb::currentPerf(), Kb::frameUpdate(), Kb::hwSave(), KPerfWidget::on_copyButton_clicked(), MPerfWidget::on_copyButton_clicked(), KbBind::perf(), and Kb::readNotify().

```
62 { return _perf; }
```

Here is the caller graph for this function:



12.34.3.12 void KbMode::save (CkbSettings & *settings*)

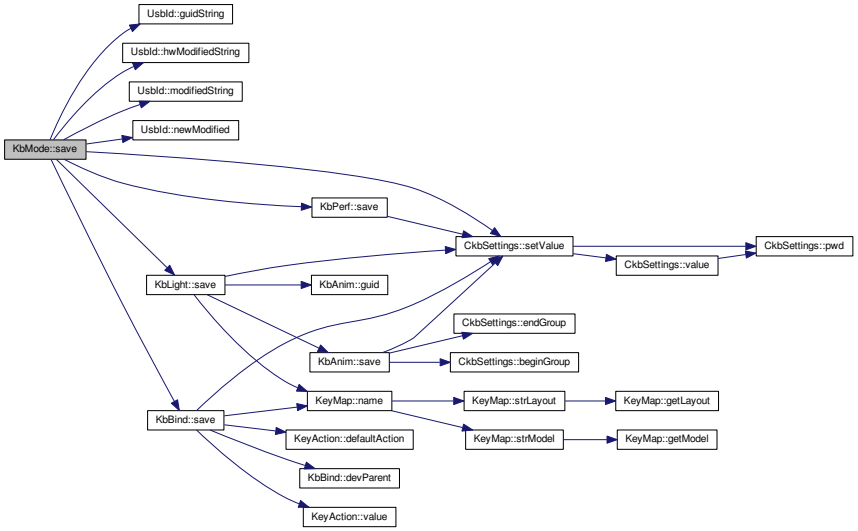
Definition at line 66 of file kbmode.cpp.

References _bind, _id, _light, _name, _needsSave, _perf, UsbId::guidString(), UsbId::hwModifiedString(), UsbId::modifiedString(), UsbId::newModified(), KbPerf::save(), KbBind::save(), KbLight::save(), and CkbSettings::setValue().

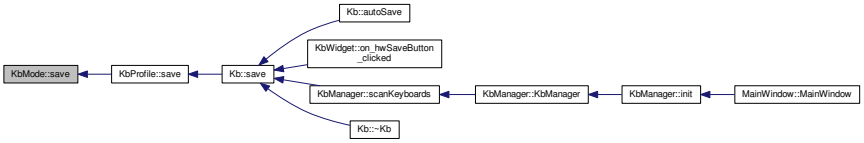
Referenced by KbProfile::save().

```
66                                    {
67     _needsSave = false;
68     _id.newModified();
69     settings.setValue("GUID", _id.guidString());
70     settings.setValue("Modified", _id.modifiedString());
71     settings.setValue("HwModified", _id.hwModifiedString());
72     settings.setValue("Name", _name);
73     _light->save(settings);
74     _bind->save(settings);
75     _perf->save(settings);
76 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.34.3.13 void KbMode::setNeedsSave ( ) [inline]

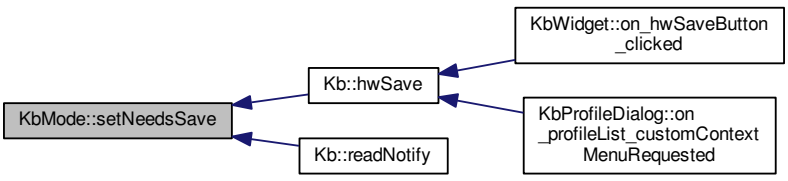
Definition at line 67 of file kbmode.h.

References _needsSave.

Referenced by Kb::hwSave(), and Kb::readNotify().

```
67 { _needsSave = true; }
```

Here is the caller graph for this function:



12.34.3.14 void KbMode::setNeedsUpdate ( ) [inline]

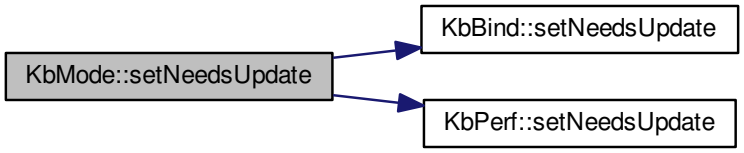
Definition at line 68 of file kbmode.h.

References _bind, _perf, KbBind::setNeedsUpdate(), and KbPerf::setNeedsUpdate().

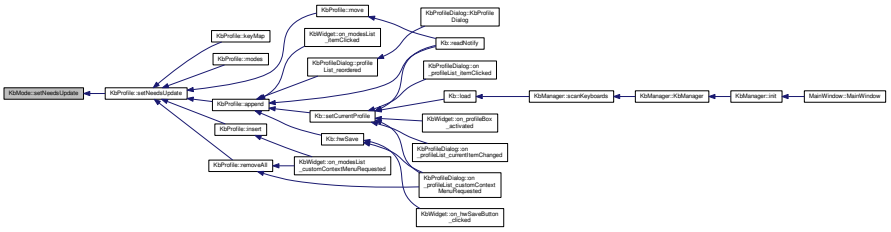
Referenced by KbProfile::setNeedsUpdate().

```
68 { _bind->setNeedsUpdate(); _perf->setNeedsUpdate(); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



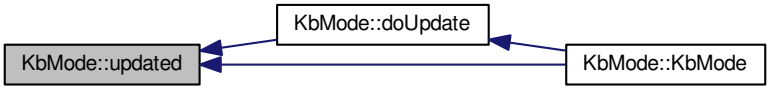
12.34.3.15 void KbMode::updated ( ) [signal]

Definition at line 131 of file moc_kbmode.cpp.

Referenced by doUpdate(), and KbMode().

```
132 {
133     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
134 }
```

Here is the caller graph for this function:



12.34.4 Field Documentation

12.34.4.1 KbBind* KbMode::_bind [private]

Definition at line 78 of file kbmode.h.

Referenced by bind(), KbMode(), keyMap(), needsSave(), save(), and setNeedsUpdate().

12.34.4.2 UsbId KbMode::_id [private]

Definition at line 75 of file kbmode.h.

Referenced by id(), KbMode(), newId(), and save().

12.34.4.3 KbLight* KbMode::_light [private]

Definition at line 77 of file kbmode.h.

Referenced by KbMode(), keyMap(), light(), needsSave(), newId(), and save().

12.34.4.4 QString KbMode::_name [private]

Definition at line 74 of file kbmode.h.

Referenced by KbMode(), name(), and save().

12.34.4.5 bool KbMode::_needsSave [private]

Definition at line 81 of file kbmode.h.

Referenced by id(), keyMap(), name(), needsSave(), newId(), save(), and setNeedsSave().

12.34.4.6 KbPerf* KbMode::_perf [private]

Definition at line 79 of file kbmode.h.

Referenced by KbMode(), needsSave(), perf(), save(), and setNeedsUpdate().

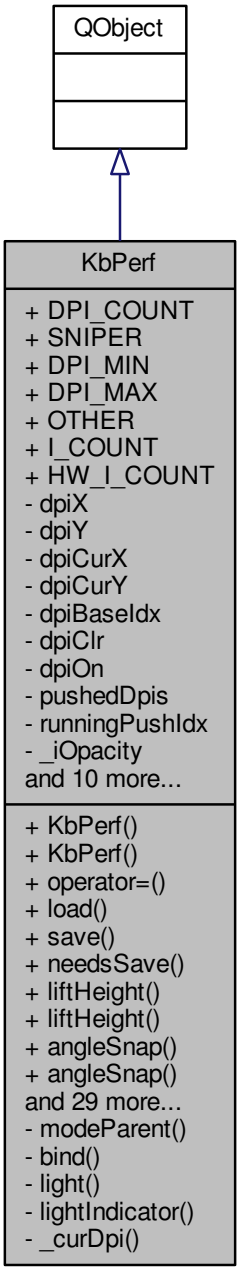
The documentation for this class was generated from the following files:

- src/ckb/kbmode.h
- src/ckb/kbmode.cpp
- src/ckb/release/moc_kbmode.cpp

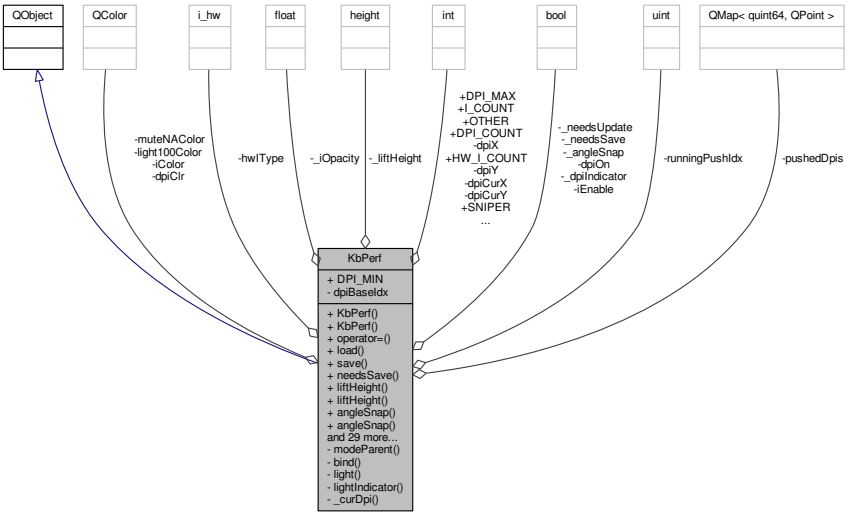
12.35 KbPerf Class Reference

`#include <src/ckb/kbperf.h>`

Inheritance diagram for KbPerf:



Collaboration diagram for KbPerf:



Public Types

- enum height {
  LOW = 1, LOWMED, MEDIUM, MEDHIGH,
  HIGH }
- enum indicator {
  NUM, CAPS, SCROLL, HW_IMAX = SCROLL,
  MODE, MACRO, LIGHT, LOCK,
  MUTE }
- enum i_hw { NONE = -1, NORMAL, ON, OFF }

Signals

- void didLoad ()
- void settingsUpdated ()

Public Member Functions

- KbPerf (KbMode *parent)
- KbPerf (KbMode *parent, const KbPerf &other)
- const KbPerf & operator= (const KbPerf &rhs)
- void load (CkbSettings &settings)
- void save (CkbSettings &settings)
- bool needsSave () const
- height liftHeight () const
- void liftHeight (height newHeight)
- bool angleSnap () const
- void angleSnap (bool newAngleSnap)
- QPoint dpi (int index) const
- void dpi (int index, const QPoint &newValue)
- QPoint sniperDpi () const
- void sniperDpi (const QPoint &newValue)

- void dpi (int index, int newValue)
- void sniperDpi (int newValue)
- int baseDpiIdx () const
- void baseDpiIdx (int newIdx)
- void dpiUp ()
- void dpiDown ()
- void dpiCycleUp ()
- void dpiCycleDown ()
- bool dpiEnabled (int index) const
- void dpiEnabled (int index, bool newEnabled)
- quint64 pushDpi (const QPoint &newDpi)
- quint64 pushDpi (int newDpi)
- quint64 pushSniper ()
- void popDpi (quint64 pushIdx)
- float iOpacity () const
- void iOpacity (float newIOpacity)
- bool dpiIndicator () const
- void dpiIndicator (bool newDpiIndicator)
- QColor dpiColor (int index) const
- void dpiColor (int index, const QColor &newColor)
- void getIndicator (indicator index, QColor &color1, QColor &color2, QColor &color3, bool &software_enable, i_hw &hardware_enable)
- void setIndicator (indicator index, const QColor &color1, const QColor &color2, const QColor &color3=QColor(), bool software_enable=true, i_hw hardware_enable=NORMAL)
- void update (QFile &cmd, int notifyNumber, bool force, bool saveCustomDpi)
- void setNeedsUpdate ()
- void applyIndicators (int modeIndex, const bool indicatorState[HW_I_COUNT])

Static Public Attributes

- static const int DPI_COUNT = 6
- static const int SNIPER = 0
- static const int DPI_MIN = 100
- static const int DPI_MAX = 12000
- static const int OTHER = DPI_COUNT
- static const int I_COUNT = (int)MUTE + 1
- static const int HW_I_COUNT = (int)HW_IMAX + 1

Private Member Functions

- KbMode * modeParent () const
- KbBind * bind () const
- KbLight * light () const
- void lightIndicator (const char *name, QRgb rgba)
- void _curDpi (const QPoint &newDpi)

Private Attributes

- int dpiX [DPI_COUNT]
- int dpiY [DPI_COUNT]
- int dpiCurX
- int dpiCurY
- int dpiBaseIdx
- QColor dpiClr [DPI_COUNT+1]
- bool dpiOn [DPI_COUNT]
- QMap< quint64, QPoint > pushedDpis
- uint runningPushIdx
- float _iOpacity
- QColor iColor [I_COUNT][2]
- QColor light100Color
- QColor muteNAColor
- bool iEnable [I_COUNT]
- i_hw hwIType [HW_I_COUNT]
- bool _dpiIndicator
- height _liftHeight
- bool _angleSnap
- bool _needsUpdate
- bool _needsSave

12.35.1 Detailed Description

Definition at line 15 of file kbperf.h.

12.35.2 Member Enumeration Documentation

12.35.2.1 enum KbPerf::height

Enumerator

LOW

LOWMED

MEDIUM

MEDHIGH

HIGH

Definition at line 31 of file kbperf.h.

```
31                  {
32          LOW = 1,
33          LOWMED,
34          MEDIUM,
35          MEDHIGH,
36          HIGH
37      };
```

12.35.2.2 enum KbPerf::i_hw

Enumerator

NONE

NORMAL

ON

OFF

Definition at line 101 of file kbperf.h.

```
101                 {
102         NONE = -1,  // For non-hardware indicators
103         NORMAL,
104         ON,
105         OFF
106     };
```

12.35.2.3 enum KbPerf::indicator

Enumerator

NUM

CAPS

SCROLL

HW_IMAX

MODE

MACRO

LIGHT

LOCK

MUTE

Definition at line 88 of file kbperf.h.

```
88                      {
89         // Hardware
90         NUM,
91         CAPS,
92         SCROLL, HW_IMAX = SCROLL,
93         // Software
94         MODE,
95         MACRO,
96         LIGHT,
97         LOCK,
98         MUTE
99     };
```

12.35.3 Constructor & Destructor Documentation

12.35.3.1 KbPerf::KbPerf (KbMode * *parent*) [explicit]

Definition at line 13 of file kbperf.cpp.

References CAPS, DPI_COUNT, dpiBaseIdx, dpiClr, dpiCurX, dpiCurY, dpiOn, dpiX, dpiY, HW_I_COUNT, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, LOCK, MACRO, MODE, MUTE, muteNAColor, NORMAL, NUM, and SCROLL.

```
13                                :
14     QObject(parent), runningPushIdx(1),
15     _iOpacity(1.f), _dpiIndicator(true), _liftHeight(
       MEDIUM), _angleSnap(false),
16     _needsUpdate(true), _needsSave(true) {
17     // Default DPI settings
18     dpiX[0] = dpiY[0] = 400;
19     dpiX[1] = dpiY[1] = 450;
20     dpiX[2] = dpiY[2] = 800;
21     dpiX[3] = dpiY[3] = 1500;
22     dpiX[4] = dpiY[4] = 3000;
23     dpiX[5] = dpiY[5] = 6000;
24     dpiClr[0] = QColor(255, 0, 0);
25     dpiClr[1] = QColor(255, 192, 0);
26     dpiClr[2] = QColor(255, 255, 0);
27     dpiClr[3] = QColor(0, 255, 0);
28     dpiClr[4] = QColor(0, 255, 255);
29     dpiClr[5] = QColor(255, 255, 255);
30     dpiClr[6] = QColor(192, 192, 192);
31     for(int i = 0; i < DPI_COUNT; i++)
32         dpiOn[i] = true;
33     dpiBaseIdx = 3;
34     dpiCurX = dpiX[dpiBaseIdx];
35     dpiCurY = dpiY[dpiBaseIdx];
36     // Default indicators
37     iColor[NUM][0] = iColor[CAPS][0] = iColor[SCROLL][0] = QColor(0, 255, 0)
       ;   // Lock lights: on = green, off = black
38     iColor[NUM][1] = iColor[CAPS][1] = iColor[SCROLL][1] = QColor(0, 0, 0);
39     iColor[MACRO][0] = QColor(255, 0, 0);                                    // Macro: on =
       red, off = black
40     iColor[LOCK][0] = QColor(255, 255, 255);                                 // Win lock: on =
       white, off = black
41     iColor[MODE][0] = iColor[MUTE][0] = QColor(255, 255, 255, 0);            //
       Mode, mute: on = transparent, off = black
42     iColor[MODE][1] = iColor[MACRO][1] = iColor[LOCK][1] =
      iColor[MUTE][1] = muteNAColor = QColor(0, 0, 0);
43     iColor[LIGHT][0] = QColor(255, 0, 0);       // Brightness: red, yellow, white
44     iColor[LIGHT][1] = QColor(255, 255, 0);
45     light100Color = QColor(255, 255, 255);
46     for(int i = 0; i < HW_I_COUNT; i++){
47         // Set all lock lights to HW mode
48         iEnable[i] = false;
49         hwIType[i] = NORMAL;
50     }
51     for(int i = HW_I_COUNT; i < I_COUNT; i++){
52         // Turn all other indicators on
53         iEnable[i] = true;
54     }
55 }
```

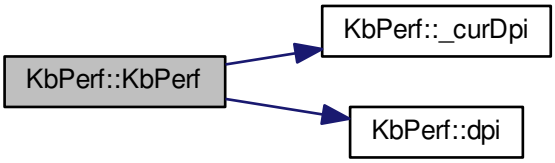
12.35.3.2 KbPerf::KbPerf (KbMode * *parent,* const KbPerf & *other*)

Definition at line 57 of file kbperf.cpp.

References _curDpi(), dpi(), DPI_COUNT, dpiBaseIdx, dpiClr, dpiOn, dpiX, dpiY, hwIType, I_COUNT, iColor, and iEnable.

```
57                                                :
58     QObject(parent), dpiCurX(other.dpiCurX), dpiCurY(other.
       dpiCurY), dpiBaseIdx(other.dpiBaseIdx),
       runningPushIdx(1),
59     _iOpacity(other._iOpacity), light100Color(other.
       light100Color), muteNAColor(other.muteNAColor),
       _dpiIndicator(other._dpiIndicator),
60     _liftHeight(other._liftHeight), _angleSnap(other.
       _angleSnap),
61     _needsUpdate(true), _needsSave(true) {
62     memcpy(dpiX, other.dpiX, sizeof(dpiX));
63     memcpy(dpiY, other.dpiY, sizeof(dpiY));
64     for(int i = 0; i < DPI_COUNT + 1; i++)
65         dpiClr[i] = other.dpiClr[i];
66     memcpy(dpiOn, other.dpiOn, sizeof(dpiOn));
67     for(int i = 0; i < I_COUNT; i++){
68         iColor[i][0] = other.iColor[i][0];
69         iColor[i][1] = other.iColor[i][1];
70     }
71     memcpy(iEnable, other.iEnable, sizeof(iEnable));
72     memcpy(hwIType, other.hwIType, sizeof(hwIType));
73     // Don't copy pushed DPI states. If the other mode has any, restore the original DPI
74     _curDpi(dpi(dpiBaseIdx));
75 }
```

Here is the call graph for this function:



12.35.4 Member Function Documentation

12.35.4.1 void KbPerf::_curDpi (const QPoint & *newDpi*) [private]

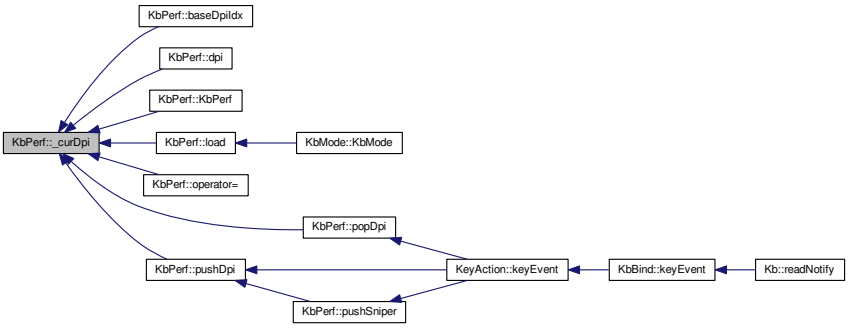
Definition at line 259 of file kbperf.cpp.

References _needsSave, _needsUpdate, dpiCurX, and dpiCurY.

Referenced by baseDpiIdx(), dpi(), KbPerf(), load(), operator=(), popDpi(), and pushDpi().

```
259                                         {
260     dpiCurX = newDpi.x();
261     dpiCurY = newDpi.y();
262     _needsUpdate = _needsSave = true;
263 }
```

Here is the caller graph for this function:



12.35.4.2 bool KbPerf::angleSnap ( ) const [inline]

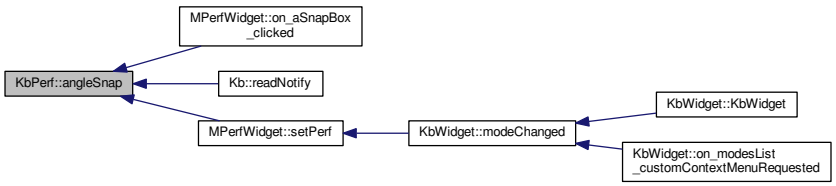
Definition at line 42 of file kbperf.h.

References _angleSnap.

Referenced by MPerfWidget::on_aSnapBox_clicked(), Kb::readNotify(), and MPerfWidget::setPerf().

```
42 { return _angleSnap; }
```

Here is the caller graph for this function:



12.35.4.3 void KbPerf::angleSnap (bool *newAngleSnap*)

Definition at line 380 of file kbperf.cpp.

References _angleSnap, _needsSave, and _needsUpdate.

```
380                                         {
381     _angleSnap = newAngleSnap;
382     _needsUpdate = _needsSave = true;
383 }
```

12.35.4.4 void KbPerf::applyIndicators (int *modeIndex,* const bool *indicatorState[HW_I_COUNT]*)

Definition at line 439 of file kbperf.cpp.

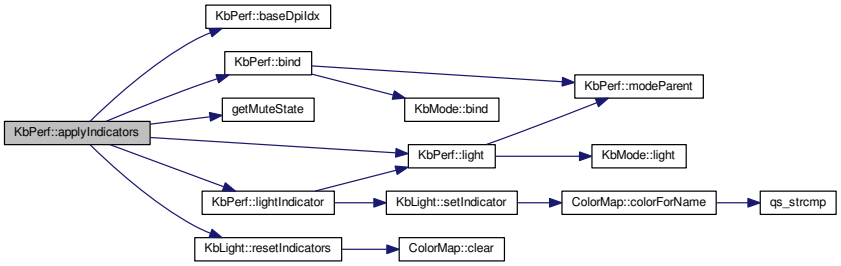
References _dpiIndicator, _iOpacity, baseDpiIdx(), bind(), CAPS, dpiClr, getMuteState(), Kb::HWMODE_MAX, iColor, iEnable, LIGHT, light(), light100Color, lightIndicator(), LOCK, MACRO, MODE, MUTE, MUTED, muteNAColor, NUM, OTHER, KbLight::resetIndicators(), SCROLL, and UNMUTED.

Referenced by Kb::frameUpdate().

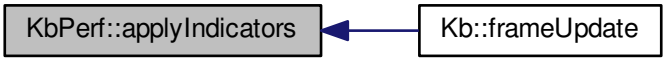
```
439                                                                         {
440     light()->resetIndicators();
441     if(_iOpacity <= 0.f)
442         return;
443     if(_dpiIndicator){
444         // Set DPI indicator according to index
445         int index = baseDpiIdx();
446         if(index == -1 || index > OTHER)
447             index = OTHER;
448         lightIndicator("dpi", dpiClr[index].rgba());
449     }
450     // KB indicators
451     if(iEnable[MODE]){
452         for(int i = 0; i < Kb::HWMODE_MAX; i++){
453             char name[4];
454             snprintf(name, sizeof(name), "m%d", i + 1);
455             if(modeIndex == i)
456                 lightIndicator(name, iColor[MODE][0].rgba());
457             else
458                 lightIndicator(name, iColor[MODE][1].rgba());
459         }
460     }
461     if(iEnable[MACRO])
462         lightIndicator("mr", iColor[MUTE][1].rgba());
463     if(iEnable[LIGHT]){
464         switch(light()->dimming()){
465         case 0: // 100%
466             lightIndicator("light", light100Color.rgba());
467             break;
468         case 1: // 67%
469             lightIndicator("light", iColor[LIGHT][1].rgba());
470             break;
471         case 2: // 33%
472         case 3: // light off
473             lightIndicator("light", iColor[LIGHT][0].rgba());
474             break;
```

```
475         }
476     }
477     if(iEnable[LOCK]){
478         if(bind()->winLock())
479             lightIndicator("lock", iColor[LOCK][0].rgba());
480         else
481             lightIndicator("lock", iColor[LOCK][1].rgba());
482     }
483     if(iEnable[MUTE]){
484         switch(getMuteState()){
485         case MUTED:
486             lightIndicator("mute", iColor[MUTE][0].rgba());
487             break;
488         case UNMUTED:
489             lightIndicator("mute", iColor[MUTE][1].rgba());
490             break;
491         default:
492             lightIndicator("mute", muteNAColor.rgba());
493             break;
494         }
495     }
496     // Lock lights
497     if(iEnable[NUM])
498         lightIndicator("numlock", indicatorState[0] ? iColor[NUM][0].rgba() :
    iColor[NUM][1].rgba());
499     if(iEnable[CAPS])
500         lightIndicator("caps", indicatorState[1] ? iColor[CAPS][0].rgba() :
    iColor[CAPS][1].rgba());
501     if(iEnable[SCROLL])
502         lightIndicator("scroll", indicatorState[2] ? iColor[SCROLL][0].rgba() :
    iColor[SCROLL][1].rgba());
503 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.5 int KbPerf::baseDpiIdx ( ) const [inline]

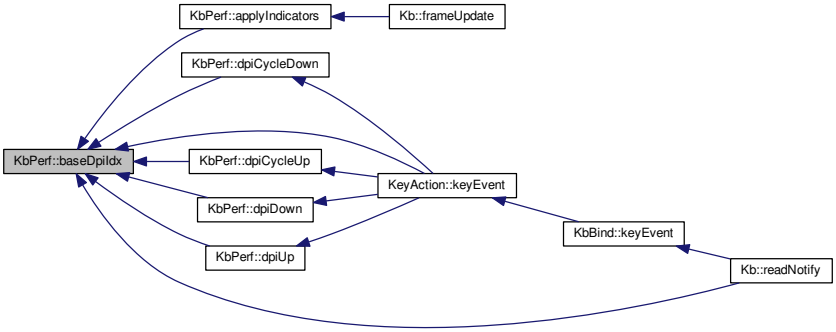
Definition at line 61 of file kbperf.h.

References dpiBaseIdx.

Referenced by applyIndicators(), dpiCycleDown(), dpiCycleUp(), dpiDown(), dpiUp(), KeyAction::keyEvent(), and Kb::readNotify().

```
61 { return dpiBaseIdx; }
```

Here is the caller graph for this function:



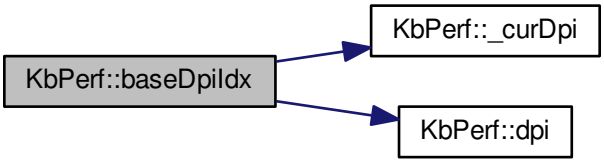
12.35.4.6 void KbPerf::baseDpiIdx (int *newIdx*)

Definition at line 265 of file kbperf.cpp.

References _curDpi(), _needsSave, _needsUpdate, dpi(), dpiBaseIdx, and pushedDpis.

```
265                                 {
266     if (pushedDpis.isEmpty() && dpiBaseIdx == newIdx)
267         return;
268     pushedDpis.clear();
269     dpiBaseIdx = newIdx;
270     _curDpi(dpi(dpiBaseIdx));
271     _needsUpdate = _needsSave = true;
272 }
```

Here is the call graph for this function:



12.35.4.7 KbBind * KbPerf::bind ( ) const [private]

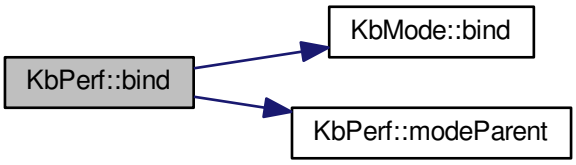
Definition at line 98 of file kbperf.cpp.

References KbMode::bind(), and modeParent().

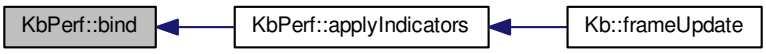
Referenced by applyIndicators().

```
98                              {
99      return modeParent()->bind();
100 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.8 void KbPerf::didLoad ( ) [signal]

Definition at line 134 of file moc_kbperf.cpp.

Referenced by load().

```
135 {
136     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
137 }
```

Here is the caller graph for this function:

12.35.4.9 QPoint KbPerf::dpi (int *index*) const [inline]

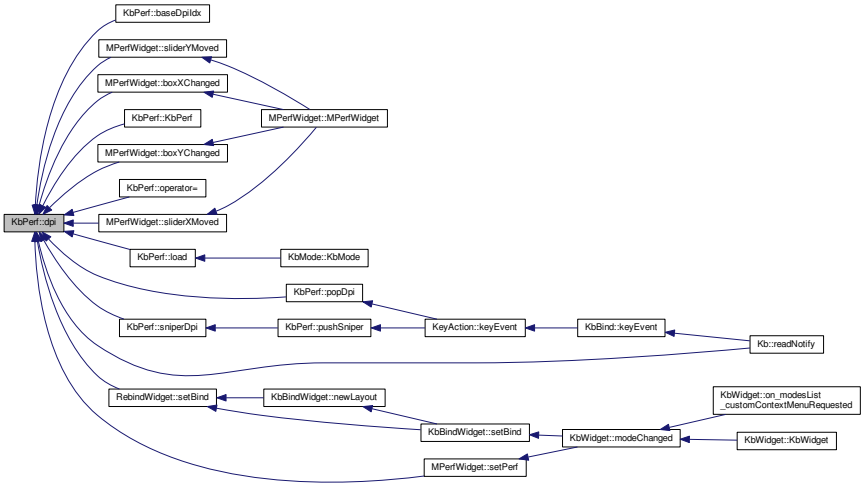
Definition at line 50 of file kbperf.h.

References DPI_COUNT, dpiX, and dpiY.

Referenced by baseDpiIdx(), MPerfWidget::boxXChanged(), MPerfWidget::boxYChanged(), KbPerf(), load(), operator=(), popDpi(), Kb::readNotify(), RebindWidget::setBind(), MPerfWidget::setPerf(), MPerfWidget::sliderXMoved(), MPerfWidget::sliderYMoved(), and sniperDpi().

```
50 { if(index < 0 || index >= DPI_COUNT) return QPoint(); return QPoint(
      dpiX[index], dpiY[index]); }
```

Here is the caller graph for this function:



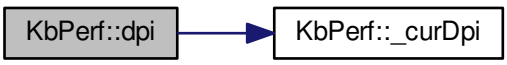
12.35.4.10 void KbPerf::dpi (int *index*, const QPoint & *newValue*)

Definition at line 247 of file kbperf.cpp.

References _curDpi(), _needsSave, _needsUpdate, DPI_COUNT, dpiBaseIdx, dpiX, dpiY, and pushedDpis.

```
247                                                 {
248     if(index < 0 || index >= DPI_COUNT)
249         return;
250     dpiX[index] = newValue.x();
251     dpiY[index] = newValue.y();
252     // Update current DPI if needed
253     if(dpiBaseIdx == index && pushedDpis.isEmpty()) {
254         _curDpi(QPoint(dpiX[index], dpiY[index]));
255     }
256     _needsUpdate = _needsSave = true;
257 }
```

Here is the call graph for this function:



12.35.4.11 void KbPerf::dpi (int *index*, int *newValue*) [inline]

Definition at line 55 of file kbperf.h.

References dpi().

Referenced by dpi().

```
55 { dpi(index, QPoint(newValue, newValue)); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.35.4.12 QColor KbPerf::dpiColor (int *index*) const [inline]

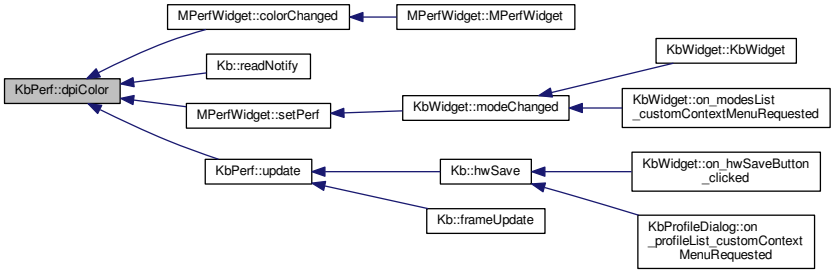
Definition at line 85 of file kbperf.h.

References dpiClr.

Referenced by MPerfWidget::colorChanged(), Kb::readNotify(), MPerfWidget::setPerf(), and update().

```
85 { return dpiClr[index]; }
```

Here is the caller graph for this function:



12.35.4.13 void KbPerf::dpiColor (int *index*, const QColor & *newColor*) [inline]

Definition at line 86 of file kbperf.h.

References _needsSave, _needsUpdate, and dpiClr.

```
86 { dpiClr[index] = newColor; _needsUpdate = _needsSave = true; }
```

12.35.4.14 void KbPerf::dpiCycleDown ( )

Definition at line 329 of file kbperf.cpp.

References baseDpiIdx(), DPI_COUNT, dpiOn, and SNIPER.

Referenced by KeyAction::keyEvent().

```
329                          {
330     int idx = baseDpiIdx();
331     do {
332         idx--;
333         if(idx <= SNIPER)
334             idx = DPI_COUNT - 1;
335     if(idx == baseDpiIdx())
336             return;
337     } while(!dpiOn[idx]);
338     baseDpiIdx(idx);
339 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.35.4.15 void KbPerf::dpiCycleUp ( )

Definition at line 317 of file kbperf.cpp.

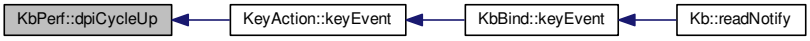
References baseDpiIdx(), DPI_COUNT, dpiOn, and SNIPER.

Referenced by KeyAction::keyEvent().

```
317                         {
318     int idx = baseDpiIdx();
319     do {
320         idx++;
321         if(idx >= DPI_COUNT)
322             idx = SNIPER + 1;
323         if(idx == baseDpiIdx())
324             return;
325     } while(!dpiOn[idx]);
326     baseDpiIdx(idx);
327 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.35.4.16 void KbPerf::dpiDown ( )

Definition at line 306 of file kbperf.cpp.

References baseDpiIdx(), dpiOn, and SNIPER.

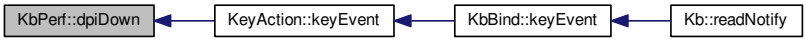
Referenced by KeyAction::keyEvent().

```
306                       {
```

```
307     int idx = baseDpiIdx();
308     do {
309         idx--;
310         if(idx <= SNIPER)
311             return;
312     } while(!dpiOn[idx]);
313     baseDpiIdx(idx);
314 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.35.4.17 bool KbPerf::dpiEnabled (int *index*) const [inline]

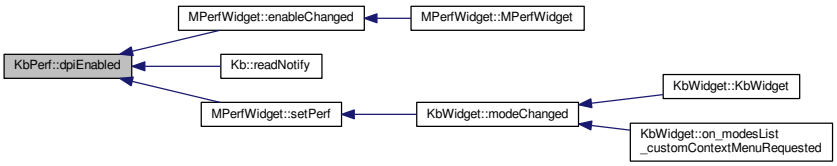
Definition at line 68 of file kbperf.h.

References dpiOn.

Referenced by MPerfWidget::enableChanged(), Kb::readNotify(), and MPerfWidget::setPerf().

```
68 { return dpiOn[index]; }
```

Here is the caller graph for this function:



12.35.4.18 void KbPerf::dpiEnabled (int *index*, bool *newEnabled*) [inline]

Definition at line 69 of file kbperf.h.

References _needsSave, _needsUpdate, and dpiOn.

```
69 { if(index <= 0) return; dpiOn[index] = newEnabled; _needsUpdate =
      _needsSave = true; }
```

12.35.4.19 bool KbPerf::dpiIndicator ( ) const [inline]

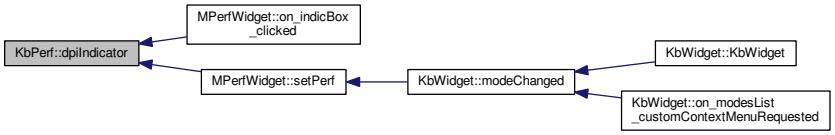
Definition at line 82 of file kbperf.h.

References _dpiIndicator.

Referenced by MPerfWidget::on_indicBox_clicked(), and MPerfWidget::setPerf().

```
82 { return _dpiIndicator; }
```

Here is the caller graph for this function:



12.35.4.20 void KbPerf::dpiIndicator (bool *newDpiIndicator*) [inline]

Definition at line 83 of file kbperf.h.

References _dpiIndicator, and _needsSave.

```
83 { _dpiIndicator = newDpiIndicator; _needsSave = true; }
```

12.35.4.21 void KbPerf::dpiUp ( )

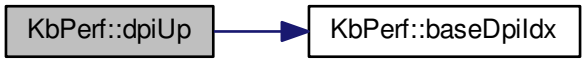
Definition at line 295 of file kbperf.cpp.

References baseDpiIdx(), DPI_COUNT, and dpiOn.

Referenced by KeyAction::keyEvent().

```
295                 {
296     // Scroll past disabled DPIs and choose the next one up
297     int idx = baseDpiIdx();
298     do {
299         idx++;
300         if(idx >= DPI_COUNT)
301             return;
302     } while(!dpiOn[idx]);
303     baseDpiIdx(idx);
304 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.35.4.22 void KbPerf::getIndicator (indicator *index*, QColor & *color1*, QColor & *color2*, QColor & *color3*, bool & *software_enable*, i_hw & *hardware_enable*)

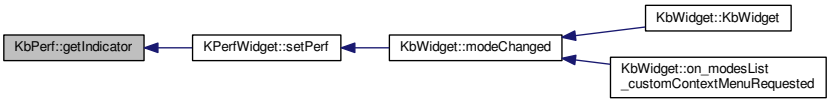
Definition at line 341 of file kbperf.cpp.

References HW_IMAX, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, MUTE, muteNAColor, and NONE.

Referenced by KPerfWidget::setPerf().

```
341
                                {
342     if(index < 0 || index >= I_COUNT)
343         return;
344     color1 = iColor[index][0];
345     color2 = iColor[index][1];
346     if(index == LIGHT)
347         color3 = light100Color;
348     else if(index == MUTE)
349         color3 = muteNAColor;
350     else
351         color3 = QColor();
352     software_enable = iEnable[index];
353     hardware_enable = (index <= HW_IMAX ? hwIType[index] : NONE);
354 }
```

Here is the caller graph for this function:



12.35.4.23 float KbPerf::iOpacity ( ) const [inline]

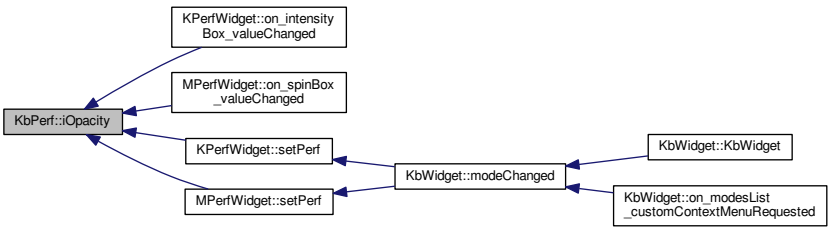
Definition at line 79 of file kbperf.h.

References _iOpacity.

Referenced by KPerfWidget::on_intensityBox_valueChanged(), MPerfWidget::on_spinBox_valueChanged(), KPerfWidget::setPerf(), and MPerfWidget::setPerf().

```
79 { return _iOpacity; }
```

Here is the caller graph for this function:



12.35.4.24 void KbPerf::iOpacity (float *newIOpacity*) [inline]

Definition at line 80 of file kbperf.h.

References _iOpacity, and _needsSave.

```
80 { _iOpacity = newIOpacity; _needsSave = true; }
```

12.35.4.25 height KbPerf::liftHeight ( ) const [inline]

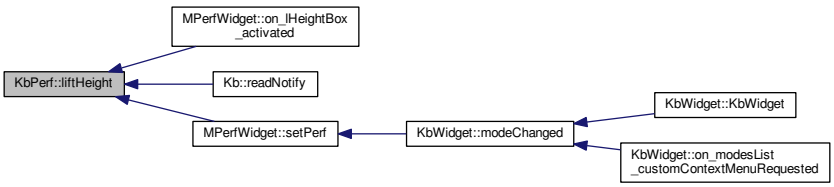
Definition at line 38 of file kbperf.h.

References _liftHeight.

Referenced by MPerfWidget::on_lHeightBox_activated(), Kb::readNotify(), and MPerfWidget::setPerf().

```
38 { return _liftHeight; }
```

Here is the caller graph for this function:



12.35.4.26 void KbPerf::liftHeight (height *newHeight*)

Definition at line 373 of file kbperf.cpp.

References _liftHeight, _needsSave, _needsUpdate, and HIGH.

```
373                                        {
374     if(newHeight < LOW || newHeight > HIGH)
375         return;
376     _liftHeight = newHeight;
377     _needsUpdate = _needsSave = true;
378 }
```

12.35.4.27 KbLight * KbPerf::light ( ) const [private]

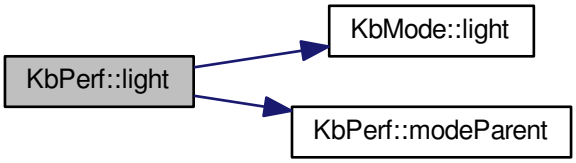
Definition at line 102 of file kbperf.cpp.

References KbMode::light(), and modeParent().

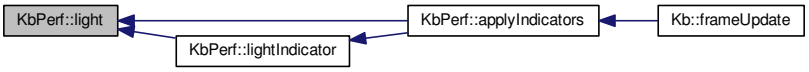
Referenced by applyIndicators(), and lightIndicator().

```
102                                {
103     return modeParent()->light();
104 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.28 void KbPerf::lightIndicator (const char * *name*, QRgb *rgba*) [private]

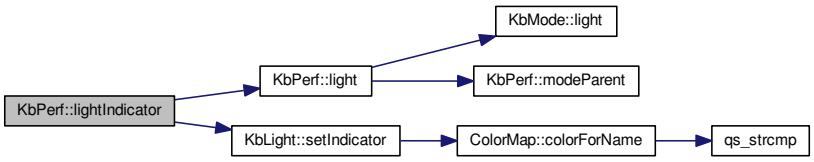
Definition at line 432 of file kbperf.cpp.

References _iOpacity, light(), and KbLight::setIndicator().

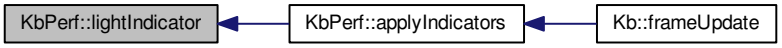
Referenced by applyIndicators().

```
432                                                    {
433     int a = round(qAlpha(rgba) * _iOpacity);
434     if(a <= 0)
435         return;
436     light()->setIndicator(name, qRgba(qRed(rgba), qGreen(rgba), qBlue(rgba), a));
437 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.29 void KbPerf::load (CkbSettings & *settings*)

Definition at line 106 of file kbperf.cpp.

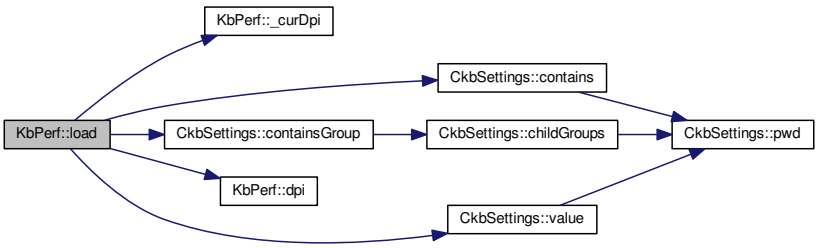
References _angleSnap, _curDpi(), _dpiIndicator, _iOpacity, _liftHeight, _needsSave, CkbSettings::contains(), CkbSettings::containsGroup(), didLoad(), dpi(), DPI_COUNT, dpiBaseIdx, dpiClr, dpiOn, dpiX, dpiY, HIGH, HW_IMAX, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, LOCK, MACRO, MEDIUM, MODE, MUTE, muteNAColor, NORMAL, OTHER, pushedDpis, runningPushIdx, and CkbSettings::value().

Referenced by KbMode::KbMode().

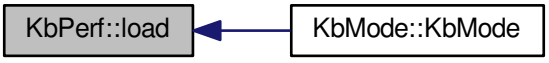
```
106                                     {
107     pushedDpis.clear();
108     runningPushIdx = 1;
109     _needsSave = false;
110     bool readIndicators = true;
111     if(!settings.containsGroup("Performance/Indicators")){
112         // Read old indicator settings from the lighting group, if present
113         // (ckb <= v0.2.0)
114         SGroup group(settings, "Lighting");
115         if(settings.contains("InactiveIndicators")){
116             bool inOk = false;
117             int inactive = settings.value("InactiveIndicators").toInt(&inOk);
118             if(!inOk || inactive > 2)
119                 inactive = 2;
120             if(inactive == 1)
121                 _iOpacity = 0.75f;
122             else if(inactive == 0)
123                 _iOpacity = 0.5f;
124             else if(inactive < 0){
125                 // Indicators disabled
126                 iEnable[MODE] = iEnable[MACRO] = iEnable[
      LIGHT] = iEnable[LOCK]  = iEnable[MUTE] = false;
127             }
128             bool showMute = (settings.value("ShowMute").toInt(&inOk) != 0);
129             if(inOk && !showMute)
130                 iEnable[MUTE] = false;
131             readIndicators = false;
132         }
133     }
134     SGroup group(settings, "Performance");
135     // Read DPI settings
136     {
137         SGroup group(settings, "DPI");
138         for(int i = 0; i < DPI_COUNT; i++){
139             QString iStr = QString::number(i);
```

```
140             QPoint value = settings.value(iStr).toPoint();
141             if(value.isNull())
142                 continue;
143             dpiX[i] = value.x(); dpiY[i] = value.y();
144             QColor color = settings.value(iStr + "RGB").toString();
145             if(color.isValid())
146                 dpiClr[i] = color;
147             if(i != 0)
148                 dpiOn[i] = !settings.value(iStr + "Disabled").toBool();
149         }
150         QColor color = settings.value("6RGB").toString();
151         if(color.isValid())
152             dpiClr[OTHER] = color;
153         if (settings.contains("CurIdx")) {
154             dpiBaseIdx = settings.value("CurIdx").toInt();
155         } else {
156             // If there isn't a setting for current DPI stage, pick the first
157             // enabled one. Failing that just pick stage 1.
158             dpiBaseIdx = 1;
159             for (int i = 1; i < DPI_COUNT; i++) {
160                 if (dpiOn[i]) {
161                     dpiBaseIdx = i;
162                     break;
163                 }
164             }
165         }
166         _curDpi(dpi(dpiBaseIdx));
167     }
168     // Read misc. mouse settings
169     _liftHeight = (height)settings.value("LiftHeight").toInt();
170     if(_liftHeight < LOW || _liftHeight > HIGH)
171         _liftHeight = MEDIUM;
172     _angleSnap = settings.value("AngleSnap").toBool();
173     if(settings.contains("NoIndicator")){
174         // ckb <= v0.2.0
175         _dpiIndicator = !settings.value("NoIndicator").toBool();
176     } else {
177         _dpiIndicator = settings.value("Indicators/DPI", true).toBool();
178     }
179     // Read indicator settings
180     if(readIndicators){
181         SGroup group(settings, "Indicators");
182         _iOpacity = settings.value("Opacity", 100).toInt() / 100.f;
183         for(int i = 0; i < I_COUNT; i++){
184             SGroup group(settings, QString::number(i));
185             QColor color = settings.value("RGB0").toString();
186             if(color.isValid())
187                 iColor[i][0] = color;
188             color = settings.value("RGB1").toString();
189             if(color.isValid())
190                 iColor[i][1] = color;
191             if(i == LIGHT){
192                 color = settings.value("RGB2").toString();
193                 if(color.isValid())
194                     light100Color = color;
195             } else if(i == MUTE){
196                 color = settings.value("RGB2").toString();
197                 if(color.isValid())
198                     muteNAColor = color;
199             }
200             if(i <= HW_IMAX){
201                 iEnable[i] = settings.value("Enable", false).toBool();
202                 hwIType[i] = (i_hw)settings.value("Hardware", (int)
    NORMAL).toInt();
203             } else {
204                 iEnable[i] = settings.value("Enable", true).toBool();
205             }
206         }
207     }
208     emit didLoad();
209 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



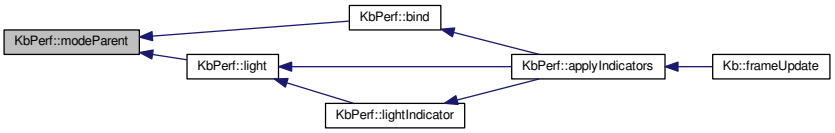
12.35.4.30 KbMode* KbPerf::modeParent ( ) const [inline], [private]

Definition at line 127 of file kbperf.h.

Referenced by bind(), and light().

```
127 { return (KbMode*)parent(); }
```

Here is the caller graph for this function:



12.35.4.31 bool KbPerf::needsSave ( ) const [inline]

Definition at line 28 of file kbperf.h.

References _needsSave.

Referenced by KbMode::needsSave().

```
28 { return _needsSave; }
```

Here is the caller graph for this function:

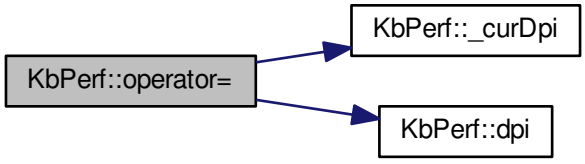
12.35.4.32 const KbPerf & KbPerf::operator= (const KbPerf & *rhs*)

Definition at line 77 of file kbperf.cpp.

References _angleSnap, _curDpi(), _dpiIndicator, _iOpacity, _liftHeight, _needsSave, _needsUpdate, dpi(), DPI_COUNT, dpiBaseIdx, dpiClr, dpiCurX, dpiCurY, dpiOn, dpiX, dpiY, hwIType, I_COUNT, iColor, iEnable, light100Color, muteNAColor, and runningPushIdx.

```
77                                                  {
78     dpiCurX = other.dpiCurX; dpiCurY = other.dpiCurY; dpiBaseIdx = other.dpiBaseIdx
       ; runningPushIdx = 1;
79     _iOpacity = other._iOpacity; light100Color = other.light100Color;
       muteNAColor = other.muteNAColor; _dpiIndicator = other._dpiIndicator;
80     _liftHeight = other._liftHeight; _angleSnap = other._angleSnap;
81     _needsUpdate = true; _needsSave = true;
82     memcpy(dpiX, other.dpiX, sizeof(dpiX));
83     memcpy(dpiY, other.dpiY, sizeof(dpiY));
84     for(int i = 0; i < DPI_COUNT + 1; i++)
85         dpiClr[i] = other.dpiClr[i];
86     memcpy(dpiOn, other.dpiOn, sizeof(dpiOn));
87     for(int i = 0; i < I_COUNT; i++){
88         iColor[i][0] = other.iColor[i][0];
89         iColor[i][1] = other.iColor[i][1];
90     }
91     memcpy(iEnable, other.iEnable, sizeof(iEnable));
92     memcpy(hwIType, other.hwIType, sizeof(hwIType));
93     // Don't copy pushed DPI states. If the other mode has any, restore the original DPI
94     _curDpi(dpi(dpiBaseIdx));
95     return other;
96 }
```

Here is the call graph for this function:



12.35.4.33 void KbPerf::popDpi (quint64 *pushIdx*)

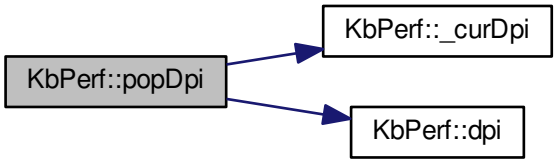
Definition at line 281 of file kbperf.cpp.

References _curDpi(), _needsSave, _needsUpdate, dpi(), dpiBaseIdx, map_last, and pushedDpis.

Referenced by KeyAction::keyEvent().

```
281                                 {
282     if(!pushedDpis.contains(pushIdx)) {
283       return;
284     }
285     pushedDpis.remove(pushIdx);
286     if (pushedDpis.isEmpty()) {
287         _curDpi(dpi(dpiBaseIdx));
288     } else {
289         // Set the DPI to the last-pushed value still on the stack
290         _curDpi(map_last(pushedDpis));
291     }
292     _needsUpdate = _needsSave = true;
293 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.35.4.34 quint64 KbPerf::pushDpi (const QPoint & *newDpi*)

Definition at line 274 of file kbperf.cpp.

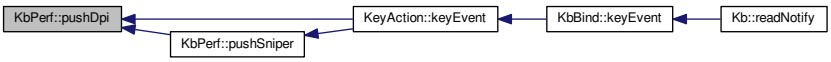
References _curDpi(), pushedDpis, and runningPushIdx.

Referenced by KeyAction::keyEvent(), and pushSniper().

```
274                                         {
275     quint64 index = runningPushIdx++;
276     pushedDpis[index] = newDpi;
277     _curDpi(newDpi);
278     return index;
279 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.35.4.35 quint64 KbPerf::pushDpi (int *newDpi*) [inline]

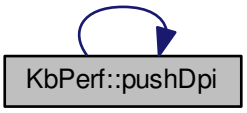
Definition at line 74 of file kbperf.h.

References pushDpi().

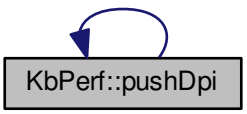
Referenced by pushDpi().

```
74 { return pushDpi(QPoint(newDpi, newDpi)); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.36 quint64 KbPerf::pushSniper ( ) [inline]

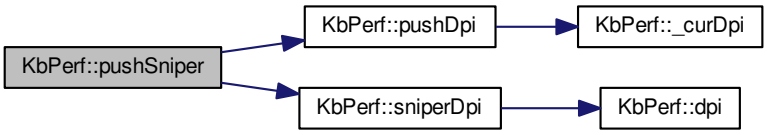
Definition at line 75 of file kbperf.h.

References pushDpi(), and sniperDpi().

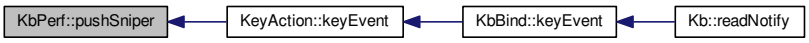
Referenced by KeyAction::keyEvent().

```
75 { return pushDpi(sniperDpi()); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.37 void KbPerf::save (CkbSettings & *settings*)

Definition at line 211 of file kbperf.cpp.

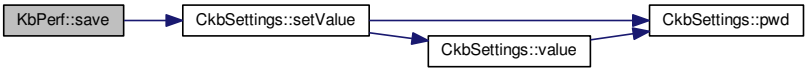
References _angleSnap, _dpiIndicator, _liftHeight, _needsSave, DPI_COUNT, dpiBaseIdx, dpiClr, dpiOn, dpiX, dpiY, HW_IMAX, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, MUTE, muteNAColor, OTHER, and CkbSettings::setValue().

Referenced by KbMode::save().

```
211                                     {
212     _needsSave = false;
213     SGroup group(settings, "Performance");
214     {
215         SGroup group(settings, "DPI");
216         for(int i = 0; i < DPI_COUNT; i++){
217             QString iStr = QString::number(i);
218             settings.setValue(iStr, QPoint(dpiX[i], dpiY[i]));
219             settings.setValue(iStr + "RGB", dpiClr[i].name(QColor::HexArgb));
220             if(i != 0)
221                 settings.setValue(iStr + "Disabled", !dpiOn[i]);
222         }
223         settings.setValue("6RGB", dpiClr[OTHER].name(QColor::HexArgb));
224     // Ignore pushed modes when saving current DPI.
225         settings.setValue("CurIdx", dpiBaseIdx);
226     }
227     settings.setValue("LiftHeight", _liftHeight);
228     settings.setValue("AngleSnap", _angleSnap);
229     {
230         SGroup group(settings, "Indicators");
231         settings.setValue("DPI", _dpiIndicator);
232         for(int i = 0; i < I_COUNT; i++){
233             SGroup group(settings, QString::number(i));
```

```
234             settings.setValue("RGB0", iColor[i][0].name(QColor::HexArgb));
235             settings.setValue("RGB1", iColor[i][1].name(QColor::HexArgb));
236             if(i == LIGHT)
237                 settings.setValue("RGB2", light100Color.name(QColor::HexArgb));
238             else if(i == MUTE)
239                 settings.setValue("RGB2", muteNAColor.name(QColor::HexArgb));
240             settings.setValue("Enable", iEnable[i]);
241             if(i <= HW_IMAX)
242                 settings.setValue("Hardware", (int)hwIType[i]);
243         }
244     }
245 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.35.4.38 void KbPerf::setIndicator (indicator *index*, const QColor & *color1*, const QColor & *color2*, const QColor & *color3* = `QColor()`, bool *software_enable* = `true`, i_hw *hardware_enable* = NORMAL)

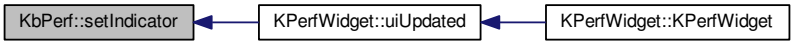
Definition at line 356 of file kbperf.cpp.

References _needsSave, _needsUpdate, HW_IMAX, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, MUTE, muteNAColor, NONE, and NORMAL.

Referenced by KPerfWidget::uiUpdated().

```
356                                                 {
357     if(index < 0 || index >= I_COUNT)
358         return;
359     iColor[index][0] = color1;
360     iColor[index][1] = color2;
361     if(index == LIGHT)
362         light100Color = color3;
363     else if(index == MUTE)
364         muteNAColor = color3;
365     iEnable[index] = software_enable;
366     if(hardware_enable == NONE)
367         hardware_enable = NORMAL;
368     if(index <= HW_IMAX)
369         hwIType[index] = hardware_enable;
370     _needsUpdate = _needsSave = true;
371 }
```

Here is the caller graph for this function:



12.35.4.39 void KbPerf::setNeedsUpdate ( ) [inline]

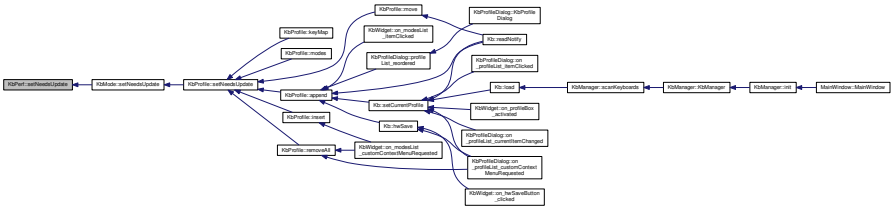
Definition at line 116 of file kbperf.h.

References _needsUpdate.

Referenced by KbMode::setNeedsUpdate().

```
116 { _needsUpdate = true; }
```

Here is the caller graph for this function:



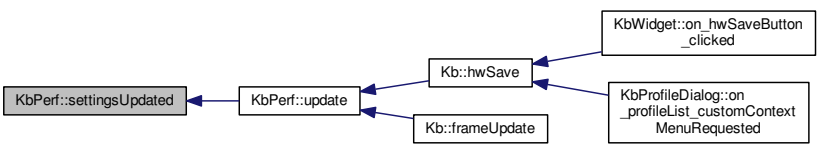
12.35.4.40 void KbPerf::settingsUpdated ( ) [signal]

Definition at line 140 of file moc_kbperf.cpp.

Referenced by update().

```
141 {
142     QMetaObject::activate(this, &staticMetaObject, 1, Q_NULLPTR);
143 }
```

Here is the caller graph for this function:



12.35.4.41 QPoint KbPerf::sniperDpi ( ) const [inline]

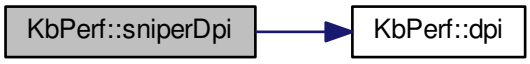
Definition at line 52 of file kbperf.h.

References dpi(), and SNIPER.

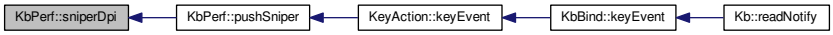
Referenced by pushSniper().

```
52 { return dpi(SNIPER); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



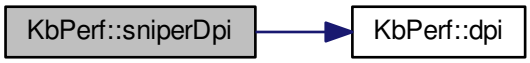
12.35.4.42 void KbPerf::sniperDpi (const QPoint & *newValue*) [inline]

Definition at line 53 of file kbperf.h.

References dpi(), and SNIPER.

```
53 { dpi(SNIPER, newValue); }
```

Here is the call graph for this function:



12.35.4.43 void KbPerf::sniperDpi (int *newValue*) [inline]

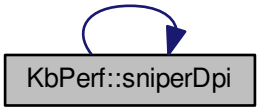
Definition at line 56 of file kbperf.h.

References sniperDpi().

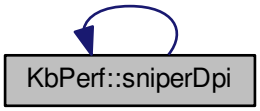
Referenced by sniperDpi().

```
56 { sniperDpi(QPoint(newValue, newValue)); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.35.4.44 void KbPerf::update (QFile & *cmd,* int *notifyNumber,* bool *force,* bool *saveCustomDpi*)

Definition at line 385 of file kbperf.cpp.

References _angleSnap, _liftHeight, _needsUpdate, DPI_COUNT, dpiBaseIdx, dpiColor(), dpiCurX, dpiCurY, dpiOn, dpiX, dpiY, HW_I_COUNT, hwIType, OFF, ON, pushedDpis, and settingsUpdated().

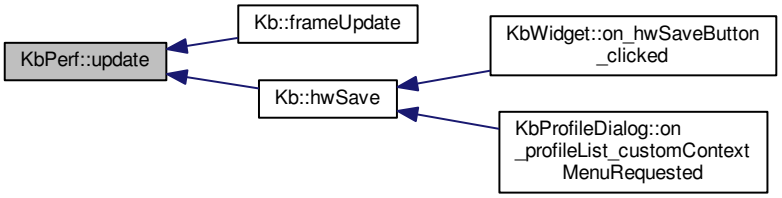
Referenced by Kb::frameUpdate(), and Kb::hwSave().

```
385                                                                                 {
386     if(!force && !_needsUpdate)
387         return;
388     emit settingsUpdated();
389     _needsUpdate = false;
390     // Save DPI stage 0 (sniper)
391     // If the mouse is set to a custom DPI, save it in stage 0
392     int stage = pushedDpis.isEmpty() ? dpiBaseIdx : 0;
393     if(!pushedDpis.isEmpty() && saveCustomDpi) {
394         cmd.write(QString("dpi 0:%1,%2").arg(dpiCurX).arg(dpiCurY).toLatin1());
395     } else {
396         // Otherwise, save stage 0 normally
397         cmd.write(QString("dpi 0:%1,%2").arg(dpiX[0]).arg(dpiY[0]).toLatin1());
398     }
399     // Save stages 1 - 5
400     for(int i = 1; i < DPI_COUNT; i++){
401         if(!dpiOn[i] && stage != i)
402             cmd.write(QString(" %1:off").arg(i).toLatin1());
403         else
404         cmd.write(QString(" %1:%2,%3").arg(i).arg(dpiX[i]).arg(dpiY[i]).toLatin1());
405     }
406     // Save stage selection, lift height, and angle snap
407     cmd.write(QString(" dpisel %1 lift %2 snap %3").arg(stage).arg(
    _liftHeight).arg(_angleSnap ? "on" : "off").toLatin1());
408     // Save DPI colors
409     cmd.write(" rgb");
410     for(int i = 0; i < DPI_COUNT; i++){
```

```
411         QColor color = dpiColor(i);
412         cmd.write(" dpi");
413         char output[9];
414         snprintf(output, sizeof(output), "%1d:%02x%02x%02x", i, color.red(), color.green(), color.blue());
415         cmd.write(output);
416     }
417     // Enable indicator notifications
418     cmd.write(QString("\n@%1 inotify all").arg(notifyNumber).toLatin1());
419     // Set indicator state
420     const char* iNames[HW_I_COUNT] = { "num", "caps", "scroll" };
421     for(int i = 0; i < HW_I_COUNT; i++){
422         if(hwIType[i] == ON)
423             cmd.write(" ion ");
424         else if(hwIType[i] == OFF)
425             cmd.write(" ioff ");
426         else
427             cmd.write(" iauto ");
428         cmd.write(iNames[i]);
429     }
430 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.35.5 Field Documentation

12.35.5.1 bool KbPerf::_angleSnap [private]

Definition at line 164 of file kbperf.h.

Referenced by angleSnap(), load(), operator=(), save(), and update().

12.35.5.2 bool KbPerf::_dpiIndicator [private]

Definition at line 160 of file kbperf.h.

Referenced by applyIndicators(), dpiIndicator(), load(), operator=(), and save().

12.35.5.3 float KbPerf::_iOpacity [private]

Definition at line 155 of file kbperf.h.

Referenced by applyIndicators(), iOpacity(), lightIndicator(), load(), and operator=().

12.35.5.4 height KbPerf::_liftHeight [private]

Definition at line 163 of file kbperf.h.

Referenced by liftHeight(), load(), operator=(), save(), and update().

12.35.5.5 bool KbPerf::_needsSave [private]

Definition at line 166 of file kbperf.h.

Referenced by _curDpi(), angleSnap(), baseDpiIdx(), dpi(), dpiColor(), dpiEnabled(), dpiIndicator(), iOpacity(), liftHeight(), load(), needsSave(), operator=(), popDpi(), save(), and setIndicator().

12.35.5.6 bool KbPerf::_needsUpdate [private]

Definition at line 166 of file kbperf.h.

Referenced by _curDpi(), angleSnap(), baseDpiIdx(), dpi(), dpiColor(), dpiEnabled(), liftHeight(), operator=(), popDpi(), setIndicator(), setNeedsUpdate(), and update().

12.35.5.7 const int KbPerf::DPI_COUNT = 6 [static]

Definition at line 46 of file kbperf.h.

Referenced by dpi(), dpiCycleDown(), dpiCycleUp(), dpiUp(), KbPerf(), KeyAction::keyEvent(), load(), operator=(), Kb::readNotify(), save(), RebindWidget::setBind(), and update().

12.35.5.8 const int KbPerf::DPI_MAX = 12000 [static]

Definition at line 47 of file kbperf.h.

12.35.5.9 const int KbPerf::DPI_MIN = 100 [static]

Definition at line 47 of file kbperf.h.

Referenced by dpiExp(), and dpiLog().

12.35.5.10 int KbPerf::dpiBaseIdx [private]

Definition at line 142 of file kbperf.h.

Referenced by baseDpiIdx(), dpi(), KbPerf(), load(), operator=(), popDpi(), save(), and update().

12.35.5.11 QColor KbPerf::dpiClr[DPI_COUNT+1] [private]

Definition at line 143 of file kbperf.h.

Referenced by applyIndicators(), dpiColor(), KbPerf(), load(), operator=(), and save().

12.35.5.12 int KbPerf::dpiCurX [private]

Definition at line 138 of file kbperf.h.

Referenced by _curDpi(), KbPerf(), operator=(), and update().

12.35.5.13 int KbPerf::dpiCurY [private]

Definition at line 138 of file kbperf.h.

Referenced by _curDpi(), KbPerf(), operator=(), and update().

12.35.5.14 bool KbPerf::dpiOn[DPI_COUNT] [private]

Definition at line 144 of file kbperf.h.

Referenced by dpiCycleDown(), dpiCycleUp(), dpiDown(), dpiEnabled(), dpiUp(), KbPerf(), load(), operator=(), save(), and update().

12.35.5.15 int KbPerf::dpiX[DPI_COUNT] [private]

Definition at line 135 of file kbperf.h.

Referenced by dpi(), KbPerf(), load(), operator=(), save(), and update().

12.35.5.16 int KbPerf::dpiY[DPI_COUNT] [private]

Definition at line 136 of file kbperf.h.

Referenced by dpi(), KbPerf(), load(), operator=(), save(), and update().

12.35.5.17 const int KbPerf::HW_I_COUNT = (int)HW_IMAX + 1 [static]

Definition at line 107 of file kbperf.h.

Referenced by KbPerf(), and update().

12.35.5.18 i_hw KbPerf::hwlType[HW_I_COUNT] [private]

Definition at line 159 of file kbperf.h.

Referenced by getIndicator(), KbPerf(), load(), operator=(), save(), setIndicator(), and update().

12.35.5.19 const int KbPerf::I_COUNT = (int)MUTE + 1 [static]

Definition at line 107 of file kbperf.h.

Referenced by getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

12.35.5.20 QColor KbPerf::iColor[I_COUNT][2] [private]

Definition at line 156 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

12.35.5.21 bool KbPerf::iEnable[I_COUNT] [private]

Definition at line 158 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

12.35.5.22 QColor KbPerf::light100Color [private]

Definition at line 157 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

12.35.5.23 QColor KbPerf::muteNAColor [private]

Definition at line 157 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

12.35.5.24 const int KbPerf::OTHER = DPI_COUNT [static]

Definition at line 84 of file kbperf.h.

Referenced by applyIndicators(), MPerfWidget::colorChanged(), load(), save(), and MPerfWidget::setPerf().

12.35.5.25 QMap<quint64, QPoint> KbPerf::pushedDpis [private]

Definition at line 148 of file kbperf.h.

Referenced by baseDpiIdx(), dpi(), load(), popDpi(), pushDpi(), and update().

12.35.5.26 uint KbPerf::runningPushIdx [private]

Definition at line 149 of file kbperf.h.

Referenced by load(), operator=(), and pushDpi().

12.35.5.27 const int KbPerf::SNIPER = 0 [static]

Definition at line 46 of file kbperf.h.

Referenced by dpiCycleDown(), dpiCycleUp(), dpiDown(), and sniperDpi().

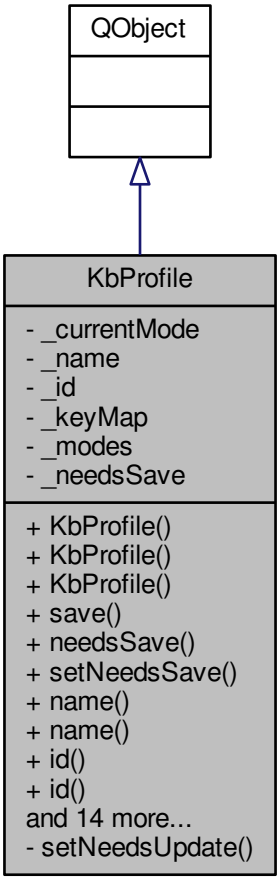
The documentation for this class was generated from the following files:

- src/ckb/kbperf.h
- src/ckb/kbperf.cpp
- src/ckb/release/moc_kbperf.cpp

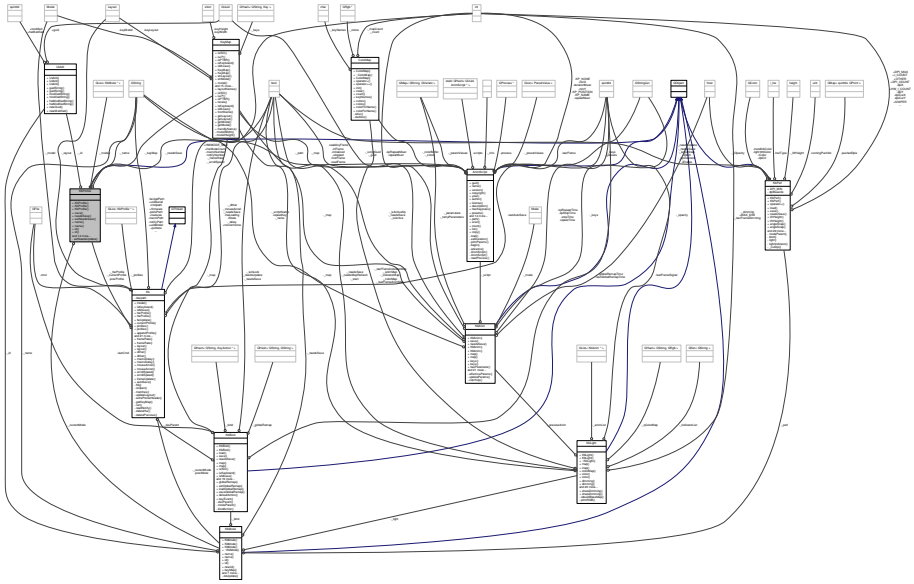
12.36 KbProfile Class Reference

`#include <src/ckb/kbprofile.h>`

Inheritance diagram for KbProfile:



Collaboration diagram for KbProfile:



Public Types

- typedef QList< KbMode * > ModeList

Public Member Functions

- KbProfile (Kb *parent, const KeyMap &keyMap, const KbProfile &other)
- KbProfile (Kb *parent, const KeyMap &keyMap, const QString &guid="", const QString &modified="")
- KbProfile (Kb *parent, const KeyMap &keyMap, CkbSettings &settings, const QString &guid)
- void save (CkbSettings &settings)
- bool needsSave () const
- void setNeedsSave ()
- QString name () const
- void name (const QString &newName)
- UsbId & id ()
- void id (const UsbId &newId)
- void newId ()
- const KeyMap & keyMap () const
- void keyMap (const KeyMap &newKeyMap)
- const ModeList & modes () const
- void modes (const QList< KbMode * > &newModes)
- void append (KbMode *newMode)
- void insert (int index, KbMode *newMode)
- void removeAll (KbMode *mode)
- void move (int from, int to)
- int modeCount () const
- int indexOf (KbMode *mode) const
- KbMode * find (const QUuid &id)
- KbMode * currentMode () const
- void currentMode (KbMode *newCurrentMode)

Private Member Functions

- void setNeedsUpdate ()

Private Attributes

- KbMode * _currentMode
- QString _name
- UsbId _id
- KeyMap _keyMap
- ModeList _modes
- bool _needsSave

12.36.1 Detailed Description

Definition at line 11 of file kbprofile.h.

12.36.2 Member Typedef Documentation

12.36.2.1 typedef QList<KbMode*> KbProfile::ModeList

Definition at line 41 of file kbprofile.h.

12.36.3 Constructor & Destructor Documentation

12.36.3.1 KbProfile::KbProfile (Kb * *parent*, const KeyMap & *keyMap*, const KbProfile & *other*) [explicit]

Definition at line 4 of file kbprofile.cpp.

References _currentMode, _modes, and modes().

```
4                                                                                   :
5     QObject(parent), _currentMode(0), _name(other._name),
      _id(other._id), _keyMap(keyMap), _needsSave(true)
6 {
7     foreach(KbMode* mode, other.modes()){
8         KbMode* newMode = new KbMode(parent, keyMap, *mode);
9         if(!_currentMode || mode == other._currentMode)
10            _currentMode = newMode;
11        _modes.append(newMode);
12    }
13 }
```

Here is the call graph for this function:

12.36.3.2 KbProfile::KbProfile (Kb * *parent*, const KeyMap & *keyMap*, const QString & *guid* = " ", const QString & *modified* = " ") [explicit]

Definition at line 15 of file kbprofile.cpp.

References _id, and UsbId::guid.

```
15                                                                                                                  :
16     QObject(parent), _currentMode(0), _id(guid, modified.toUInt(0, 16)),
      _keyMap(keyMap), _needsSave(true)
17 {
18     if(_id.guid.isNull())
19         _id.guid = QUuid::createUuid();
20 }
```

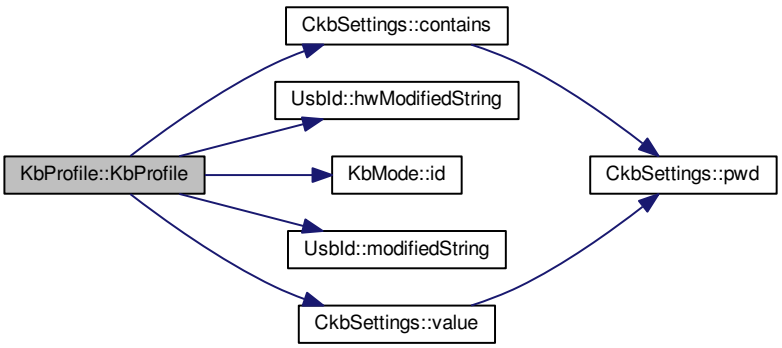
12.36.3.3 KbProfile::KbProfile (Kb * *parent*, const KeyMap & *keyMap*, CkbSettings & *settings*, const QString & *guid*) [explicit]

Definition at line 22 of file kbprofile.cpp.

References _currentMode, _id, _keyMap, _modes, _name, CkbSettings::contains(), current, UsbId::guid, UsbId::hwModified, UsbId::hwModifiedString(), KbMode::id(), UsbId::modified, UsbId::modifiedString(), and CkbSettings::value().

```
22                                                                                                                  :
23     QObject(parent), _currentMode(0), _id(guid, 0),
      _keyMap(keyMap), _needsSave(false)
24 {
25     // Load data from preferences
26     SGroup group(settings, guid);
27     _name = settings.value("Name").toString().trimmed();
28     if(_name == "")
29         _name = "Unnamed";
30     _id.modifiedString(settings.value("Modified").toString());
31     if(settings.contains("HwModified"))
32         _id.hwModifiedString(settings.value("HwModified").toString());
33     else
34         _id.hwModified = _id.modified;
35     QUuid current = settings.value("CurrentMode").toString().trimmed();
36     // Load modes
37     uint count = settings.value("ModeCount").toUInt();
38     for(uint i = 0; i < count; i++){
39         SGroup group(settings, QString::number(i));
40         KbMode* mode = new KbMode(parent, _keyMap, settings);
41         _modes.append(mode);
42         // Set currentMode to the mode matching the current GUID, or the first mode in case it's not found
43         if(current == mode->id().guid || !_currentMode)
44             _currentMode = mode;
45     }
46 }
```

Here is the call graph for this function:



12.36.4 Member Function Documentation

12.36.4.1 void KbProfile::append (KbMode * *newMode*) [inline]

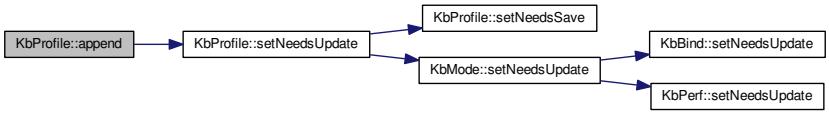
Definition at line 44 of file kbprofile.h.

References _modes, and setNeedsUpdate().

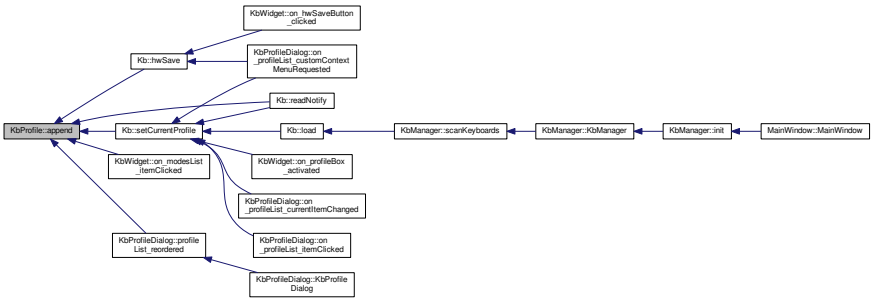
Referenced by Kb::hwSave(), KbWidget::on_modesList_itemClicked(), KbProfileDialog::profileList_reordered(), Kb::readNotify(), and Kb::setCurrentProfile().

```
44 { setNeedsUpdate(); _modes.append(newMode); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.4.2 KbMode* KbProfile::currentMode ( ) const [inline]

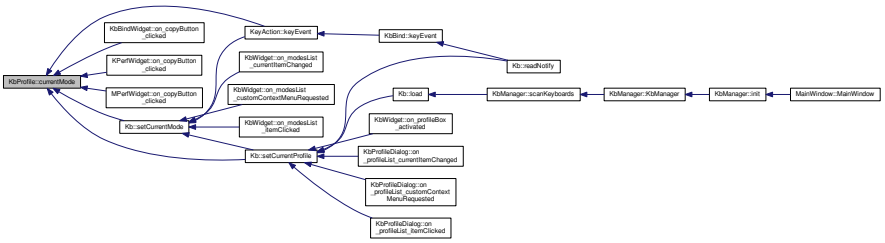
Definition at line 54 of file kbprofile.h.

References _currentMode.

Referenced by KeyAction::keyEvent(), KbBindWidget::on_copyButton_clicked(), KPerfWidget::on_copyButton_clicked(), MPerfWidget::on_copyButton_clicked(), Kb::setCurrentMode(), and Kb::setCurrentProfile().

```
54 { return _currentMode; }
```

Here is the caller graph for this function:



12.36.4.3 void KbProfile::currentMode (KbMode * *newCurrentMode*) [inline]

Definition at line 55 of file kbprofile.h.

References _currentMode, and _needsSave.

```
55 { _needsSave = true; _currentMode = newCurrentMode; }
```

12.36.4.4 KbMode* KbProfile::find (const QUuid & *id*) [inline]

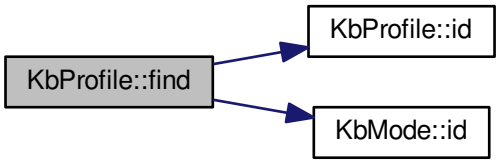
Definition at line 51 of file kbprofile.h.

References _modes, UsbId::guid, id(), and KbMode::id().

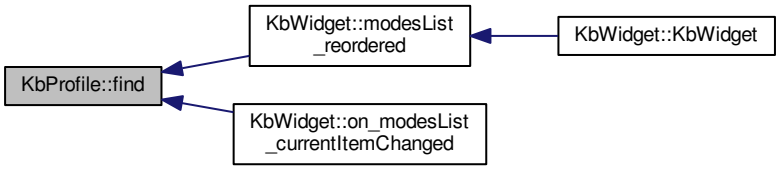
Referenced by KbWidget::modesList_reordered(), and KbWidget::on_modesList_currentItemChanged().

```
51 { foreach(KbMode* mode, _modes) { if(mode->id().guid == id) return mode; } return 0; }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.4.5 UsbId& KbProfile::id ( ) [inline]

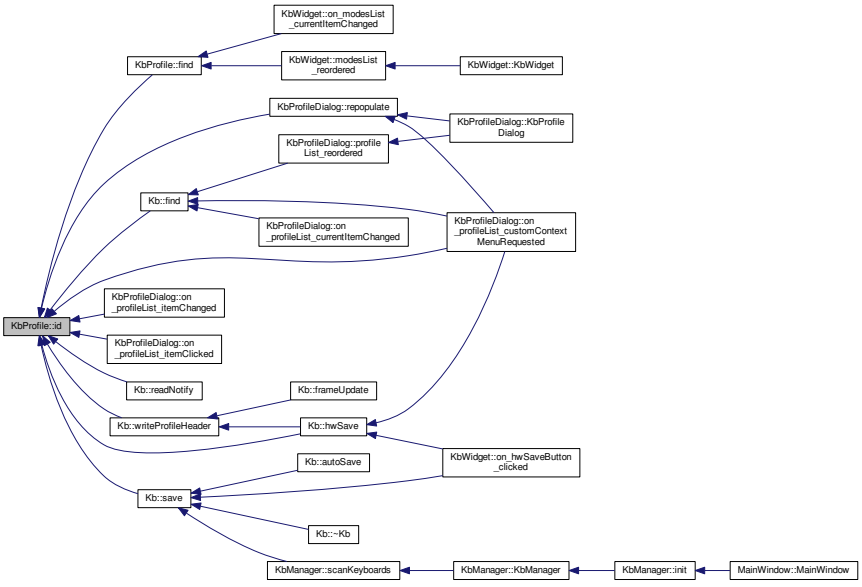
Definition at line 30 of file kbprofile.h.

References _id.

Referenced by find(), Kb::find(), Kb::hwSave(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_itemChanged(), KbProfileDialog::on_profileList_itemClicked(), Kb::readNotify(), KbProfileDialog::repopulate(), Kb::save(), and Kb::writeProfileHeader().

```
30 { return _id; }
```

Here is the caller graph for this function:



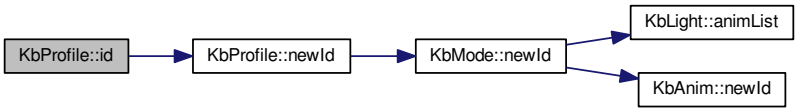
12.36.4.6 void KbProfile::id (const UsbId & *newId*) [inline]

Definition at line 31 of file kbprofile.h.

References _id, _needsSave, and newId().

```
31 { _needsSave = true; _id = newId; }
```

Here is the call graph for this function:



12.36.4.7 int KbProfile::indexOf (KbMode * *mode*) const [inline]

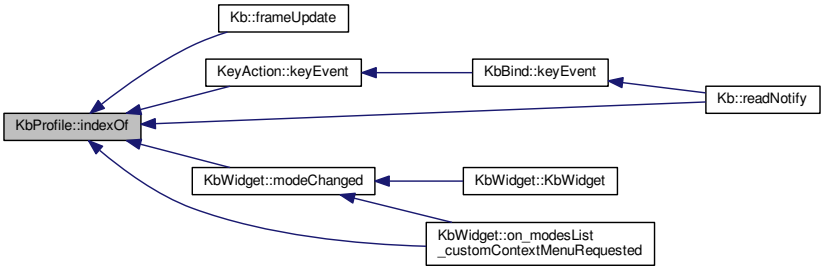
Definition at line 50 of file kbprofile.h.

References _modes.

Referenced by Kb::frameUpdate(), KeyAction::keyEvent(), KbWidget::modeChanged(), KbWidget::on_modesList_customContextMenuRequested(), and Kb::readNotify().

```
50 { return _modes.indexOf(mode); }
```

Here is the caller graph for this function:



12.36.4.8 void KbProfile::insert (int *index*, KbMode * *newMode*) [inline]

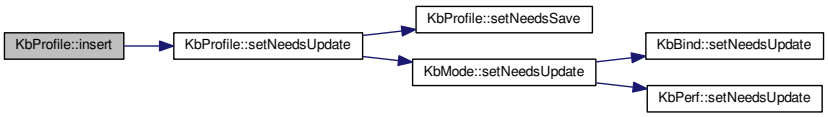
Definition at line 45 of file kbprofile.h.

References _modes, and setNeedsUpdate().

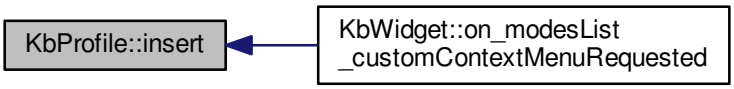
Referenced by KbWidget::on_modesList_customContextMenuRequested().

```
45 { setNeedsUpdate(); _modes.insert(index, newMode); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.4.9 const KeyMap& KbProfile::keyMap ( ) const [inline]

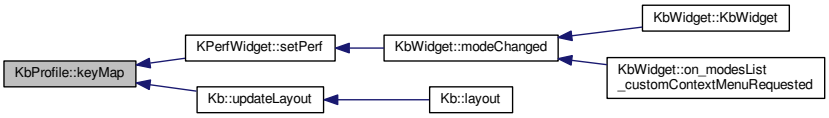
Definition at line 37 of file kbprofile.h.

References _keyMap.

Referenced by KPerfWidget::setPerf(), and Kb::updateLayout().

```
37 { return _keyMap; }
```

Here is the caller graph for this function:



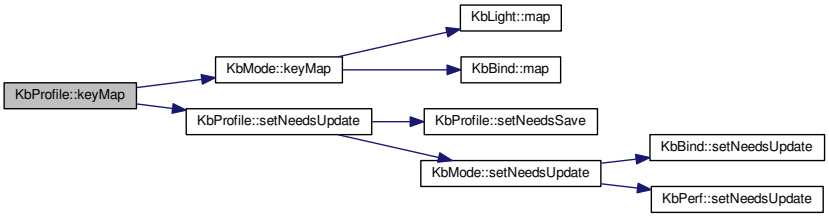
12.36.4.10 void KbProfile::keyMap (const KeyMap & *newKeyMap*)

Definition at line 85 of file kbprofile.cpp.

References _keyMap, _modes, KbMode::keyMap(), and setNeedsUpdate().

```
85                                          {
86     _keyMap = newKeyMap;
87     foreach(KbMode* mode, _modes)
88         mode->keyMap(newKeyMap);
89     setNeedsUpdate();
90 }
```

Here is the call graph for this function:



12.36.4.11 int KbProfile::modeCount ( ) const [inline]

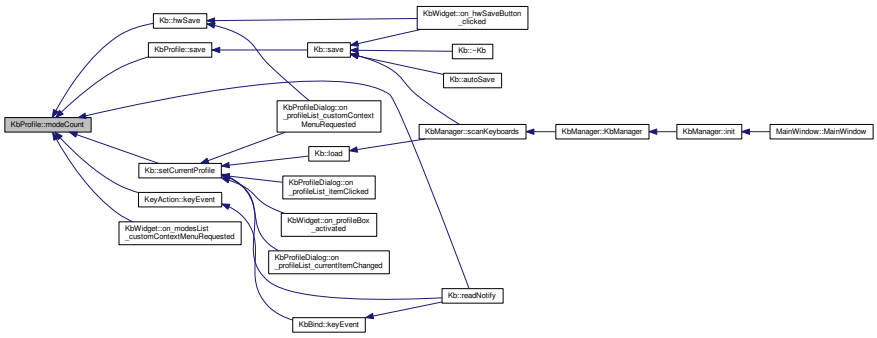
Definition at line 49 of file kbprofile.h.

References _modes.

Referenced by Kb::hwSave(), KeyAction::keyEvent(), KbWidget::on_modesList_customContextMenuRequested(), Kb::readNotify(), save(), and Kb::setCurrentProfile().

```
49 { return _modes.count(); }
```

Here is the caller graph for this function:



12.36.4.12 const ModeList& KbProfile::modes ( ) const [inline]

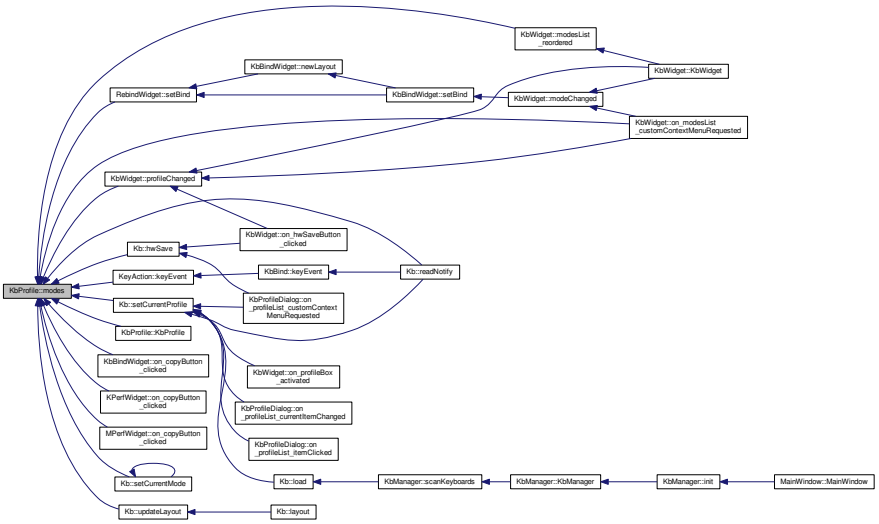
Definition at line 42 of file kbprofile.h.

References _modes.

Referenced by Kb::hwSave(), KbProfile(), KeyAction::keyEvent(), KbWidget::modesList_reordered(), KbBindWidget::on_copyButton_clicked(), KPerfWidget::on_copyButton_clicked(), MPerfWidget::on_copyButton_clicked(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::profileChanged(), Kb::readNotify(), RebindWidget::setBind(), Kb::setCurrentMode(), Kb::setCurrentProfile(), and Kb::updateLayout().

```
42 { return _modes; }
```

Here is the caller graph for this function:



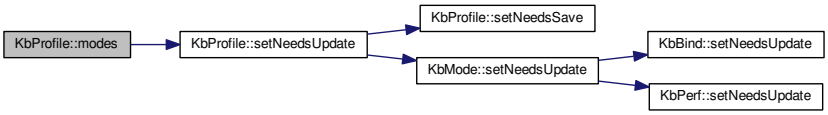
12.36.4.13 void KbProfile::modes (const QList< KbMode * > & *newModes*) [inline]

Definition at line 43 of file kbprofile.h.

References _modes, and setNeedsUpdate().

```
43 { setNeedsUpdate(); _modes = newModes; }
```

Here is the call graph for this function:



12.36.4.14 void KbProfile::move (int *from*, int *to*) [inline]

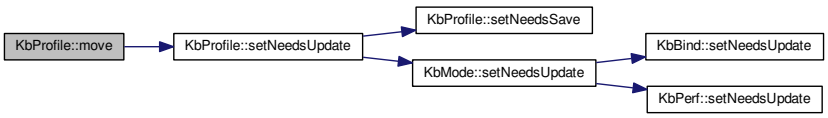
Definition at line 47 of file kbprofile.h.

References _modes, and setNeedsUpdate().

Referenced by Kb::readNotify().

```
47 { setNeedsUpdate(); _modes.move(from, to); }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.36.4.15 QString KbProfile::name ( ) const [inline]

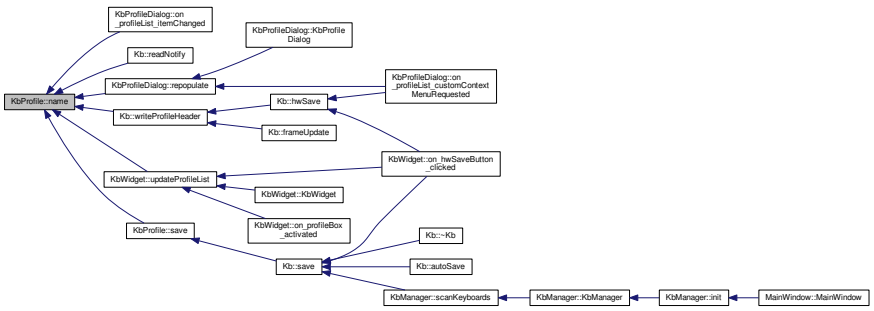
Definition at line 28 of file kbprofile.h.

References _name.

Referenced by KbProfileDialog::on_profileList_itemChanged(), Kb::readNotify(), KbProfileDialog::repopulate(), save(), KbWidget::updateProfileList(), and Kb::writeProfileHeader().

```
28 { return _name; }
```

Here is the caller graph for this function:



12.36.4.16 void KbProfile::name (const QString & *newName*) [inline]

Definition at line 29 of file kbprofile.h.

References _name, and _needsSave.

```
29 { _needsSave = true; _name = newName.trimmed(); if(_name == "")
      _name = "Unnamed"; }
```

12.36.4.17 bool KbProfile::needsSave ( ) const

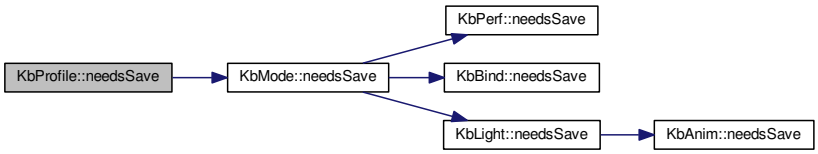
Definition at line 68 of file kbprofile.cpp.

References _modes, _needsSave, and KbMode::needsSave().

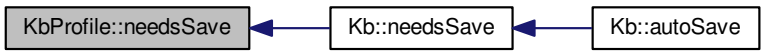
Referenced by Kb::needsSave().

```
68                                  {
69     if(_needsSave)
70         return true;
71     foreach(KbMode* mode, _modes){
72         if(mode->needsSave())
73             return true;
74     }
75     return false;
76 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.4.18 void KbProfile::newId ( )

Definition at line 78 of file kbprofile.cpp.

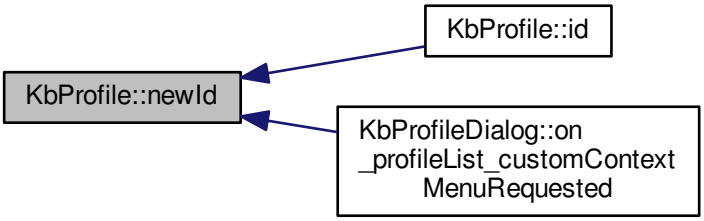
References _id, _modes, _needsSave, and KbMode::newId().

Referenced by id(), and KbProfileDialog::on_profileList_customContextMenuRequested().

```
78                         {
79     _needsSave = true;
80     _id = UsbId();
81     foreach(KbMode* mode, _modes)
82         mode->newId();
83 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.36.4.19 void KbProfile::removeAll (KbMode * *mode*) [inline]

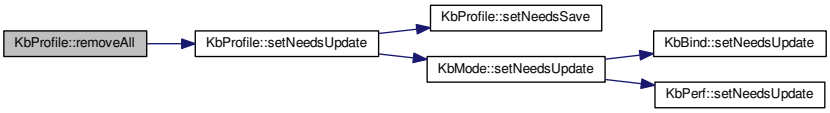
Definition at line 46 of file kbprofile.h.

References _modes, and setNeedsUpdate().

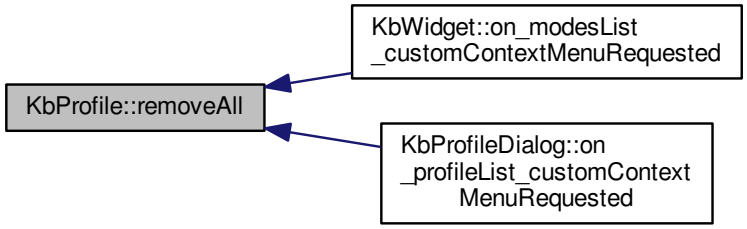
Referenced by KbWidget::on_modesList_customContextMenuRequested(), and KbProfileDialog::on_profileList_-customContextMenuRequested().

```
46 { setNeedsUpdate(); _modes.removeAll(mode); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.4.20 void KbProfile::save (CkbSettings & *settings*)

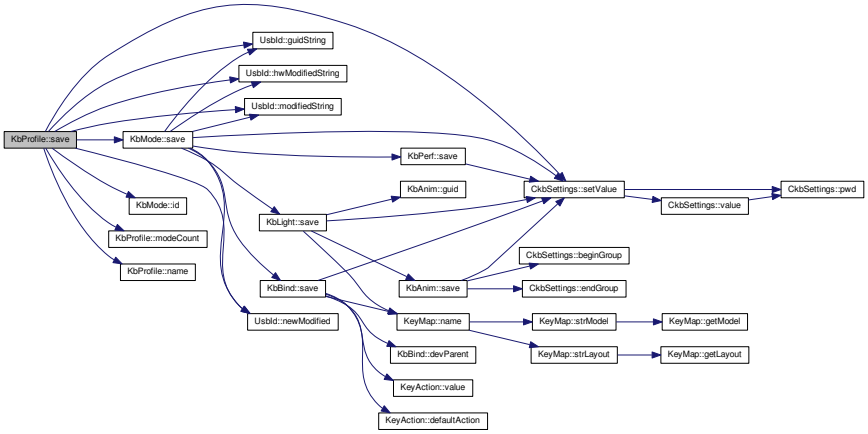
Definition at line 48 of file kbprofile.cpp.

References _currentMode, _id, _modes, _needsSave, UsbId::guidString(), UsbId::hwModifiedString(), KbMode::id(), modeCount(), UsbId::modifiedString(), name(), UsbId::newModified(), KbMode::save(), and CkbSettings::setValue().

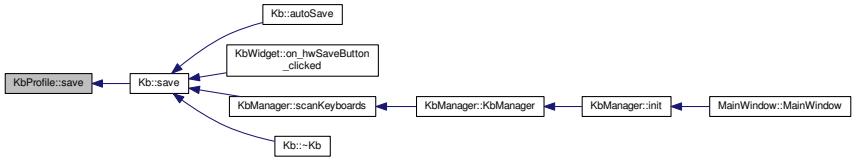
Referenced by Kb::save().

```
48                                          {
49     _needsSave = false;
50     _id.newModified();
51     // Save data to preferences
52     SGroup group(settings, id().guidString());
53     settings.setValue("Name", name());
54     settings.setValue("Modified", _id.modifiedString());
55     settings.setValue("HwModified", _id.hwModifiedString());
56     if(_currentMode)
57         settings.setValue("CurrentMode", _currentMode->id().
      guidString());
58     // Save modes
59     uint count = modeCount();
60     settings.setValue("ModeCount", count);
61     for(uint i = 0; i < count; i++){
62         SGroup group(settings, QString::number(i));
63         KbMode* mode = _modes.at(i);
64         mode->save(settings);
65     }
66 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.4.21 void KbProfile::setNeedsSave ( ) [inline]

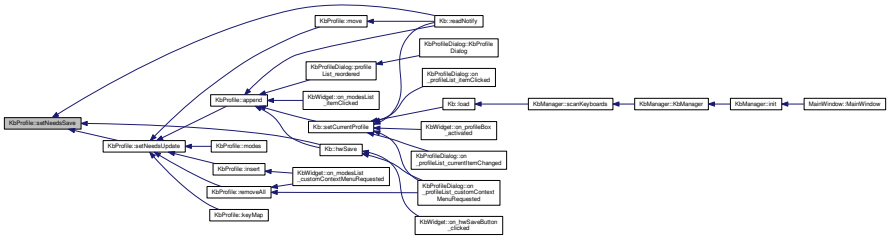
Definition at line 25 of file kbprofile.h.

References _needsSave.

Referenced by Kb::hwSave(), Kb::readNotify(), and setNeedsUpdate().

```
25 { _needsSave = true; }
```

Here is the caller graph for this function:



12.36.4.22 void KbProfile::setNeedsUpdate ( ) [inline], [private]

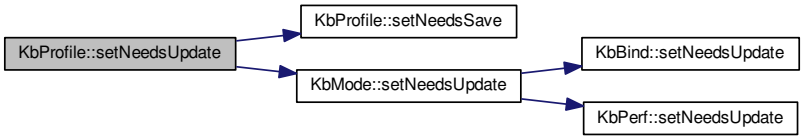
Definition at line 66 of file kbprofile.h.

References _modes, setNeedsSave(), and KbMode::setNeedsUpdate().

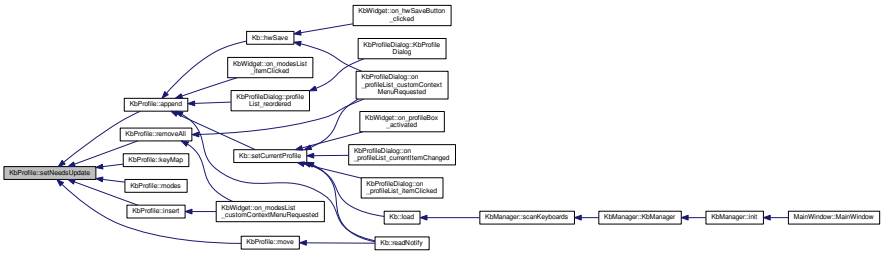
Referenced by append(), insert(), keyMap(), modes(), move(), and removeAll().

```
66 { setNeedsSave(); foreach(KbMode* mode, _modes){ mode->
     setNeedsUpdate(); } }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.36.5 Field Documentation

12.36.5.1 KbMode* KbProfile::_currentMode [private]

Definition at line 58 of file kbprofile.h.

Referenced by currentMode(), KbProfile(), and save().

12.36.5.2 UsbId KbProfile::_id [private]

Definition at line 60 of file kbprofile.h.

Referenced by id(), KbProfile(), newId(), and save().

12.36.5.3 KeyMap KbProfile::_keyMap [private]

Definition at line 61 of file kbprofile.h.

Referenced by KbProfile(), and keyMap().

12.36.5.4 ModeList KbProfile::_modes [private]

Definition at line 62 of file kbprofile.h.

Referenced by append(), find(), indexOf(), insert(), KbProfile(), keyMap(), modeCount(), modes(), move(), needsSave(), newId(), removeAll(), save(), and setNeedsUpdate().

12.36.5.5 QString KbProfile::_name [private]

Definition at line 59 of file kbprofile.h.

Referenced by KbProfile(), and name().

12.36.5.6 bool KbProfile::_needsSave [private]

Definition at line 63 of file kbprofile.h.

Referenced by currentMode(), id(), name(), needsSave(), newId(), save(), and setNeedsSave().

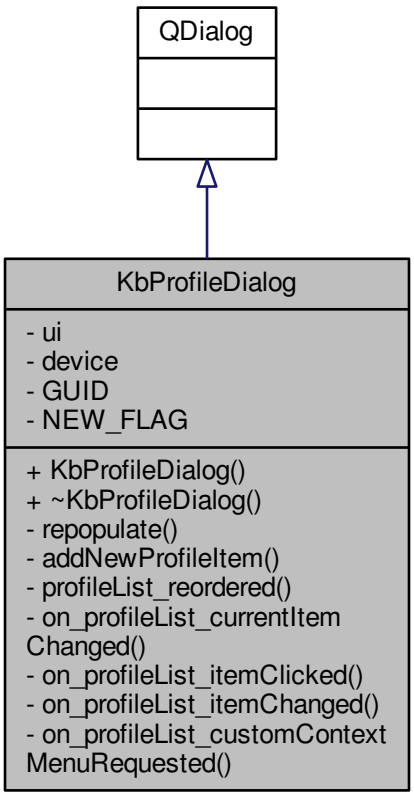
The documentation for this class was generated from the following files:

- src/ckb/kbprofile.h
- src/ckb/kbprofile.cpp

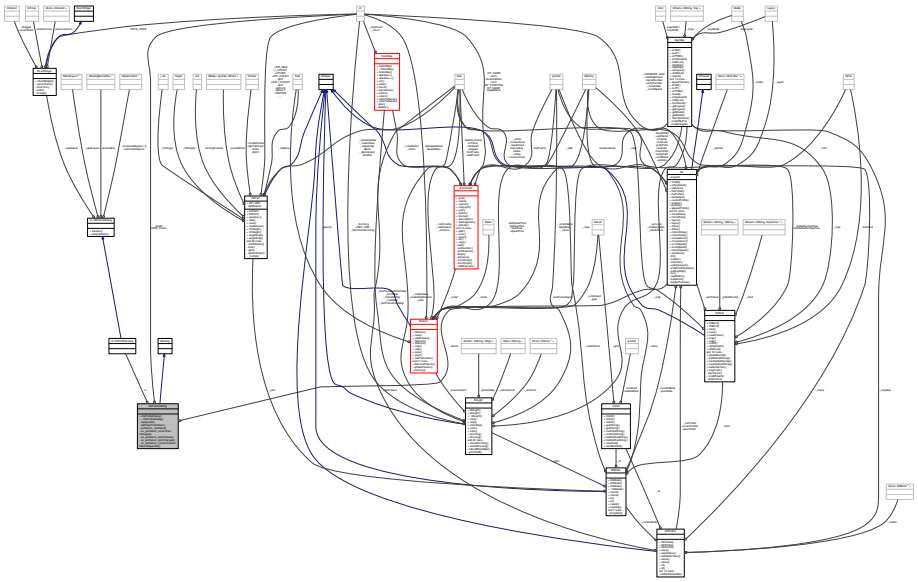
12.37 KbProfileDialog Class Reference

`#include <src/ckb/kbprofiledialog.h>`

Inheritance diagram for KbProfileDialog:



Collaboration diagram for KbProfileDialog:



Public Member Functions

- KbProfileDialog (KbWidget *parent=0)
- ~KbProfileDialog ()

Private Slots

- void profileList_reordered ()
- void on_profileList_currentItemChanged (QListWidgetItem *current, QListWidgetItem *previous)
- void on_profileList_itemClicked (QListWidgetItem *item)
- void on_profileList_itemChanged (QListWidgetItem *item)
- void on_profileList_customContextMenuRequested (const QPoint &pos)

Private Member Functions

- void repopulate ()
- void addNewProfileItem ()

Private Attributes

- Ui::KbProfileDialog * ui
- Kb * device

Static Private Attributes

- static const int GUID = Qt::UserRole
- static const int NEW_FLAG = Qt::UserRole + 1

12.37.1 Detailed Description

Definition at line 11 of file kbprofiledialog.h.

12.37.2 Constructor & Destructor Documentation

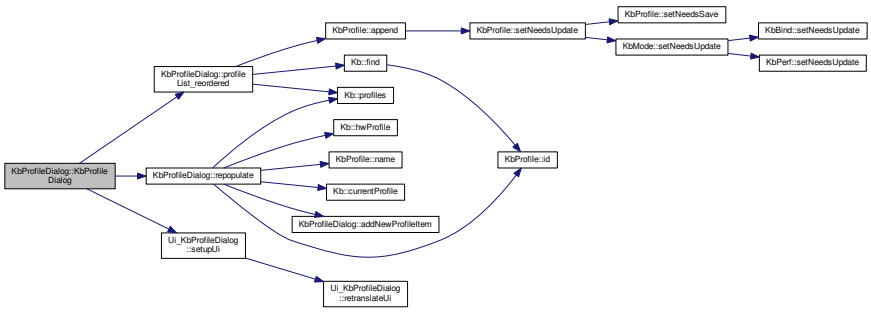
12.37.2.1 KbProfileDialog::KbProfileDialog (KbWidget * *parent* = 0) [explicit]

Definition at line 5 of file kbprofiledialog.cpp.

References Ui_KbProfileDialog::profileList, profileList_reordered(), repopulate(), Ui_KbProfileDialog::setupUi(), and ui.

```
5                                                :
6     QDialog(parent),
7     ui(new Ui::KbProfileDialog), device(parent->
      device)
8 {
9     ui->setupUi(this);
10     connect(ui->profileList, SIGNAL(orderChanged()), this, SLOT(
      profileList_reordered()));
11
12     // Populate profile list
13     repopulate();
14 }
```

Here is the call graph for this function:



12.37.2.2 KbProfileDialog::~KbProfileDialog ( )

Definition at line 16 of file kbprofiledialog.cpp.

References ui.

```
16                                    {
17     delete ui;
18 }
```

12.37.3 Member Function Documentation

12.37.3.1 void KbProfileDialog::addNewProfileItem ( ) [private]

Definition at line 66 of file kbprofiledialog.cpp.

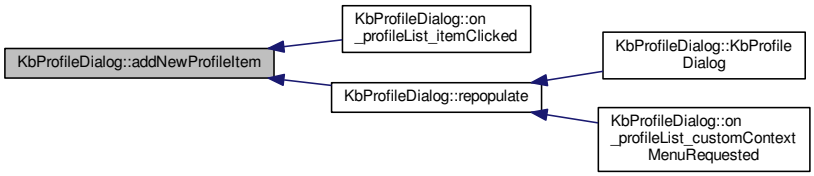
References NEW_FLAG, Ui_KbProfileDialog::profileList, and ui.

Referenced by on_profileList_itemClicked(), and repopulate().

```
66                                        {
67     // Add an item for creating a new profile. Make it editable but not dragable.
68     QListWidgetItem* item = new QListWidgetItem("New profile...", ui->
      profileList);
69     item->setFlags((item->flags() | Qt::ItemIsEditable) & ~Qt::ItemIsDragEnabled & ~Qt::ItemIsDropEnabled);
70     item->setData(NEW_FLAG, 1);
```

```
71     QFont font = item->font();
72     font.setItalic(true);
73     item->setFont(font);
74     item->setIcon(QIcon(":/img/icon_plus.png"));
75     ui->profileList->addItem(item);
76 }
```

Here is the caller graph for this function:



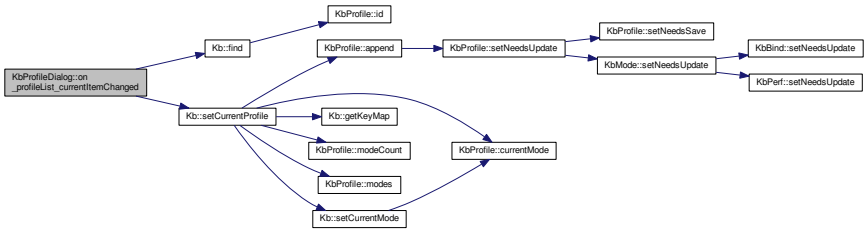
12.37.3.2 void KbProfileDialog::on_profileList_currentItemChanged (QListWidgetItem * *current*, QListWidgetItem * *previous*) [private], [slot]

Definition at line 39 of file kbprofiledialog.cpp.

References device, Kb::find(), GUID, and Kb::setCurrentProfile().

```
39                                                                                                                  {
40     if(!current)
41         return;
42     KbProfile* profile = device->find(current->data(
       GUID).toUuid());
43     if(!profile)
44         return;
45     device->setCurrentProfile(profile);
46 }
```

Here is the call graph for this function:



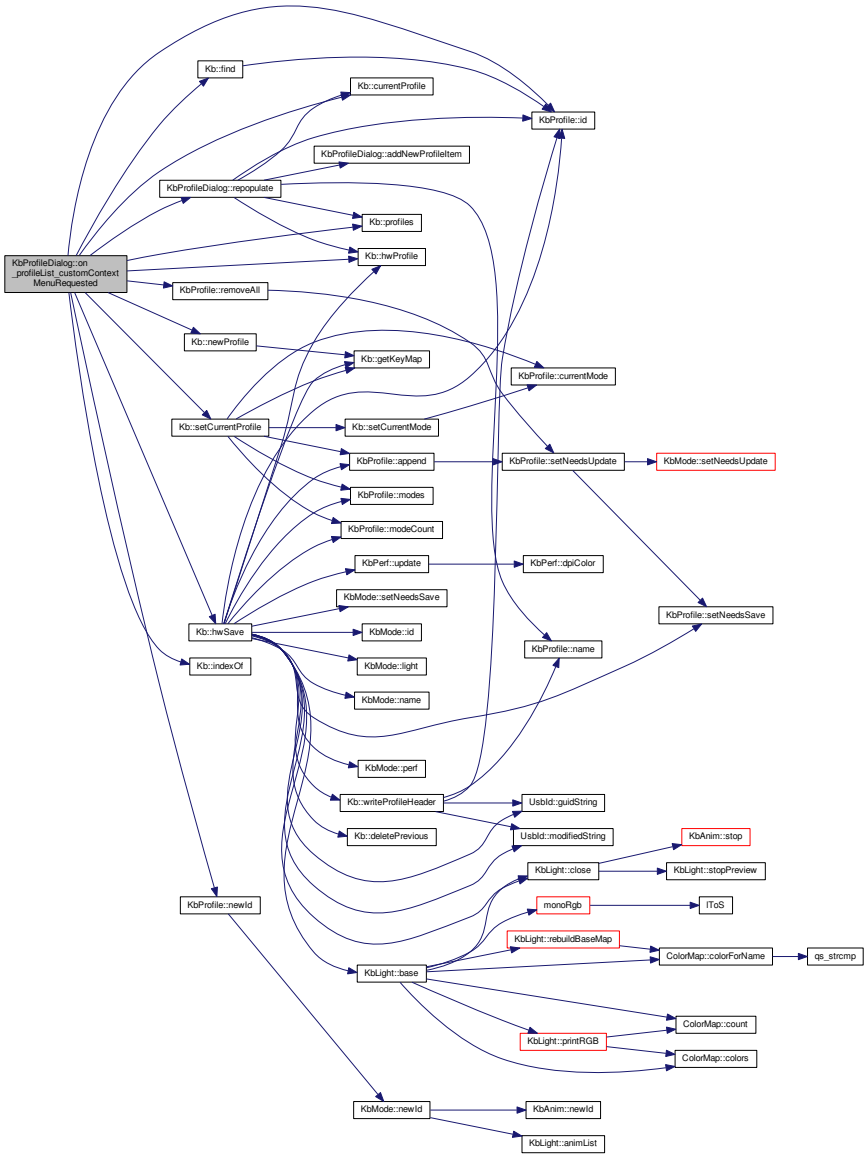
12.37.3.3 void KbProfileDialog::on_profileList_customContextMenuRequested (const QPoint & *pos*) [private], [slot]

Definition at line 109 of file kbprofiledialog.cpp.

References Kb::currentProfile(), device, Kb::find(), UsbId::guid, GUID, Kb::hwProfile(), Kb::hwSave(), KbProfile::id(), Kb::indexOf(), NEW_FLAG, KbProfile::newId(), Kb::newProfile(), Ui_KbProfileDialog::profileList, Kb::profiles(), KbProfile::removeAll(), repopulate(), Kb::setCurrentProfile(), and ui.

```
109                                                                                 {
110     QListWidgetItem* item = ui->profileList->itemAt(pos);
111     KbProfile* currentProfile = device->currentProfile();
112     if(!item || !currentProfile || item->data(GUID).toUuid() != currentProfile->
    id().guid)
113         return;
114     int index = device->indexOf(currentProfile);
115     QList<KbProfile*> profiles = device->profiles();
116 
117     QMenu menu(this);
118     QAction* rename = new QAction("Rename...", this);
119     QAction* duplicate = new QAction("Duplicate", this);
120     QAction* del = new QAction("Delete", this);
121     bool canDelete = (profiles.count() > 1);
122     if(!canDelete)
123         // Can't delete the last profile on the device
124         del->setEnabled(false);
125     QAction* hwsave = new QAction("Save to Hardware", this);
126     QAction* moveup = new QAction("Move Up", this);
127     if(index == 0)
128         moveup->setEnabled(false);
129     QAction* movedown = new QAction("Move Down", this);
130     if(index >= profiles.count() - 1)
131         movedown->setEnabled(false);
132     menu.addAction(rename);
133     menu.addAction(duplicate);
134     menu.addAction(del);
135     menu.addSeparator();
136     menu.addAction(hwsave);
137     menu.addSeparator();
138     menu.addAction(moveup);
139     menu.addAction(movedown);
140     QAction* result = menu.exec(QCursor::pos());
141     if(result == rename){
142         ui->profileList->editItem(item);
143     } else if(result == duplicate){
144         KbProfile* newProfile = device->newProfile(currentProfile);
145         newProfile->newId();
146         profiles.insert(index + 1, newProfile);
147         device->profiles(profiles);
148         device->setCurrentProfile(newProfile);
149     } else if(result == del){
150         if(!canDelete)
151             return;
152         profiles.removeAll(currentProfile);
153         device->profiles(profiles);
154         currentProfile->deleteLater();
155         // Select next profile
156         if(index < profiles.count())
157             device->setCurrentProfile(profiles.at(index));
158         else
159             device->setCurrentProfile(profiles.last());
160     } else if(result == hwsave){
161         device->hwSave();
162         // Refresh item icons
163         int count = ui->profileList->count();
164         for(int i = 0; i < count; i++){
165             QListWidgetItem* item = ui->profileList->item(i);
166             if(item->data(NEW_FLAG).toUInt() == 1)
167                 continue;
168             KbProfile* profile = device->find(item->data(GUID).toUuid());
169             item->setIcon(QIcon((profile == device->hwProfile()) ? "
    :/img/icon_profile_hardware.png" : ":/img/icon_profile.png"));
170         }
171     } else if(result == moveup){
172         profiles.removeAll(currentProfile);
173         profiles.insert(index - 1, currentProfile);
174         device->profiles(profiles);
175     } else if(result == movedown){
176         profiles.removeAll(currentProfile);
177         profiles.insert(index + 1, currentProfile);
178         device->profiles(profiles);
179     }
180     repopulate();
181 }
```

Here is the call graph for this function:



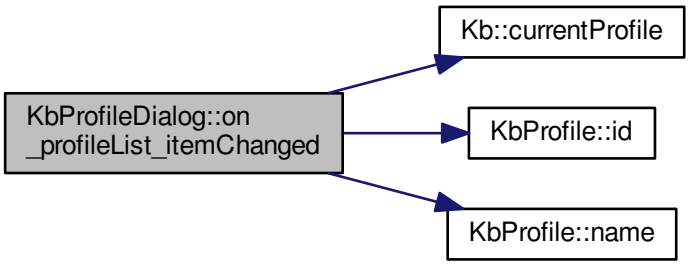
12.37.3.4 void KbProfileDialog::on_profileList_itemChanged (QListWidgetItem * *item*) [private], [slot]

Definition at line 100 of file kbprofiledialog.cpp.

References Kb::currentProfile(), device, UsbId::guid, GUID, KbProfile::id(), and KbProfile::name().

```
100                                                                 {
101     KbProfile* currentProfile = device->currentProfile();
102     if(!item || !currentProfile || item->data(GUID).toUuid() != currentProfile->
      id().guid)
103         return;
104     currentProfile->name(item->text());
105     // Set the text to the actual name (trimmed, "" replaced with "Unnamed")
106     item->setText(currentProfile->name());
107 }
```

Here is the call graph for this function:



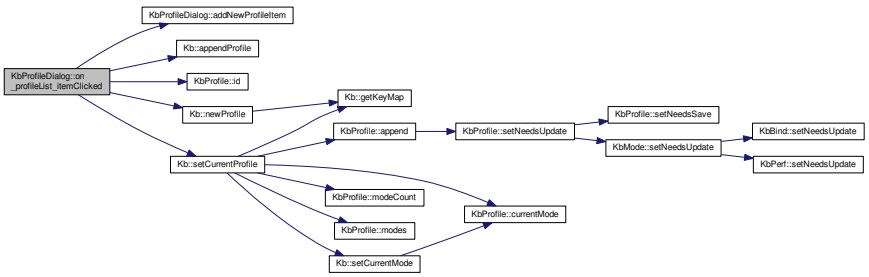
12.37.3.5 void KbProfileDialog::on_profileList_itemClicked (QListWidgetItem * *item*) [private], [slot]

Definition at line 78 of file kbprofiledialog.cpp.

References addNewProfileItem(), Kb::appendProfile(), device, UsbId::guid, GUID, KbProfile::id(), NEW_FLAG, Kb::newProfile(), Ui_KbProfileDialog::profileList, Kb::setCurrentProfile(), and ui.

```
78                                                                  {
79     QUuid guid = item->data(GUID).toUuid();
80     if(guid.isNull() && item->data(NEW_FLAG).toInt() == 1){
81         // New profile
82         item->setText("");
83         ui->profileList->editItem(item);
84         item->setFlags(item->flags() | Qt::ItemIsEditable | Qt::ItemIsDragEnabled | Qt::ItemIsDropEnabled);
85         QFont font = item->font();
86         font.setItalic(false);
87         item->setFont(font);
88         item->setIcon(QIcon(":/img/icon_profile.png"));
89         // Add the new profile and assign it to this item
90         KbProfile* newProfile = device->newProfile();
91         device->appendProfile(newProfile);
92         item->setData(GUID, newProfile->id().guid);
93         item->setData(NEW_FLAG, 0);
94         device->setCurrentProfile(newProfile);
95         // Create another "new profile" item to replace this one
96         addNewProfileItem();
97     }
98 }
```

Here is the call graph for this function:



12.37.3.6 void KbProfileDialog::profileList_reordered ( ) [private],[slot]

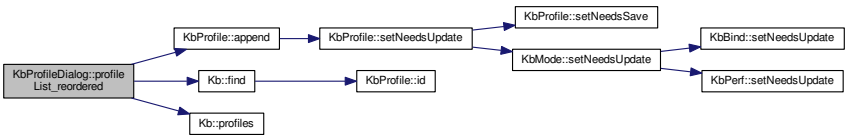
Definition at line 20 of file kbprofiledialog.cpp.

References KbProfile::append(), device, Kb::find(), GUID, Ui_KbProfileDialog::profileList, Kb::profiles(), and ui.

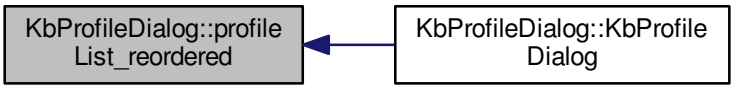
Referenced by KbProfileDialog().

```
20                                          {
21     // Rebuild profile list from items
22     QList<KbProfile*> newProfiles;
23     int count = ui->profileList->count();
24     for(int i = 0; i < count; i++){
25         QListWidgetItem* item = ui->profileList->item(i);
26         KbProfile* profile = device->find(item->data(GUID).toUuid());
27         if(profile && !newProfiles.contains(profile))
28             newProfiles.append(profile);
29         item->setFlags(item->flags() | Qt::ItemIsEditable);
30     }
31     // Add any missing profiles at the end of the list
32     foreach(KbProfile* profile, device->profiles()){
33         if(!newProfiles.contains(profile))
34             newProfiles.append(profile);
35     }
36     device->profiles(newProfiles);
37 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.37.3.7 void KbProfileDialog::repopulate ( ) [private]

Definition at line 48 of file kbprofiledialog.cpp.

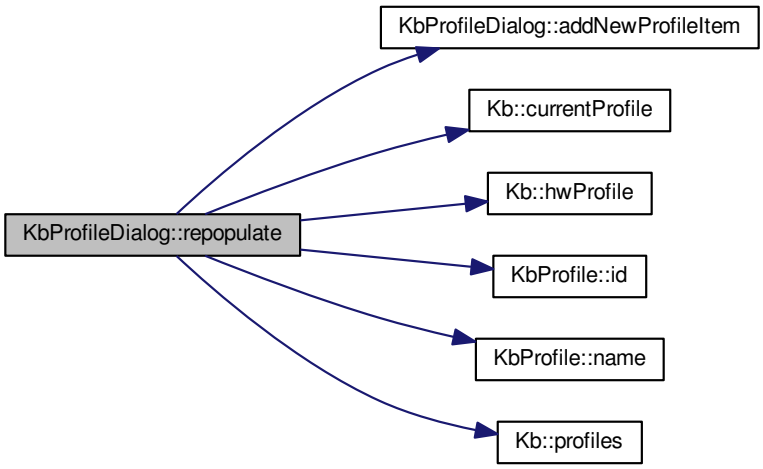
References addNewProfileItem(), current, Kb::currentProfile(), device, UsbId::guid, GUID, Kb::hwProfile(), KbProfile::id(), KbProfile::name(), Ui_KbProfileDialog::profileList, Kb::profiles(), and ui.

Referenced by KbProfileDialog(), and on_profileList_customContextMenuRequested().

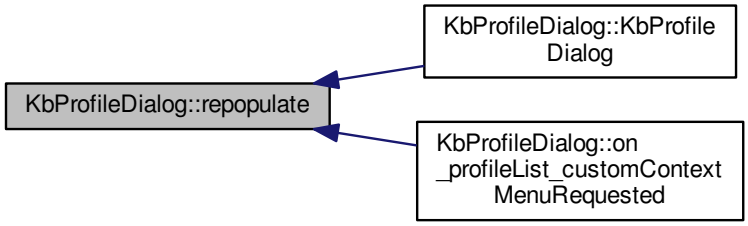
```
48                                  {
49     ui->profileList->clear();
50     QListWidgetItem* current = 0;
51     foreach(KbProfile* profile, device->profiles()){
52         QListWidgetItem* item = new QListWidgetItem(QIcon((profile == device->
      hwProfile()) ? ":/img/icon_profile_hardware.png" : ":/img/icon_profile.png"), profile->
```

```
      name(), ui->profileList);
53         item->setData(GUID, profile->id().guid);
54         item->setFlags(item->flags() | Qt::ItemIsEditable);
55         if(profile == device->currentProfile()){
56             item->setSelected(true);
57             current = item;
58         }
59         ui->profileList->addItem(item);
60     }
61     if(current)
62         ui->profileList->setCurrentItem(current);
63     addNewProfileItem();
64 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.37.4 Field Documentation

12.37.4.1 Kb* KbProfileDialog::device [private]

Definition at line 29 of file kbprofiledialog.h.

Referenced by on_profileList_currentItemChanged(), on_profileList_customContextMenuRequested(), on_profileList_itemChanged(), on_profileList_itemClicked(), profileList_reordered(), and repopulate().

12.37.4.2 const int KbProfileDialog::GUID = Qt::UserRole [static], [private]

Definition at line 30 of file kbprofiledialog.h.

Referenced by on_profileList_currentItemChanged(), on_profileList_customContextMenuRequested(), on_profileList_itemChanged(), on_profileList_itemClicked(), profileList_reordered(), and repopulate().

12.37.4.3 const int KbProfileDialog::NEW_FLAG = Qt::UserRole + 1 [static], [private]

Definition at line 31 of file kbprofiledialog.h.

Referenced by addNewProfileItem(), on_profileList_customContextMenuRequested(), and on_profileList_itemClicked().

12.37.4.4 Ui::KbProfileDialog* KbProfileDialog::ui [private]

Definition at line 27 of file kbprofiledialog.h.

Referenced by addNewProfileItem(), KbProfileDialog(), on_profileList_customContextMenuRequested(), on_profileList_itemClicked(), profileList_reordered(), repopulate(), and ~KbProfileDialog().

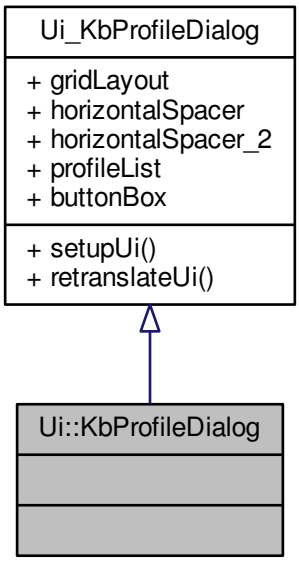
The documentation for this class was generated from the following files:

- src/ckb/kbprofiledialog.h
- src/ckb/kbprofiledialog.cpp

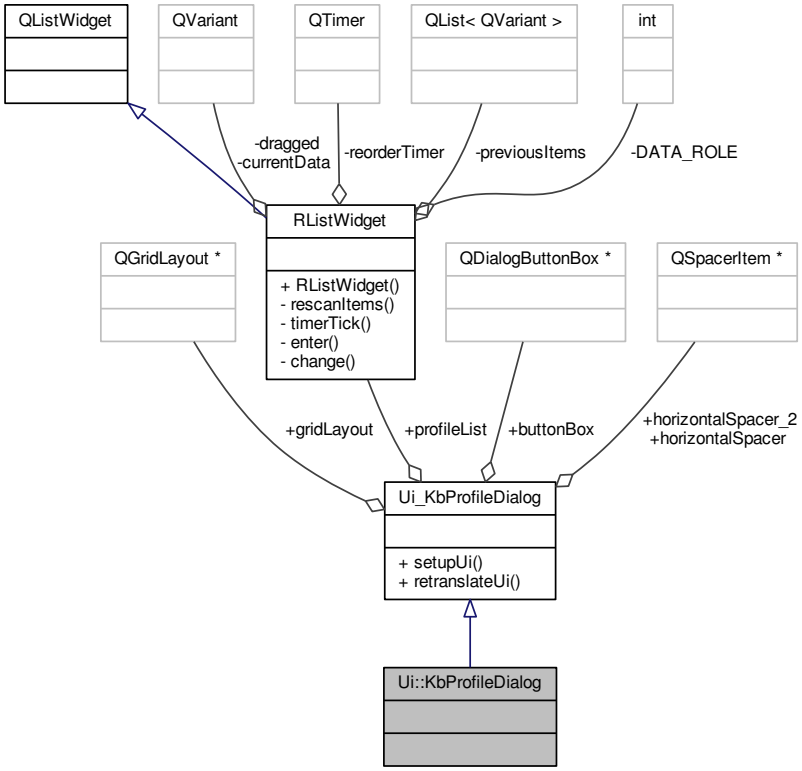
12.38 Ui::KbProfileDialog Class Reference

#include <src/ckb/ui_kbprofiledialog.h>

Inheritance diagram for Ui::KbProfileDialog:



Collaboration diagram for Ui::KbProfileDialog:



Public Member Functions

- void setupUi (QDialog *KbProfileDialog)
- void retranslateUi (QDialog *KbProfileDialog)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer
- QSpacerItem * horizontalSpacer_2
- RListWidget * profileList
- QDialogButtonBox * buttonBox

12.38.1 Detailed Description

Definition at line 81 of file ui_kbprofiledialog.h.

12.38.2 Member Function Documentation

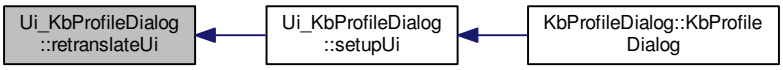
12.38.2.1 void Ui_KbProfileDialog::retranslateUi (QDialog * *KbProfileDialog*) [inline], [inherited]

Definition at line 73 of file ui_kbprofiledialog.h.

Referenced by Ui_KbProfileDialog::setupUi().

```
74     {
75         KbProfileDialog->setWindowTitle(QApplication::translate("KbProfileDialog", "Keyboard Profiles",
       Q_NULLPTR));
76     } // retranslateUi
```

Here is the caller graph for this function:



12.38.2.2 void Ui_KbProfileDialog::setupUi (QDialog * *KbProfileDialog*) [inline],[inherited]

Definition at line 34 of file ui_kbprofiledialog.h.

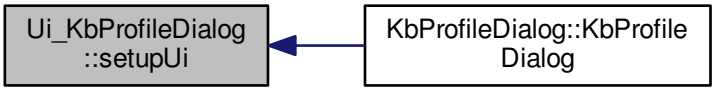
References Ui_KbProfileDialog::buttonBox, Ui_KbProfileDialog::gridLayout, Ui_KbProfileDialog::horizontalSpacer, Ui_KbProfileDialog::horizontalSpacer_2, Ui_KbProfileDialog::profileList, and Ui_KbProfileDialog::retranslateUi().

Referenced by KbProfileDialog::KbProfileDialog().

```
35     {
36         if (KbProfileDialog->objectName().isEmpty())
37             KbProfileDialog->setObjectName(QStringLiteral("KbProfileDialog"));
38         KbProfileDialog->resize(400, 300);
39         KbProfileDialog->setFocusPolicy(Qt::ClickFocus);
40         gridLayout = new QGridLayout(KbProfileDialog);
41         gridLayout->setObjectName(QStringLiteral("gridLayout"));
42         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
       QSizePolicy::Minimum);
43
44         gridLayout->addItem(horizontalSpacer, 0, 0, 1, 1);
45
46         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
       QSizePolicy::Minimum);
47
48         gridLayout->addItem(horizontalSpacer_2, 0, 2, 1, 1);
49
50         profileList = new RListWidget(KbProfileDialog);
51         profileList->setObjectName(QStringLiteral("profileList"));
52         profileList->setContextMenuPolicy(Qt::CustomContextMenu);
53         profileList->setFrameShadow(QFrame::Plain);
54         profileList->setIconSize(QSize(24, 24));
55
56         gridLayout->addWidget(profileList, 0, 1, 1, 1);
57
58         buttonBox = new QDialogButtonBox(KbProfileDialog);
59         buttonBox->setObjectName(QStringLiteral("buttonBox"));
60         buttonBox->setOrientation(Qt::Horizontal);
61         buttonBox->setStandardButtons(QDialogButtonBox::Ok);
62
63         gridLayout->addWidget(buttonBox, 1, 0, 1, 3);
64
65
66         retranslateUi(KbProfileDialog);
67         QObject::connect(buttonBox, SIGNAL(accepted()), KbProfileDialog, SLOT(accept()));
68         QObject::connect(buttonBox, SIGNAL(rejected()), KbProfileDialog, SLOT(reject()));
69
70         QMetaObject::connectSlotsByName(KbProfileDialog);
71     } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.38.3 Field Documentation

12.38.3.1 QDialogButtonBox* Ui_KbProfileDialog::buttonBox [inherited]

Definition at line 32 of file ui_kbprofiledialog.h.

Referenced by Ui_KbProfileDialog::setupUi().

12.38.3.2 QGridLayout* Ui_KbProfileDialog::gridLayout [inherited]

Definition at line 28 of file ui_kbprofiledialog.h.

Referenced by Ui_KbProfileDialog::setupUi().

12.38.3.3 QSpacerItem* Ui_KbProfileDialog::horizontalSpacer [inherited]

Definition at line 29 of file ui_kbprofiledialog.h.

Referenced by Ui_KbProfileDialog::setupUi().

12.38.3.4 QSpacerItem* Ui_KbProfileDialog::horizontalSpacer_2 [inherited]

Definition at line 30 of file ui_kbprofiledialog.h.

Referenced by Ui_KbProfileDialog::setupUi().

12.38.3.5 RListWidget* Ui_KbProfileDialog::profileList [inherited]

Definition at line 31 of file ui_kbprofiledialog.h.

Referenced by KbProfileDialog::addNewProfileItem(), KbProfileDialog::KbProfileDialog(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_itemClicked(), KbProfileDialog::profileList_reordered(), KbProfileDialog::repopulate(), and Ui_KbProfileDialog::setupUi().

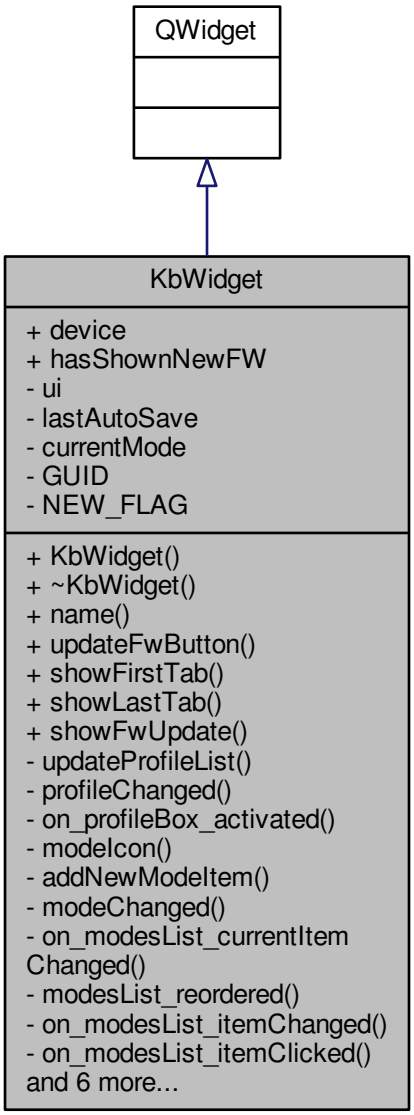
The documentation for this class was generated from the following file:

- src/ckb/ui_kbprofiledialog.h

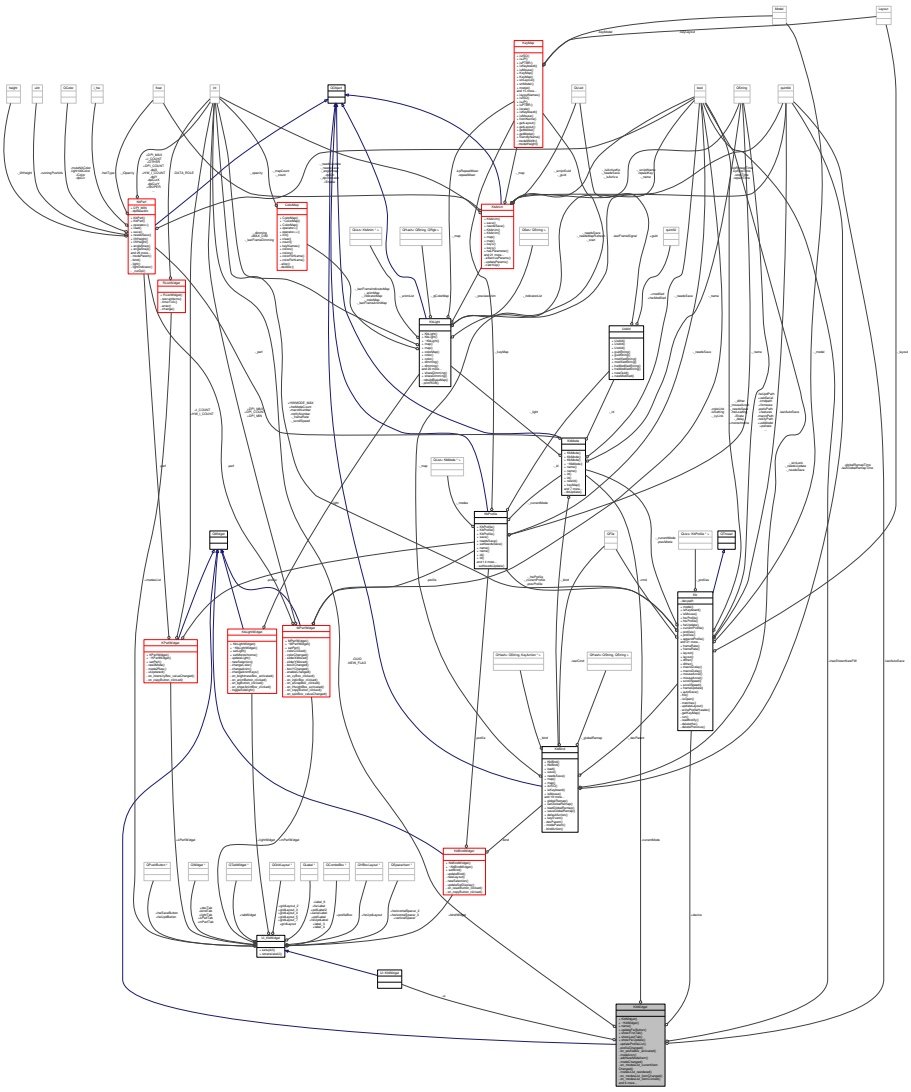
12.39 KbWidget Class Reference

`#include <src/ckb/kbwidget.h>`

Inheritance diagram for KbWidget:



Collaboration diagram for KbWidget:



Public Slots

- void showFirstTab ()
- void showLastTab ()
- void showFwUpdate ()

Public Member Functions

- KbWidget (QWidget *parent, Kb *_device)
- ~KbWidget ()
- QString name () const
- void updateFwButton ()

Data Fields

- Kb * device
- bool hasShownNewFW

Private Slots

- void updateProfileList ()
- void profileChanged ()
- void on_profileBox_activated (int index)
- QIcon modeIcon (int i)
- void addNewModeItem ()
- void modeChanged (bool spontaneous=true)
- void on_modesList_currentItemChanged (QListWidgetItem ∗current, QListWidgetItem ∗previous)
- void modesList_reordered ()
- void on_modesList_itemChanged (QListWidgetItem ∗item)
- void on_modesList_itemClicked (QListWidgetItem ∗item)
- void on_modesList_customContextMenuRequested (const QPoint &pos)

  KbWidget::on_modesList_customContextMenuRequested.
- void devUpdate ()
- void modeUpdate ()
- void on_hwSaveButton_clicked ()
- void on_tabWidget_currentChanged (int index)
- void on_fwUpdButton_clicked ()

Private Attributes

- Ui::KbWidget ∗ ui
- quint64 lastAutoSave
- KbMode ∗ currentMode

Static Private Attributes

- static const int GUID = Qt::UserRole
- static const int NEW_FLAG = Qt::UserRole + 1

12.39.1 Detailed Description

Definition at line 15 of file kbwidget.h.

12.39.2 Constructor & Destructor Documentation

12.39.2.1 KbWidget::KbWidget (QWidget ∗ *parent*, Kb ∗ *_device*) [explicit]

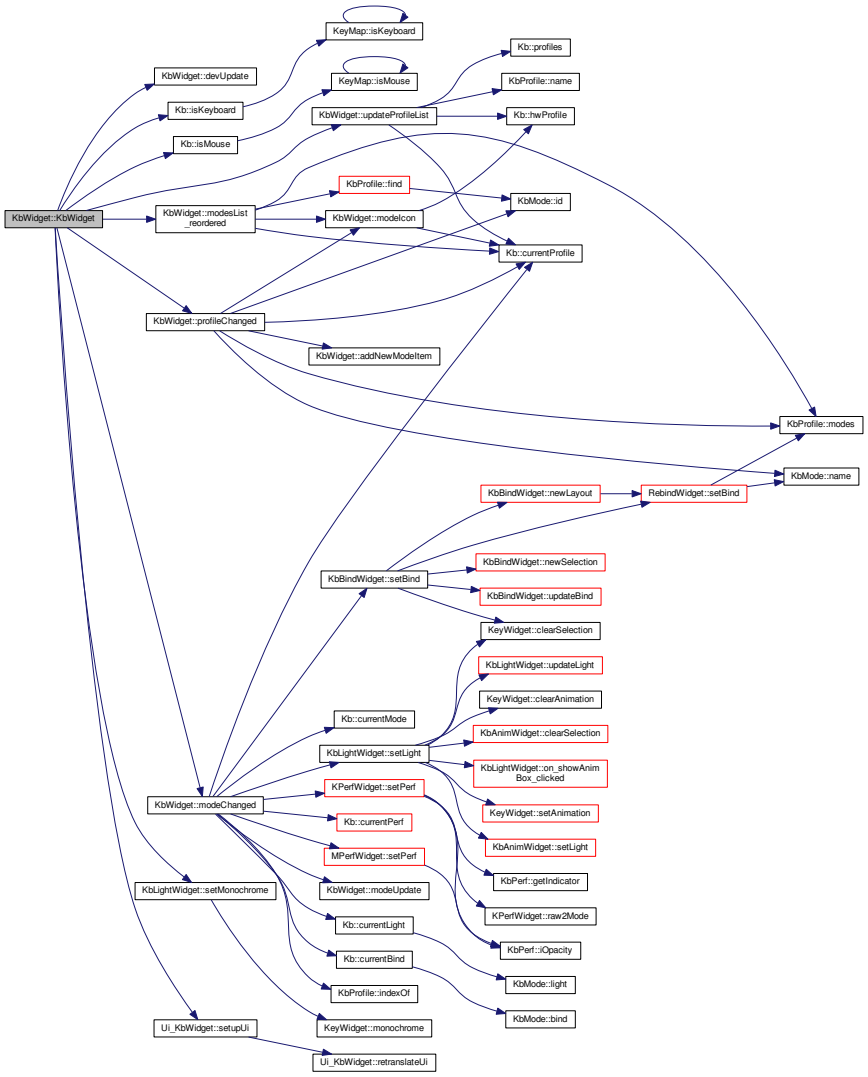
Definition at line 15 of file kbwidget.cpp.

References device, devUpdate(), Kb::features, Ui_KbWidget::fwUpdButton, Ui_KbWidget::fwUpdLabel, Ui_KbWidget::fwUpdLayout, Kb::isKeyboard(), Kb::isMouse(), Ui_KbWidget::kPerfTab, Ui_KbWidget::lightTab, Ui_KbWidget::lightWidget, modeChanged(), Ui_KbWidget::modesList, modesList_reordered(), Kb::monochrome, Ui_KbWidget::mPerfTab, Ui_KbWidget::pollLabel, Ui_KbWidget::pollLabel2, profileChanged(), KbLightWidget::setMonochrome(), Ui_KbWidget::setupUi(), Ui_KbWidget::tabWidget, ui, and updateProfileList().

```
15                                              :
16     QWidget(parent),
17     device(_device), hasShownNewFW(false),
18     ui(new Ui::KbWidget),
19     currentMode(0)
20 {
21     ui->setupUi(this);
22     connect(ui->modesList, SIGNAL(orderChanged()), this, SLOT(
       modesList_reordered()));
23
```

```
24      connect(device, SIGNAL(infoUpdated()), this, SLOT(devUpdate()));
25      connect(device, SIGNAL(profileAdded()), this, SLOT(updateProfileList()));
26      connect(device, SIGNAL(profileChanged()), this, SLOT(
    profileChanged()));
27      connect(device, SIGNAL(profileRenamed()), this, SLOT(updateProfileList()));
28      connect(device, SIGNAL(modeRenamed()), this, SLOT(profileChanged()));
29      connect(device, SIGNAL(modeRenamed()), this, SLOT(modeChanged()));
30      connect(device, SIGNAL(modeChanged(bool)), this, SLOT(
    modeChanged(bool)));
31
32      // Remove the Lighting and Performance tabs from non-RGB keyboards
33      if(!device->features.contains("rgb")){
34          ui->tabWidget->removeTab(ui->tabWidget->indexOf(ui->
    lightTab));
35          ui->tabWidget->removeTab(ui->tabWidget->indexOf(ui->
    kPerfTab));
36          ui->tabWidget->removeTab(ui->tabWidget->indexOf(ui->
    mPerfTab));
37      } else {
38          // Remove mouse Performance tab from non-mice
39          if(!device->isMouse())
40              ui->tabWidget->removeTab(ui->tabWidget->indexOf(
    ui->mPerfTab));
41          // Remove keyboard Performance tab from non-keyboards
42          if(!device->isKeyboard())
43              ui->tabWidget->removeTab(ui->tabWidget->indexOf(
    ui->kPerfTab));
44      }
45      // Hide poll rate and FW update as appropriate
46      if(!device->features.contains("pollrate")){
47          ui->pollLabel->hide();
48          ui->pollLabel2->hide();
49      }
50      if(!device->features.contains("fwupdate")){
51          ui->fwUpdButton->hide();
52          ui->fwUpdLabel->hide();
53          ui->fwUpdLayout->removeItem(ui->fwUpdLayout->itemAt(1));
54      }
55      // Set monochrome mode according to hardware
56      if(device->monochrome)
57          ui->lightWidget->setMonochrome();
58  }
```

Here is the call graph for this function:



12.39.2.2 KbWidget::~KbWidget ( )

Definition at line 60 of file kbwidget.cpp.

References ui.

```
60                      {
61     delete ui;
62 }
```

12.39.3 Member Function Documentation

12.39.3.1 void KbWidget::addNewModelItem ( ) [private], [slot]

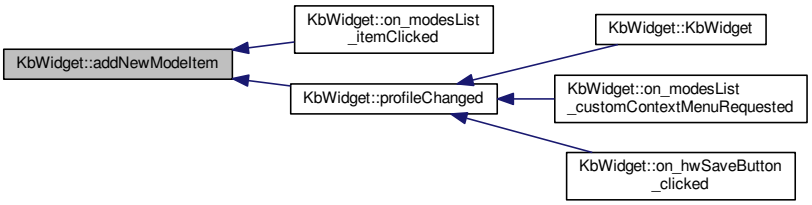
Definition at line 136 of file kbwidget.cpp.

References Ui_KbWidget::modesList, NEW_FLAG, and ui.

Referenced by on_modesList_itemClicked(), and profileChanged().

```
136                             {
137     // Add an item for creating a new mode. Make it editable but not dragable.
138     QListWidgetItem* item = new QListWidgetItem("New mode...", ui->modesList);
139     item->setFlags((item->flags() | Qt::ItemIsEditable) & ~Qt::ItemIsDragEnabled & ~Qt::ItemIsDropEnabled);
140     item->setData(NEW_FLAG, 1);
141     QFont font = item->font();
142     font.setItalic(true);
143     item->setFont(font);
144     item->setIcon(QIcon(":/img/icon_plus.png"));
145     ui->modesList->addItem(item);
146 }
```

Here is the caller graph for this function:



12.39.3.2 void KbWidget::devUpdate ( ) [private],[slot]

Definition at line 300 of file kbwidget.cpp.

References device, Kb::firmware, Ui_KbWidget::fwLabel, Ui_KbWidget::pollLabel, Kb::pollrate, Ui_KbWidget::serialLabel, ui, and Kb::usbSerial.

Referenced by KbWidget().

```
300                         {
301     // Update device tab
302     ui->serialLabel->setText(device->usbSerial);
303     ui->fwLabel->setText(device->firmware);
304     ui->pollLabel->setText(device->pollrate);
305 }
```

Here is the caller graph for this function:

12.39.3.3 void KbWidget::modeChanged (bool *spontaneous* = true) [private],[slot]

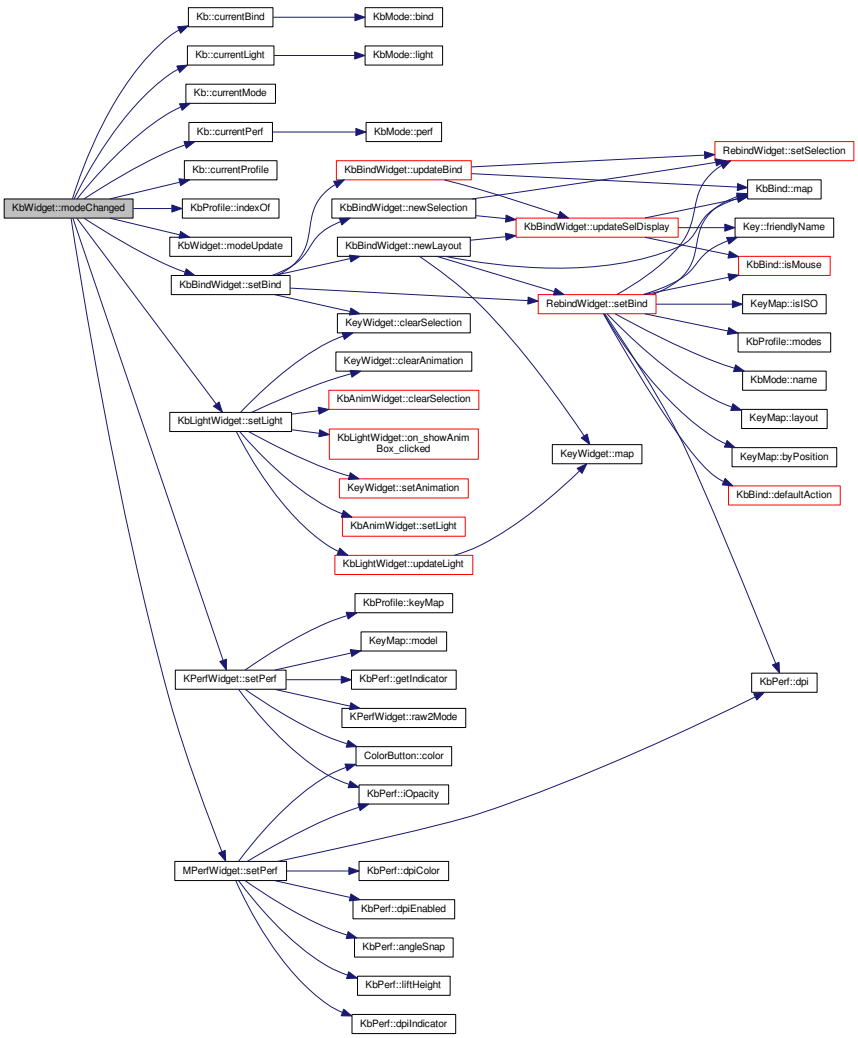
Definition at line 148 of file kbwidget.cpp.

References Ui_KbWidget::bindWidget, Kb::currentBind(), Kb::currentLight(), currentMode, Kb::currentMode(), Kb::currentPerf(), Kb::currentProfile(), device, KbProfile::indexOf(), Ui_KbWidget::kPerfWidget, Ui_KbWidget::lightWidget, Ui_KbWidget::modesList, modeUpdate(), Ui_KbWidget::mPerfWidget, KbBindWidget::setBind(), KbLightWidget::setLight(), KPerfWidget::setPerf(), MPerfWidget::setPerf(), and ui.

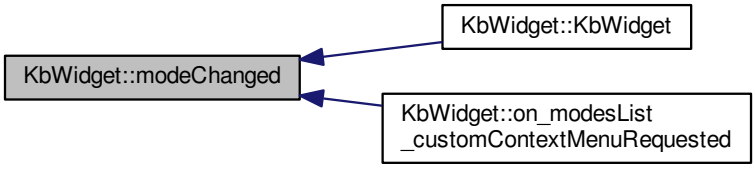
Referenced by KbWidget(), and on_modesList_customContextMenuRequested().

```
148                                         {
149     int index = device->currentProfile()->indexOf(
      device->currentMode());
150     if(index < 0)
151         return;
152     // Update tabs
153     ui->lightWidget->setLight(device->currentLight());
154     ui->bindWidget->setBind(device->currentBind(),
      device->currentProfile());
155     ui->kPerfWidget->setPerf(device->currentPerf(),
      device->currentProfile());
156     ui->mPerfWidget->setPerf(device->currentPerf(),
      device->currentProfile());
157     // Update selection
158     if(spontaneous)
159         ui->modesList->setCurrentRow(index);
160     // Connect signals
161     if(currentMode)
162         disconnect(currentMode, SIGNAL(updated()), this, SLOT(
      modeUpdate()));
163     currentMode = device->currentMode();
164     connect(currentMode, SIGNAL(updated()), this, SLOT(modeUpdate()));
165     modeUpdate();
166 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.3.4 QIcon KbWidget::modeIcon (int *i*) [private], [slot]

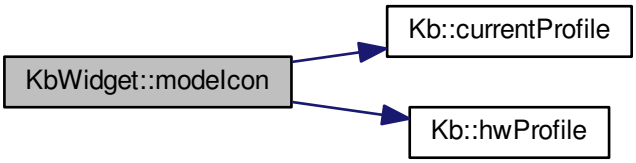
Definition at line 127 of file kbwidget.cpp.

References Kb::currentProfile(), device, Kb::hwModeCount, and Kb::hwProfile().

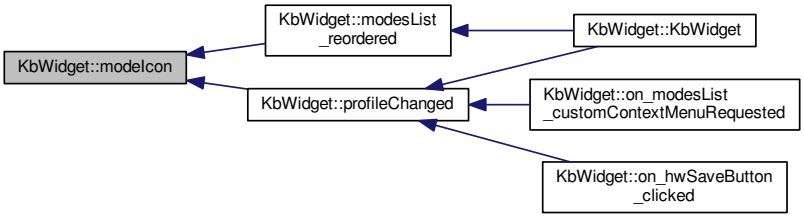
Referenced by modesList_reordered(), and profileChanged().

```
127                             {
128     KbProfile* currentProfile = device->currentProfile(), *hwProfile =
      device->hwProfile();
129     int hwModeCount = device->hwModeCount;
130     if(i >= hwModeCount)
131         return QIcon(":/img/icon_mode.png");
132     else
133         return QIcon(QString(currentProfile == hwProfile ? ":/img/icon_mode%1_hardware.png" : "
      :/img/icon_mode%1.png").arg(i + 1));
134 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.3.5 void KbWidget::modesList_reordered ( ) [private], [slot]

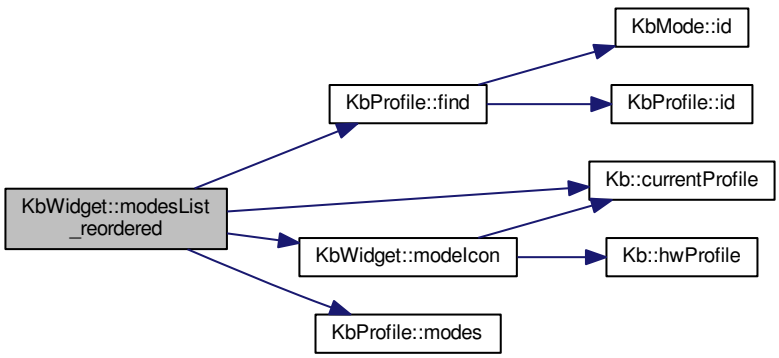
Definition at line 177 of file kbwidget.cpp.

References Kb::currentProfile(), device, KbProfile::find(), GUID, modelcon(), KbProfile::modes(), Ui_KbWidget-::modesList, NEW_FLAG, and ui.

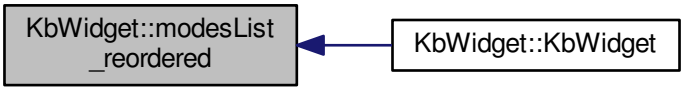
Referenced by KbWidget().

```
177                                 {
178     KbProfile* currentProfile = device->currentProfile();
179     // Rebuild mode list from items
180     QList<KbMode*> newModes;
181     int count = ui->modesList->count();
182     for(int i = 0; i < count; i++){
183         QListWidgetItem* item = ui->modesList->item(i);
184         KbMode* mode = currentProfile->find(item->data(GUID).toUuid());
185         if(mode && !newModes.contains(mode))
186             newModes.append(mode);
187         if(item->data(NEW_FLAG).toInt() != 1)
188             item->setIcon(modelcon(i));
189         item->setFlags(item->flags() | Qt::ItemIsEditable);
190     }
191     // Add any missing modes at the end of the list
192     foreach(KbMode* mode, currentProfile->modes()){
193         if(!newModes.contains(mode))
194             newModes.append(mode);
195     }
196     currentProfile->modes(newModes);
197 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.3.6 void KbWidget::modeUpdate ( ) [private],[slot]

Definition at line 307 of file kbwidget.cpp.

Referenced by modeChanged().

```
307                          {
308 }
```

Here is the caller graph for this function:

12.39.3.7 QString KbWidget::name ( ) const [inline]

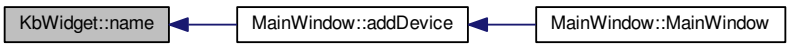
Definition at line 25 of file kbwidget.h.

References device, and Kb::usbModel.

Referenced by MainWindow::addDevice().

```
25 { return device->usbModel; }
```

Here is the caller graph for this function:



12.39.3.8 void KbWidget::on_fwUpdButton_clicked ( ) [private],[slot]

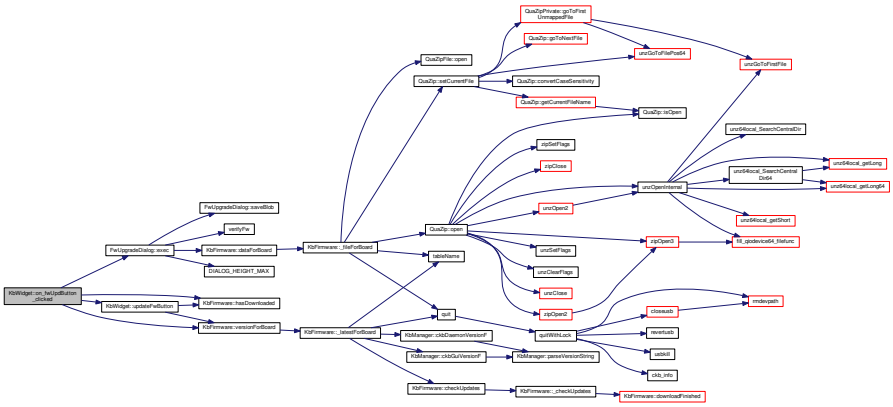
Definition at line 339 of file kbwidget.cpp.

References device, FwUpgradeDialog::exec(), Kb::features, Kb::firmware, Ui_KbWidget::fwUpdButton, KbFirmware::hasDownloaded(), ui, updateFwButton(), and KbFirmware::versionForBoard().

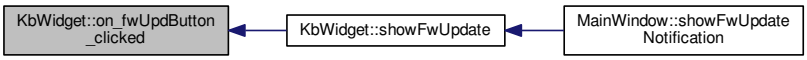
Referenced by showFwUpdate().

```
339                                     {
340     // If alt is pressed, ignore upgrades and go straight to the manual prompt
341     if(!(qApp->keyboardModifiers() & Qt::AltModifier)){
342         // Check version numbers
343         if(!KbFirmware::hasDownloaded()){
344             ui->fwUpdButton->setText("Checking...");
345             ui->fwUpdButton->setEnabled(false);
346         }
347         float newVersion = KbFirmware::versionForBoard(
    device->features, true);
348         float oldVersion = device->firmware.toFloat();
349         ui->fwUpdButton->setEnabled(true);
350         updateFwButton();
351         if(newVersion == -1.f){
352             QMessageBox::information(this, "Firmware update", "<center>There is a new firmware available
     for this device.<br />However, it requires a newer version of ckb.<br />Please upgrade ckb and try
     again.</center>");
353             return;
354         } else if(newVersion == 0.f){
355             if(QMessageBox::question(this, "Firmware update", "<center>There was a problem getting the
     status for this device.<br />Would you like to select a file manually?</center>") != QMessageBox::Yes)
356                 return;
357             // "Yes" -> fall through to browse file
358         } else if(newVersion <= oldVersion){
359             if(QMessageBox::question(this, "Firmware update", "<center>Your firmware is already up to
     date.<br />Would you like to select a file manually?</center>") != QMessageBox::Yes)
360                 return;
361             // "Yes" -> fall through to browse file
362         } else {
363             // Automatic upgrade. Fetch file from web.
364             // FwUpgradeDialog can't be parented to KbWidget because KbWidget may be deleted before the
     dialog exits
365             FwUpgradeDialog dialog(parentWidget(), newVersion, "",
    device);
366             dialog.exec();
367             return;
368         }
369     }
370     // Browse for file
371     QString path = QFileDialog::getOpenFileName(this, "Select firmware file",
    QStandardPaths::writableLocation(QStandardPaths::DownloadLocation), "Firmware blobs (*.bin)");
372     if(path.isEmpty())
373         return;
374     QFile file(path);
375     if(!file.open(QIODevice::ReadOnly)){
376         QMessageBox::warning(parentWidget(), "Error", "<center>File could not be read.</center>");
377         return;
378     }
379     QByteArray blob = file.readAll();
380     FwUpgradeDialog dialog(parentWidget(), 0.f, blob, device);
381     dialog.exec();
382 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



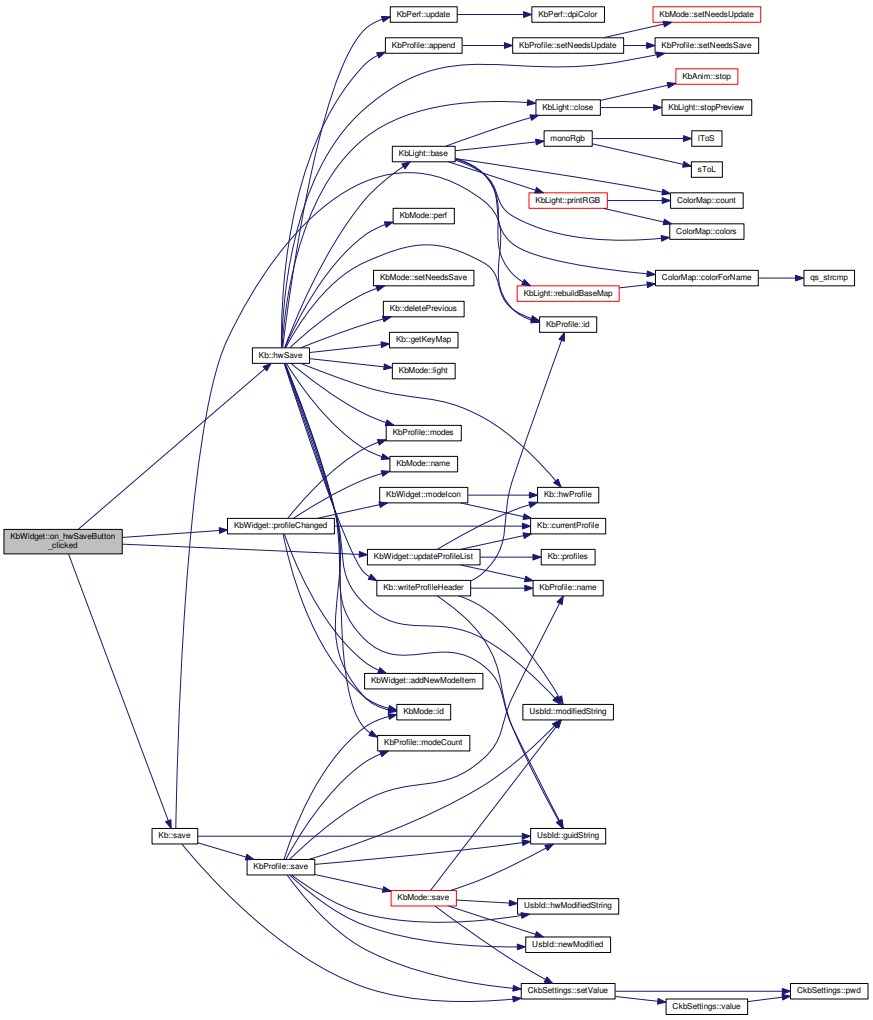
12.39.3.9 void KbWidget::on_hwSaveButton_clicked ( ) [private], [slot]

Definition at line 310 of file kbwidget.cpp.

References device, Kb::hwSave(), profileChanged(), Kb::save(), and updateProfileList().

```
310                                    {
311     device->save();
312     device->hwSave();
313     updateProfileList();
314     profileChanged();
315 }
```

Here is the call graph for this function:



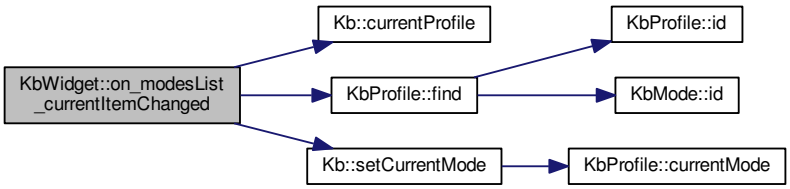
12.39.3.10 void KbWidget::on_modesList_currentItemChanged (QListWidgetItem * *current*, QListWidgetItem * *previous*) [private], [slot]

Definition at line 168 of file kbwidget.cpp.

References Kb::currentProfile(), device, KbProfile::find(), GUID, and Kb::setCurrentMode().

```
168                                                                                                 {
169     if(!current)
170         return;
171     KbMode* mode = device->currentProfile()->find(
      current->data(GUID).toUuid());
172     if(!mode)
173         return;
174     device->setCurrentMode(mode, false);
175 }
```

Here is the call graph for this function:



12.39.3.11 void KbWidget::on_modesList_customContextMenuRequested (const QPoint & *pos*) [private], [slot]

Parameters

pos	Opens on right click in the profiles list a context sensitive menue at position pos.

When clicking on a command it is located and executed.

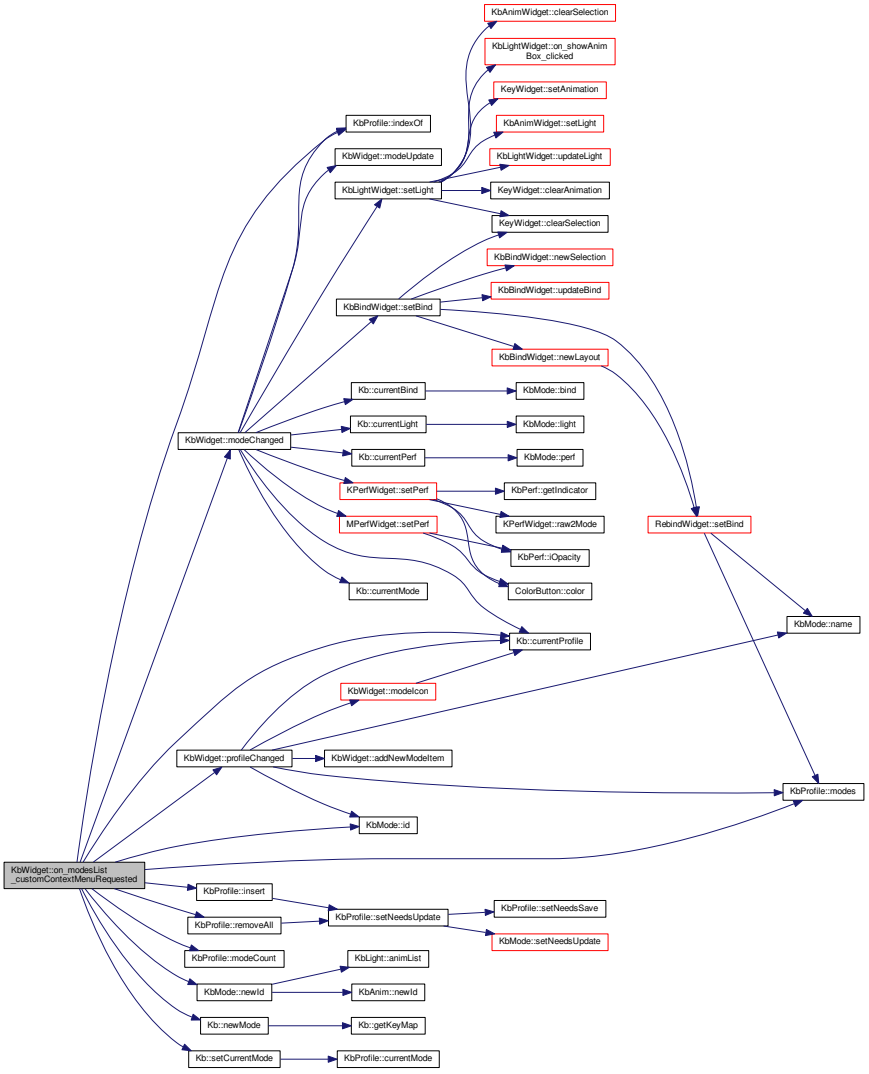
Definition at line 236 of file kbwidget.cpp.

References currentMode, Kb::currentProfile(), device, UsbId::guid, GUID, Kb::hwModeCount, KbMode::id(), KbProfile::indexOf(), KbProfile::insert(), modeChanged(), KbProfile::modeCount(), KbProfile::modes(), Ui_KbWidget::modesList, KbMode::newId(), Kb::newMode(), profileChanged(), KbProfile::removeAll(), Kb::setCurrentMode(), and ui.

```
236                                                                    {
237     QListWidgetItem* item = ui->modesList->itemAt(pos);
238     if(!item || !currentMode || item->data(GUID).toUuid() !=
    currentMode->id().guid)
239         return;
240     KbProfile* currentProfile = device->currentProfile();
241     int index = currentProfile->indexOf(currentMode);
242
243     QMenu menu(this);
244     QAction* rename = new QAction("Rename...", this);
245     QAction* duplicate = new QAction("Duplicate", this);
246     QAction* del = new QAction("Delete", this);
247     bool canDelete = (device->currentProfile()->modeCount() >
    device->hwModeCount);
248     if(!canDelete)
249         // Can't delete modes if they're required by hardware
250         del->setEnabled(false);
251     QAction* moveup = new QAction("Move Up", this);
252     if(index == 0)
253         moveup->setEnabled(false);
254     QAction* movedown = new QAction("Move Down", this);
255     if(index >= currentProfile->modeCount() - 1)
256         movedown->setEnabled(false);
257     menu.addAction(rename);
258     menu.addAction(duplicate);
259     menu.addAction(del);
260     menu.addSeparator();
261     menu.addAction(moveup);
262     menu.addAction(movedown);
263     QAction* result = menu.exec(QCursor::pos());
264     if(result == rename){
265         ui->modesList->editItem(item);
266     } else if(result == duplicate){
267         KbMode* newMode = device->newMode(currentMode);
268         newMode->newId();
269         currentProfile->insert(index + 1, newMode);
270         // Update UI
271         profileChanged();
272         device->setCurrentMode(newMode);
273     } else if(result == del){
274         if(!canDelete)
275             return;
276         currentProfile->removeAll(currentMode);
277         currentMode->deleteLater();
```

```
278         currentMode = 0;
279         // Select next mode
280         profileChanged();
281         if(index < currentProfile->modeCount())
282             device->setCurrentMode(currentProfile->modes()[index]);
283         else
284             device->setCurrentMode(currentProfile->modes().last());
285     } else if(result == moveup){
286         currentProfile->removeAll(currentMode);
287         currentProfile->insert(index - 1, currentMode);
288         // Update UI
289         profileChanged();
290         modeChanged(true);
291     } else if(result == movedown){
292         currentProfile->removeAll(currentMode);
293         currentProfile->insert(index + 1, currentMode);
294         // Update UI
295         profileChanged();
296         modeChanged(true);
297     }
298 }
```

Here is the call graph for this function:



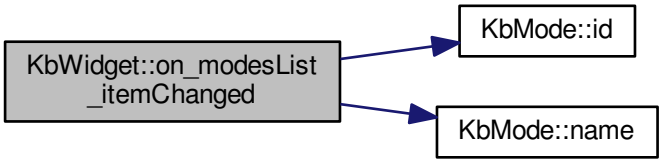
12.39.3.12 void KbWidget::on_modesList_itemChanged (QListWidgetItem * *item*) [private],[slot]

Definition at line 199 of file kbwidget.cpp.

References currentMode, UsbId::guid, GUID, KbMode::id(), and KbMode::name().

```
199                                                         {
200     if(!item || !currentMode || item->data(GUID).toUuid() !=
      currentMode->id().guid)
201         return;
202     currentMode->name(item->text());
203     // Set the text to the actual name (trimmed, "" replaced with "Unnamed")
204     item->setText(currentMode->name());
205 }
```

Here is the call graph for this function:



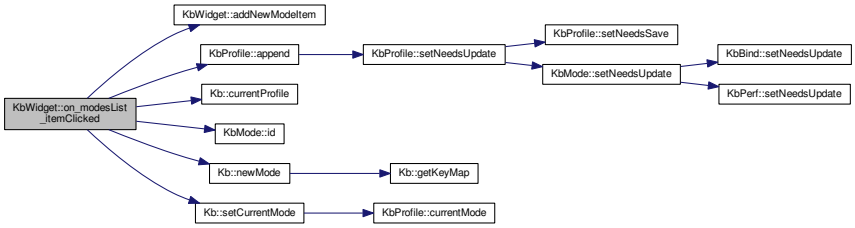
12.39.3.13 void KbWidget::on_modesList_itemClicked (QListWidgetItem * *item*) [private],[slot]

Definition at line 207 of file kbwidget.cpp.

References addNewModeItem(), KbProfile::append(), Kb::currentProfile(), device, UsbId::guid, GUID, KbMode::id(), Ui_KbWidget::modesList, NEW_FLAG, Kb::newMode(), Kb::setCurrentMode(), and ui.

```
207                                                         {
208     QUuid guid = item->data(GUID).toUuid();
209     if(guid.isNull() && item->data(NEW_FLAG).toInt() == 1){
210         // "New mode" item. Clear text and start editing
211         item->setText("");
212         ui->modesList->editItem(item);
213         item->setFlags(item->flags() | Qt::ItemIsEditable | Qt::ItemIsDragEnabled | Qt::ItemIsDropEnabled);
214         QFont font = item->font();
215         font.setItalic(false);
216         item->setFont(font);
217         item->setIcon(QIcon(":/img/icon_mode.png"));
218         // Add the new mode and assign it to this item
219         KbMode* newMode = device->newMode();
220         device->currentProfile()->append(newMode);
221         item->setData(GUID, newMode->id().guid);
222         item->setData(NEW_FLAG, 0);
223         device->setCurrentMode(newMode, false);
224         // Create another "new mode" item to replace this one
225         addNewModeItem();
226     }
227 }
```

Here is the call graph for this function:



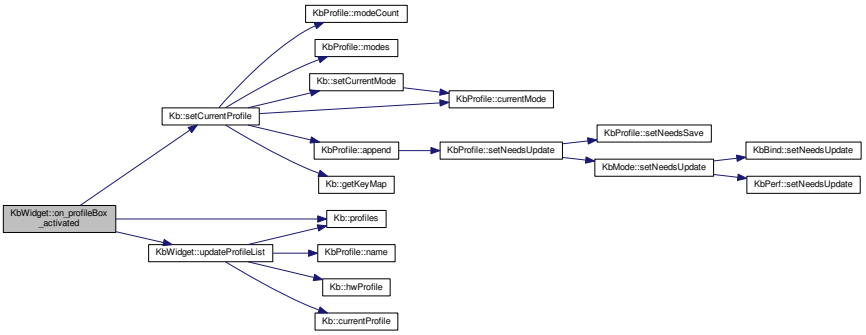
12.39.3.14 void KbWidget::on_profileBox_activated (int *index*) [private], [slot]

Definition at line 113 of file kbwidget.cpp.

References device, Kb::profiles(), Kb::setCurrentProfile(), and updateProfileList().

```
113                                                 {
114     if(index < 0)
115         return;
116     if(index >= device->profiles().count()){
117         // "Manage profiles" option
118         KbProfileDialog dialog(this);
119         dialog.exec();
120         updateProfileList();
121         return;
122     }
123     device->setCurrentProfile(device->profiles()[index]);
124     // Device will emit profileChanged() and modeChanged() signals to update UI
125 }
```

Here is the call graph for this function:



12.39.3.15 void KbWidget::on_tabWidget_currentChanged (int *index*) [private], [slot]

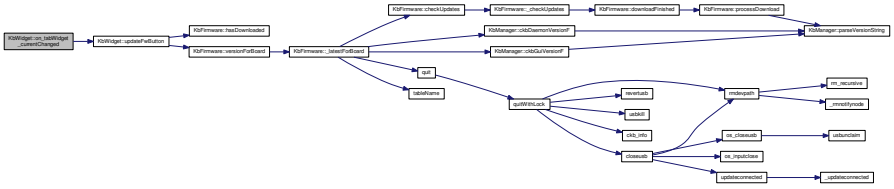
Definition at line 317 of file kbwidget.cpp.

References device, Ui_KbWidget::tabWidget, ui, and updateFwButton().

```
317                                                     {
318     if(!device)
319         return;
320     if(index == ui->tabWidget->count() - 1){
321         // Device tab
```

```
322         updateFwButton();
323     }
324 }
```

Here is the call graph for this function:



12.39.3.16 void KbWidget::profileChanged ( ) [private],[slot]

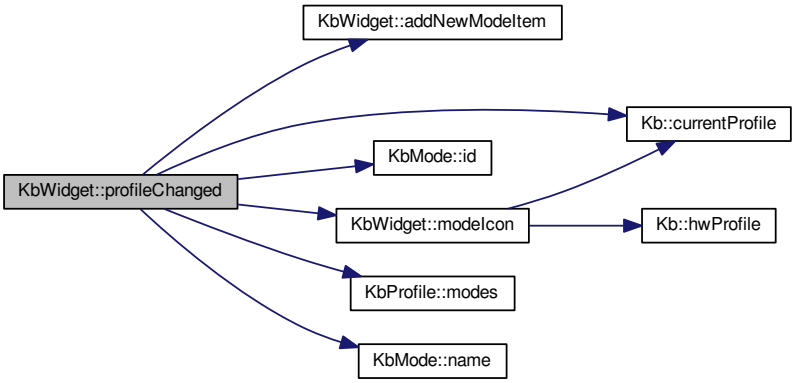
Definition at line 91 of file kbwidget.cpp.

References addNewModeItem(), current, currentMode, Kb::currentProfile(), device, UsbId::guid, GUID, KbMode::id(), modeIcon(), KbProfile::modes(), Ui_KbWidget::modesList, KbMode::name(), and ui.

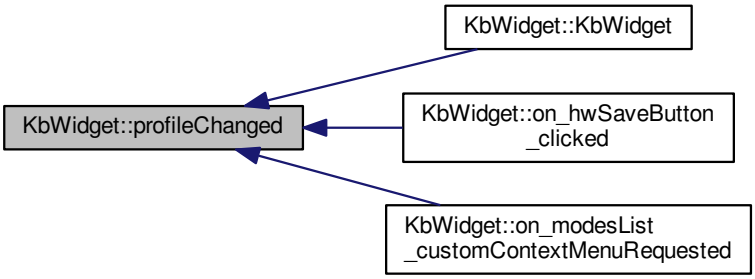
Referenced by KbWidget(), on_hwSaveButton_clicked(), and on_modesList_customContextMenuRequested().

```
91                          {
92     // Rebuild mode list
93     ui->modesList->clear();
94     int i = 0;
95     QListWidgetItem* current = 0;
96     foreach(KbMode* mode, device->currentProfile()->
       modes()){
97         QListWidgetItem* item = new QListWidgetItem(modeIcon(i), mode->
       name(), ui->modesList);
98         item->setData(GUID, mode->id().guid);
99         item->setFlags(item->flags() | Qt::ItemIsEditable);
100         if(mode == currentMode){
101             item->setSelected(true);
102             current = item;
103         }
104         ui->modesList->addItem(item);
105         i++;
106     }
107     if(current)
108         ui->modesList->setCurrentItem(current);
109     addNewModeItem();
110     // Wait for modeChanged() to refresh the rest of the UI
111 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.3.17 void KbWidget::showFirstTab ( ) [slot]

Definition at line 64 of file kbwidget.cpp.

References Ui_KbWidget::tabWidget, and ui.

```
64                              {
65     ui->tabWidget->setCurrentIndex(0);
66 }
```

12.39.3.18 void KbWidget::showFwUpdate ( ) [inline], [slot]

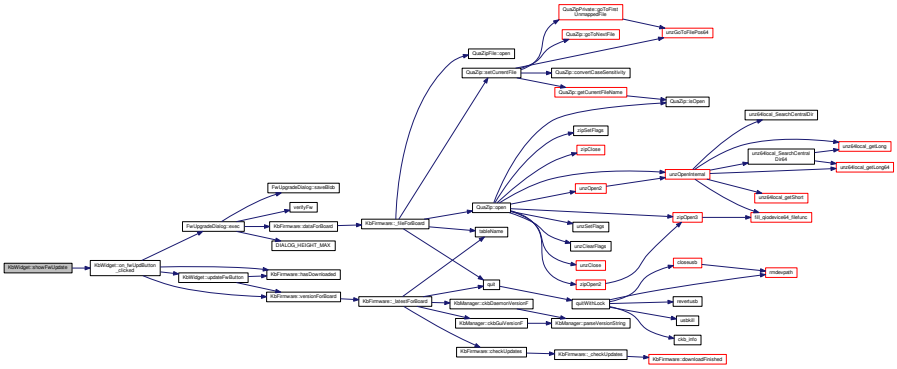
Definition at line 38 of file kbwidget.h.

References on_fwUpdButton_clicked().

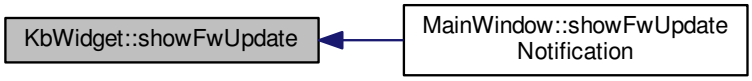
Referenced by MainWindow::showFwUpdateNotification().

```
38 { on_fwUpdButton_clicked(); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.3.19 void KbWidget::showLastTab ( ) [slot]

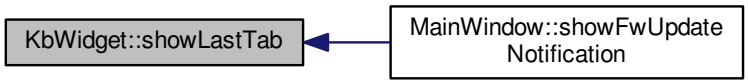
Definition at line 68 of file kbwidget.cpp.

References Ui_KbWidget::tabWidget, and ui.

Referenced by MainWindow::showFwUpdateNotification().

```
68                          {
69     ui->tabWidget->setCurrentIndex(ui->tabWidget->count() - 1);
70 }
```

Here is the caller graph for this function:



12.39.3.20 void KbWidget::updateFwButton ( )

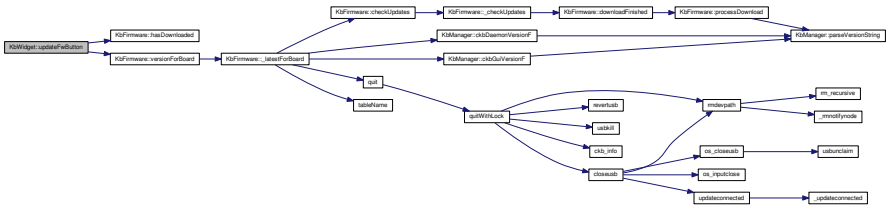
Definition at line 326 of file kbwidget.cpp.

References device, Kb::features, Kb::firmware, Ui_KbWidget::fwUpdButton, KbFirmware::hasDownloaded(), ui, and KbFirmware::versionForBoard().

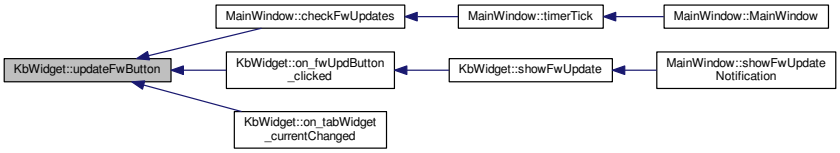
Referenced by MainWindow::checkFwUpdates(), on_fwUpdButton_clicked(), and on_tabWidget_currentChanged().

```
326                             {
327     if(!KbFirmware::hasDownloaded())
328         ui->fwUpdButton->setText("Check for updates");
329     else {
330         float newVersion = KbFirmware::versionForBoard(
      device->features);
331         float oldVersion = device->firmware.toFloat();
332         if(newVersion <= 0.f || newVersion <= oldVersion)
333             ui->fwUpdButton->setText("Up to date");
334         else
335             ui->fwUpdButton->setText(tr("Upgrade to v%1").arg(QString::number(newVersion, 'f',
       2)));
336     }
337 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.3.21 void KbWidget::updateProfileList ( ) [private],[slot]

Definition at line 73 of file kbwidget.cpp.

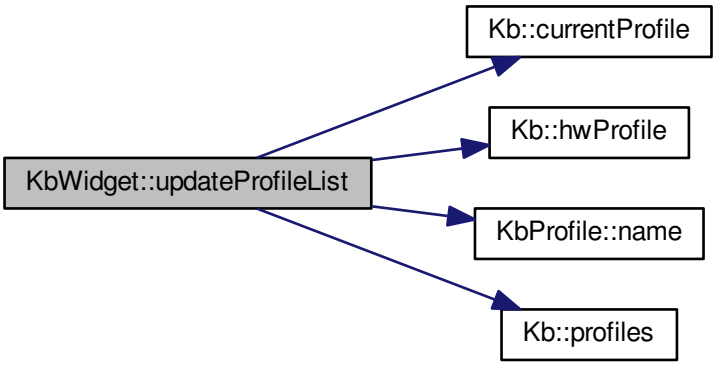
References Kb::currentProfile(), device, Kb::hwProfile(), KbProfile::name(), Ui_KbWidget::profileBox, Kb::profiles(), and ui.

Referenced by KbWidget(), on_hwSaveButton_clicked(), and on_profileBox_activated().

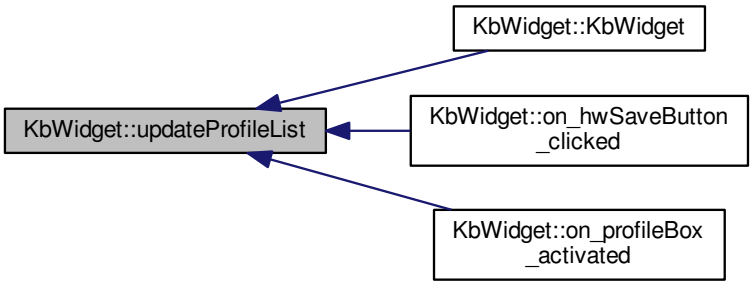
```
73                                 {
74     // Clear profile list and rebuild
75     KbProfile* hwProfile = device->hwProfile(), *currentProfile =
      device->currentProfile();
76     ui->profileBox->clear();
77     int i = 0;
78     foreach(KbProfile* profile, device->profiles()){
79         ui->profileBox->addItem((profile == hwProfile) ? QIcon(":/img/icon_profile_hardware.png
      ") : QIcon(":/img/icon_profile.png"),
80                                 profile->name());
81         if(profile == currentProfile)
82             ui->profileBox->setCurrentIndex(i);
83         i++;
84     }
```

```
85     ui->profileBox->addItem(QIcon(":/img/icon_blank.png"), "Manage profiles...");
86     QFont font = ui->profileBox->font();
87     font.setItalic(true);
88     ui->profileBox->setItemData(ui->profileBox->count() - 1, font, Qt::FontRole);
89 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.39.4 Field Documentation

12.39.4.1 KbMode* KbWidget::currentMode [private]

Definition at line 44 of file kbwidget.h.

Referenced by modeChanged(), on_modesList_customContextMenuRequested(), on_modesList_itemChanged(), and profileChanged().

12.39.4.2 Kb* KbWidget::device

Definition at line 24 of file kbwidget.h.

Referenced by MainWindow::addDevice(), MainWindow::checkFwUpdates(), devUpdate(), KbWidget(), modeChanged(), modeIcon(), modesList_reordered(), name(), on_fwUpdButton_clicked(), on_hwSaveButton_clicked(), on_modesList_currentItemChanged(), on_modesList_customContextMenuRequested(), on_modesList_itemClicked(), on_profileBox_activated(), on_tabWidget_currentChanged(), profileChanged(), MainWindow::removeDevice(), MainWindow::showFwUpdateNotification(), updateFwButton(), and updateProfileList().

12.39.4.3 const int KbWidget::GUID = Qt::UserRole [static], [private]

Definition at line 46 of file kbwidget.h.

Referenced by modesList_reordered(), on_modesList_currentItemChanged(), on_modesList_customContextMenuRequested(), on_modesList_itemChanged(), on_modesList_itemClicked(), and profileChanged().

12.39.4.4 bool KbWidget::hasShownNewFW

Definition at line 28 of file kbwidget.h.

Referenced by MainWindow::checkFwUpdates().

12.39.4.5 quint64 KbWidget::lastAutoSave [private]

Definition at line 42 of file kbwidget.h.

12.39.4.6 const int KbWidget::NEW_FLAG = Qt::UserRole + 1 [static], [private]

Definition at line 47 of file kbwidget.h.

Referenced by addNewModeItem(), modesList_reordered(), and on_modesList_itemClicked().

12.39.4.7 Ui::KbWidget* KbWidget::ui [private]

Definition at line 41 of file kbwidget.h.

Referenced by addNewModeItem(), devUpdate(), KbWidget(), modeChanged(), modesList_reordered(), on_fwUpdButton_clicked(), on_modesList_customContextMenuRequested(), on_modesList_itemClicked(), on_tabWidget_currentChanged(), profileChanged(), showFirstTab(), showLastTab(), updateFwButton(), updateProfileList(), and ~KbWidget().

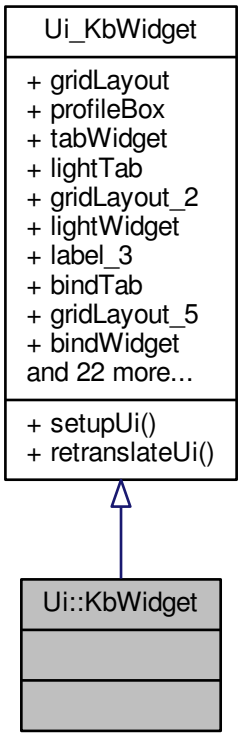
The documentation for this class was generated from the following files:

- src/ckb/kbwidget.h
- src/ckb/kbwidget.cpp

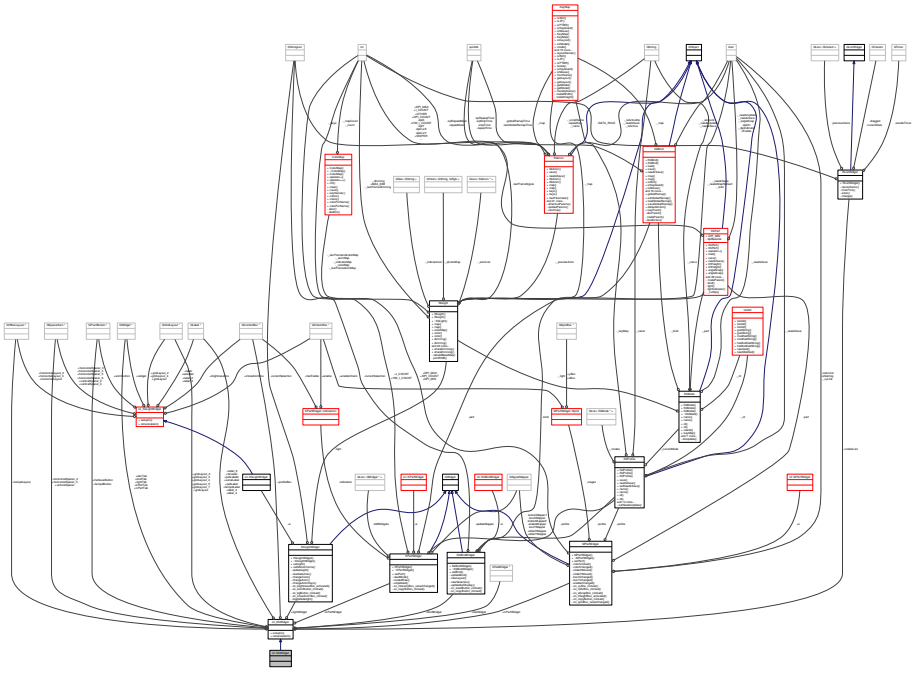
12.40 Ui::KbWidget Class Reference

`#include <src/ckb/ui_kbwidget.h>`

Inheritance diagram for Ui::KbWidget:



Collaboration diagram for Ui::KbWidget:



Public Member Functions

- void setupUi (QWidget *KbWidget)
- void retranslateUi (QWidget *KbWidget)

Data Fields

- QGridLayout * gridLayout
- QComboBox * profileBox
- QTabWidget * tabWidget
- QWidget * lightTab
- QGridLayout * gridLayout_2
- KbLightWidget * lightWidget
- QLabel * label_3
- QWidget * bindTab
- QGridLayout * gridLayout_5
- KbBindWidget * bindWidget
- QWidget * kPerfTab
- QGridLayout * gridLayout_4
- KPerfWidget * kPerfWidget
- QWidget * mPerfTab
- QGridLayout * gridLayout_7
- MPerfWidget * mPerfWidget
- QWidget * devTab
- QGridLayout * gridLayout_3
- QLabel * fwUpdLabel
- QSpacerItem * verticalSpacer
- QLabel * fwLabel
- QLabel * label_4
- QSpacerItem * horizontalSpacer_2
- QLabel * label_9
- QLabel * serialLabel
- QHBoxLayout * fwUpdLayout
- QPushButton * fwUpdButton
- QSpacerItem * horizontalSpacer_3
- QLabel * pollLabel2
- QLabel * pollLabel
- RListWidget * modesList
- QPushButton * hwSaveButton

12.40.1 Detailed Description

Definition at line 291 of file ui_kbwidget.h.

12.40.2 Member Function Documentation

12.40.2.1 void Ui_KbWidget::retranslateUi (QWidget * *KbWidget*) [inline], [inherited]

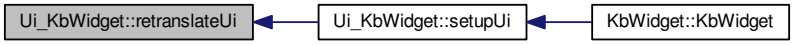
Definition at line 268 of file ui_kbwidget.h.

References Ui_KbWidget::bindTab, Ui_KbWidget::devTab, Ui_KbWidget::fwLabel, Ui_KbWidget::fwUpdButton, Ui_KbWidget::fwUpdLabel, Ui_KbWidget::hwSaveButton, Ui_KbWidget::kPerfTab, Ui_KbWidget::label_3, Ui_KbWidget::label_4, Ui_KbWidget::label_9, Ui_KbWidget::lightTab, Ui_KbWidget::mPerfTab, Ui_KbWidget::pollLabel, Ui_KbWidget::pollLabel2, Ui_KbWidget::serialLabel, and Ui_KbWidget::tabWidget.

Referenced by Ui_KbWidget::setupUi().

```
269     {
270         KbWidget->setWindowTitle(QApplication::translate("KbWidget", "Form", Q_NULLPTR));
271         label_3->setText(QApplication::translate("KbWidget", "Tip: Drag+drop items to reorder.
        Right-click for menu.", Q_NULLPTR));
272         tabWidget->setTabText(tabWidget->indexOf(lightTab),
       QApplication::translate("KbWidget", "Lighting", Q_NULLPTR));
273         tabWidget->setTabText(tabWidget->indexOf(bindTab), QApplication::translate
       ("KbWidget", "Binding", Q_NULLPTR));
274         tabWidget->setTabText(tabWidget->indexOf(kPerfTab),
       QApplication::translate("KbWidget", "Performance", Q_NULLPTR));
275         tabWidget->setTabText(tabWidget->indexOf(mPerfTab),
       QApplication::translate("KbWidget", "Performance", Q_NULLPTR));
276         fwUpdLabel->setText(QApplication::translate("KbWidget", "Status:", Q_NULLPTR));
277         fwLabel->setText(QApplication::translate("KbWidget", "N/A", Q_NULLPTR));
278         label_4->setText(QApplication::translate("KbWidget", "Serial Number:", Q_NULLPTR));
279         label_9->setText(QApplication::translate("KbWidget", "Firmware version:", Q_NULLPTR));
280         serialLabel->setText(QApplication::translate("KbWidget", "N/A", Q_NULLPTR));
281         fwUpdButton->setText(QApplication::translate("KbWidget", "Check for updates", Q_NULLPTR)
       );
282         pollLabel2->setText(QApplication::translate("KbWidget", "Poll rate:", Q_NULLPTR));
283         pollLabel->setText(QApplication::translate("KbWidget", "N/A", Q_NULLPTR));
284         tabWidget->setTabText(tabWidget->indexOf(devTab), QApplication::translate("
       KbWidget", "Device", Q_NULLPTR));
285         hwSaveButton->setText(QApplication::translate("KbWidget", "Save to hardware", Q_NULLPTR
       ));
286     } // retranslateUi
```

Here is the caller graph for this function:



12.40.2.2 void Ui_KbWidget::setupUi (QWidget * *KbWidget*) [inline], [inherited]

Definition at line 69 of file ui_kbwidget.h.

References Ui_KbWidget::bindTab, Ui_KbWidget::bindWidget, Ui_KbWidget::devTab, Ui_KbWidget::fwLabel, Ui_KbWidget::fwUpdButton, Ui_KbWidget::fwUpdLabel, Ui_KbWidget::fwUpdLayout, Ui_KbWidget::gridLayout, Ui_KbWidget::gridLayout_2, Ui_KbWidget::gridLayout_3, Ui_KbWidget::gridLayout_4, Ui_KbWidget::gridLayout_5, Ui_KbWidget::gridLayout_7, Ui_KbWidget::horizontalSpacer_2, Ui_KbWidget::horizontalSpacer_3, Ui_KbWidget::hwSaveButton, Ui_KbWidget::kPerfTab, Ui_KbWidget::kPerfWidget, Ui_KbWidget::label_3, Ui_KbWidget::label_4, Ui_KbWidget::label_9, Ui_KbWidget::lightTab, Ui_KbWidget::lightWidget, Ui_KbWidget::modesList, Ui_KbWidget::mPerfTab, Ui_KbWidget::mPerfWidget, Ui_KbWidget::pollLabel, Ui_KbWidget::pollLabel2, Ui_KbWidget::profileBox, Ui_KbWidget::retranslateUi(), Ui_KbWidget::serialLabel, Ui_KbWidget::tabWidget, and Ui_KbWidget::verticalSpacer.

Referenced by KbWidget::KbWidget().

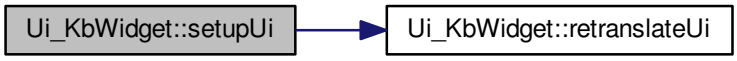
```
70     {
71         if (KbWidget->objectName().isEmpty())
72             KbWidget->setObjectName(QStringLiteral("KbWidget"));
73         KbWidget->resize(773, 539);
74         gridLayout = new QGridLayout(KbWidget);
75         gridLayout->setSpacing(9);
76         gridLayout->setObjectName(QStringLiteral("gridLayout"));
77         gridLayout->setContentsMargins(9, -1, 9, 9);
78         profileBox = new QComboBox(KbWidget);
79         profileBox->setObjectName(QStringLiteral("profileBox"));
80         QSizePolicy sizePolicy(QSizePolicy::Preferred, QSizePolicy::Fixed);
81         sizePolicy.setHorizontalStretch(18);
82         sizePolicy.setVerticalStretch(0);
83         sizePolicy.setHeightForWidth(profileBox->sizePolicy().hasHeightForWidth());
84         profileBox->setSizePolicy(sizePolicy);
85         profileBox->setIconSize(QSize(24, 24));
86
87         gridLayout->addWidget(profileBox, 0, 0, 1, 1);
88
```

```
89          tabWidget = new QTabWidget(KbWidget);
90          tabWidget->setObjectName(QStringLiteral("tabWidget"));
91          QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Expanding);
92          sizePolicy1.setHorizontalStretch(82);
93          sizePolicy1.setVerticalStretch(0);
94          sizePolicy1.setHeightForWidth(tabWidget->sizePolicy().hasHeightForWidth());
95          tabWidget->setSizePolicy(sizePolicy1);
96          lightTab = new QWidget();
97          lightTab->setObjectName(QStringLiteral("lightTab"));
98          gridLayout_2 = new QGridLayout(lightTab);
99          gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
100          gridLayout_2->setHorizontalSpacing(0);
101          gridLayout_2->setVerticalSpacing(6);
102          gridLayout_2->setContentsMargins(6, 6, 6, 6);
103          lightWidget = new KbLightWidget(lightTab);
104          lightWidget->setObjectName(QStringLiteral("lightWidget"));
105          QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Expanding);
106          sizePolicy2.setHorizontalStretch(0);
107          sizePolicy2.setVerticalStretch(0);
108          sizePolicy2.setHeightForWidth(lightWidget->sizePolicy().hasHeightForWidth());
109          lightWidget->setSizePolicy(sizePolicy2);
110
111          gridLayout_2->addWidget(lightWidget, 0, 0, 1, 1);
112
113          label_3 = new QLabel(lightTab);
114          label_3->setObjectName(QStringLiteral("label_3"));
115          QSizePolicy sizePolicy3(QSizePolicy::Preferred, QSizePolicy::Minimum);
116          sizePolicy3.setHorizontalStretch(0);
117          sizePolicy3.setVerticalStretch(0);
118          sizePolicy3.setHeightForWidth(label_3->sizePolicy().hasHeightForWidth());
119          label_3->setSizePolicy(sizePolicy3);
120
121          gridLayout_2->addWidget(label_3, 1, 0, 1, 1);
122
123          tabWidget->addTab(lightTab, QString());
124          bindTab = new QWidget();
125          bindTab->setObjectName(QStringLiteral("bindTab"));
126          gridLayout_5 = new QGridLayout(bindTab);
127          gridLayout_5->setSpacing(0);
128          gridLayout_5->setObjectName(QStringLiteral("gridLayout_5"));
129          gridLayout_5->setContentsMargins(6, 6, 6, 6);
130          bindWidget = new KbBindWidget(bindTab);
131          bindWidget->setObjectName(QStringLiteral("bindWidget"));
132          sizePolicy2.setHeightForWidth(bindWidget->sizePolicy().hasHeightForWidth());
133          bindWidget->setSizePolicy(sizePolicy2);
134
135          gridLayout_5->addWidget(bindWidget, 0, 0, 1, 1);
136
137          tabWidget->addTab(bindTab, QString());
138          kPerfTab = new QWidget();
139          kPerfTab->setObjectName(QStringLiteral("kPerfTab"));
140          gridLayout_4 = new QGridLayout(kPerfTab);
141          gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
142          gridLayout_4->setContentsMargins(6, 6, 6, 6);
143          kPerfWidget = new KPerfWidget(kPerfTab);
144          kPerfWidget->setObjectName(QStringLiteral("kPerfWidget"));
145
146          gridLayout_4->addWidget(kPerfWidget, 0, 0, 1, 1);
147
148          tabWidget->addTab(kPerfTab, QString());
149          mPerfTab = new QWidget();
150          mPerfTab->setObjectName(QStringLiteral("mPerfTab"));
151          gridLayout_7 = new QGridLayout(mPerfTab);
152          gridLayout_7->setObjectName(QStringLiteral("gridLayout_7"));
153          gridLayout_7->setContentsMargins(6, 6, 6, 6);
154          mPerfWidget = new MPerfWidget(mPerfTab);
155          mPerfWidget->setObjectName(QStringLiteral("mPerfWidget"));
156
157          gridLayout_7->addWidget(mPerfWidget, 0, 0, 1, 1);
158
159          tabWidget->addTab(mPerfTab, QString());
160          devTab = new QWidget();
161          devTab->setObjectName(QStringLiteral("devTab"));
162          gridLayout_3 = new QGridLayout(devTab);
163          gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
164          gridLayout_3->setContentsMargins(6, 6, 6, 6);
165          fwUpdLabel = new QLabel(devTab);
166          fwUpdLabel->setObjectName(QStringLiteral("fwUpdLabel"));
167          fwUpdLabel->setMinimumSize(QSize(0, 34));
168
169          gridLayout_3->addWidget(fwUpdLabel, 4, 0, 1, 1);
170
171          verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
172
173          gridLayout_3->addItem(verticalSpacer, 6, 2, 1, 1);
174
```

```
175         fwLabel = new QLabel(devTab);
176         fwLabel->setObjectName(QStringLiteral("fwLabel"));
177         fwLabel->setCursor(QCursor(Qt::IBeamCursor));
178         fwLabel->setTextInteractionFlags(Qt::LinksAccessibleByMouse|Qt::TextSelectableByMouse);
179 
180         gridLayout_3->addWidget(fwLabel, 2, 2, 1, 1);
181 
182         label_4 = new QLabel(devTab);
183         label_4->setObjectName(QStringLiteral("label_4"));
184         label_4->setMinimumSize(QSize(0, 34));
185 
186         gridLayout_3->addWidget(label_4, 0, 0, 1, 1);
187 
188         horizontalSpacer_2 = new QSpacerItem(40, 1, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
189 
190         gridLayout_3->addItem(horizontalSpacer_2, 2, 1, 1, 1);
191 
192         label_9 = new QLabel(devTab);
193         label_9->setObjectName(QStringLiteral("label_9"));
194         label_9->setMinimumSize(QSize(0, 34));
195 
196         gridLayout_3->addWidget(label_9, 2, 0, 1, 1);
197 
198         serialLabel = new QLabel(devTab);
199         serialLabel->setObjectName(QStringLiteral("serialLabel"));
200         serialLabel->setCursor(QCursor(Qt::IBeamCursor));
201         serialLabel->setTextInteractionFlags(Qt::LinksAccessibleByMouse|
    Qt::TextSelectableByMouse);
202 
203         gridLayout_3->addWidget(serialLabel, 0, 2, 1, 1);
204 
205         fwUpdLayout = new QHBoxLayout();
206         fwUpdLayout->setObjectName(QStringLiteral("fwUpdLayout"));
207         fwUpdButton = new QPushButton(devTab);
208         fwUpdButton->setObjectName(QStringLiteral("fwUpdButton"));
209 
210         fwUpdLayout->addWidget(fwUpdButton);
211 
212         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
213 
214         fwUpdLayout->addItem(horizontalSpacer_3);
215 
216 
217         gridLayout_3->addLayout(fwUpdLayout, 4, 2, 1, 1);
218 
219         pollLabel2 = new QLabel(devTab);
220         pollLabel2->setObjectName(QStringLiteral("pollLabel2"));
221         pollLabel2->setMinimumSize(QSize(0, 34));
222 
223         gridLayout_3->addWidget(pollLabel2, 1, 0, 1, 1);
224 
225         pollLabel = new QLabel(devTab);
226         pollLabel->setObjectName(QStringLiteral("pollLabel"));
227 
228         gridLayout_3->addWidget(pollLabel, 1, 2, 1, 1);
229 
230         tabWidget->addTab(devTab, QString());
231 
232         gridLayout->addWidget(tabWidget, 0, 1, 3, 1);
233 
234         modesList = new RListWidget(KbWidget);
235         modesList->setObjectName(QStringLiteral("modesList"));
236         QSizePolicy sizePolicy4(QSizePolicy::Expanding, QSizePolicy::Expanding);
237         sizePolicy4.setHorizontalStretch(18);
238         sizePolicy4.setVerticalStretch(0);
239         sizePolicy4.setHeightForWidth(modesList->sizePolicy().hasHeightForWidth());
240         modesList->setSizePolicy(sizePolicy4);
241         modesList->setContextMenuPolicy(Qt::CustomContextMenu);
242         modesList->setFrameShadow(QFrame::Plain);
243         modesList->setDragDropMode(QAbstractItemView::InternalMove);
244         modesList->setIconSize(QSize(24, 24));
245         modesList->setMovement(QListView::Snap);
246 
247         gridLayout->addWidget(modesList, 1, 0, 1, 1);
248 
249         hwSaveButton = new QPushButton(KbWidget);
250         hwSaveButton->setObjectName(QStringLiteral("hwSaveButton"));
251         QSizePolicy sizePolicy5(QSizePolicy::Minimum, QSizePolicy::Fixed);
252         sizePolicy5.setHorizontalStretch(18);
253         sizePolicy5.setVerticalStretch(0);
254         sizePolicy5.setHeightForWidth(hwSaveButton->sizePolicy().hasHeightForWidth());
255         hwSaveButton->setSizePolicy(sizePolicy5);
256 
257         gridLayout->addWidget(hwSaveButton, 2, 0, 1, 1);
258 
```

```
259
260         retranslateUi(KbWidget);
261
262         tabWidget->setCurrentIndex(0);
263
264
265         QMetaObject::connectSlotsByName(KbWidget);
266     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.40.3 Field Documentation

12.40.3.1 QWidget* Ui_KbWidget::bindTab [inherited]

Definition at line 43 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.2 KbBindWidget* Ui_KbWidget::bindWidget [inherited]

Definition at line 45 of file ui_kbwidget.h.

Referenced by KbWidget::modeChanged(), and Ui_KbWidget::setupUi().

12.40.3.3 QWidget* Ui_KbWidget::devTab [inherited]

Definition at line 52 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.4 QLabel* Ui_KbWidget::fwLabel [inherited]

Definition at line 56 of file ui_kbwidget.h.

Referenced by KbWidget::devUpdate(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.5 QPushButton* Ui_KbWidget::fwUpdButton [inherited]

Definition at line 62 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), KbWidget::on_fwUpdButton_clicked(), Ui_KbWidget::retranslateUi(), Ui_KbWidget::setupUi(), and KbWidget::updateFwButton().

12.40.3.6 QLabel* Ui_KbWidget::fwUpdLabel [inherited]

Definition at line 54 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.7 QHBoxLayout* Ui_KbWidget::fwUpdLayout [inherited]

Definition at line 61 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), and Ui_KbWidget::setupUi().

12.40.3.8 QGridLayout* Ui_KbWidget::gridLayout [inherited]

Definition at line 36 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.9 QGridLayout* Ui_KbWidget::gridLayout_2 [inherited]

Definition at line 40 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.10 QGridLayout* Ui_KbWidget::gridLayout_3 [inherited]

Definition at line 53 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.11 QGridLayout* Ui_KbWidget::gridLayout_4 [inherited]

Definition at line 47 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.12 QGridLayout* Ui_KbWidget::gridLayout_5 [inherited]

Definition at line 44 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.13 QGridLayout* Ui_KbWidget::gridLayout_7 [inherited]

Definition at line 50 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.14 QSpacerItem* Ui_KbWidget::horizontalSpacer_2 [inherited]

Definition at line 58 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.15 QSpacerItem* Ui_KbWidget::horizontalSpacer_3 [inherited]

Definition at line 63 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

12.40.3.16 QPushButton* Ui_KbWidget::hwSaveButton [inherited]

Definition at line 67 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.17 QWidget* Ui_KbWidget::kPerfTab [inherited]

Definition at line 46 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.18 KPerfWidget* Ui_KbWidget::kPerfWidget [inherited]

Definition at line 48 of file ui_kbwidget.h.

Referenced by KbWidget::modeChanged(), and Ui_KbWidget::setupUi().

12.40.3.19 QLabel* Ui_KbWidget::label_3 [inherited]

Definition at line 42 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.20 QLabel* Ui_KbWidget::label_4 [inherited]

Definition at line 57 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.21 QLabel* Ui_KbWidget::label_9 [inherited]

Definition at line 59 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.22 QWidget* Ui_KbWidget::lightTab [inherited]

Definition at line 39 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.23 KbLightWidget* Ui_KbWidget::lightWidget [inherited]

Definition at line 41 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), KbWidget::modeChanged(), and Ui_KbWidget::setupUi().

12.40.3.24 RListWidget* Ui_KbWidget::modesList [inherited]

Definition at line 66 of file ui_kbwidget.h.

Referenced by KbWidget::addNewModeItem(), KbWidget::KbWidget(), KbWidget::modeChanged(), KbWidget::modesList_reordered(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::on_modesList_itemClicked(), KbWidget::profileChanged(), and Ui_KbWidget::setupUi().

12.40.3.25 QWidget* Ui_KbWidget::mPerfTab [inherited]

Definition at line 49 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.26 MPerfWidget* Ui_KbWidget::mPerfWidget [inherited]

Definition at line 51 of file ui_kbwidget.h.

Referenced by KbWidget::modeChanged(), and Ui_KbWidget::setupUi().

12.40.3.27 QLabel* Ui_KbWidget::pollLabel [inherited]

Definition at line 65 of file ui_kbwidget.h.

Referenced by KbWidget::devUpdate(), KbWidget::KbWidget(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.28 QLabel* Ui_KbWidget::pollLabel2 [inherited]

Definition at line 64 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.29 QComboBox* Ui_KbWidget::profileBox [inherited]

Definition at line 37 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi(), and KbWidget::updateProfileList().

12.40.3.30 QLabel* Ui_KbWidget::serialLabel [inherited]

Definition at line 60 of file ui_kbwidget.h.

Referenced by KbWidget::devUpdate(), Ui_KbWidget::retranslateUi(), and Ui_KbWidget::setupUi().

12.40.3.31 QTabWidget* Ui_KbWidget::tabWidget [inherited]

Definition at line 38 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), KbWidget::on_tabWidget_currentChanged(), Ui_KbWidget::retranslateUi(), Ui_KbWidget::setupUi(), KbWidget::showFirstTab(), and KbWidget::showLastTab().

12.40.3.32 QSpacerItem* Ui_KbWidget::verticalSpacer [inherited]

Definition at line 55 of file ui_kbwidget.h.

Referenced by Ui_KbWidget::setupUi().

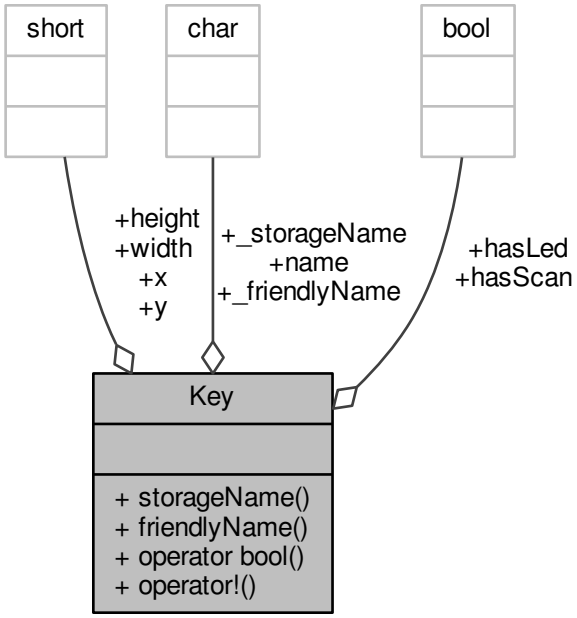
The documentation for this class was generated from the following file:

- src/ckb/ui_kbwidget.h

12.41 Key Struct Reference

`#include <src/ckb/keymap.h>`

Collaboration diagram for Key:



Public Member Functions

- const char * storageName () const
- QString friendlyName (bool os=true) const
- operator bool () const
- bool operator! () const

Data Fields

- const char * _storageName
- const char * _friendlyName
- const char * name

- short x
- short y
- short width
- short height
- bool hasLed
- bool hasScan

12.41.1 Detailed Description

Definition at line 8 of file keymap.h.

12.41.2 Member Function Documentation

12.41.2.1 QString Key::friendlyName (bool *os* = `true`) const [inline]

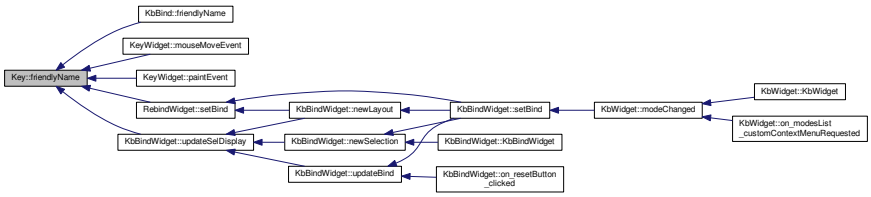
Definition at line 23 of file keymap.h.

References _friendlyName, and name.

Referenced by KbBind::friendlyName(), KeyWidget::mouseMoveEvent(), KeyWidget::paintEvent(), RebindWidget::setBind(), and KbBindWidget::updateSelDisplay().

```
23                                                      {
24         if(os){
25 #ifdef Q_OS_MACX
26             if(!strcmp(name, "lwin")) return "Left Cmd";
27             if(!strcmp(name, "rwin")) return "Right Cmd";
28             if(!strcmp(name, "lalt")) return "Left Option";
29             if(!strcmp(name, "ralt")) return "Right Option";
30             if(!strcmp(name, "prtscn")) return "F13";
31             if(!strcmp(name, "scroll")) return "F14";
32             if(!strcmp(name, "pause")) return "F15";
33             if(!strcmp(name, "ins")) return "Help";
34             if(!strcmp(name, "numlock")) return "Clear";
35 #elif defined(Q_OS_LINUX)
36             if(!strcmp(name, "lwin")) return "Left Super";
37             if(!strcmp(name, "rwin")) return "Right Super";
38 #endif
39         }
40         return _friendlyName ? _friendlyName : QString(
      name).toUpper();
41     }
```

Here is the caller graph for this function:



12.41.2.2 Key::operator bool ( ) const [inline]

Definition at line 44 of file keymap.h.

References name.

```
44 { return name != 0; }
```

12.41.2.3 bool Key::operator! ( ) const [inline]

Definition at line 45 of file keymap.h.

```
45 { return !(bool)*this; }
```

12.41.2.4 const char* Key::storageName ( ) const [inline]

Definition at line 11 of file keymap.h.

References _storageName, and name.

Referenced by KeyMap::toStorage().

```
11 { return _storageName ? _storageName : name; }
```

Here is the caller graph for this function:

12.41.3 Field Documentation

12.41.3.1 const char* Key::_friendlyName

Definition at line 13 of file keymap.h.

Referenced by _patch(), and friendlyName().

12.41.3.2 const char* Key::_storageName

Definition at line 10 of file keymap.h.

Referenced by _patch(), and storageName().

12.41.3.3 bool Key::hasLed

Definition at line 19 of file keymap.h.

Referenced by ColorMap::init(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and KeyWidget::selectAll().

12.41.3.4 bool Key::hasScan

Definition at line 20 of file keymap.h.

Referenced by KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and KeyWidget::selectAll().

12.41.3.5 short Key::height

Definition at line 17 of file keymap.h.

Referenced by getMap(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and patchANSI().

12.41.3.6 const char* Key::name

Definition at line 14 of file keymap.h.

Referenced by friendlyName(), ColorMap::init(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), operator bool(), KeyWidget::paintEvent(), KeyWidget::selectAll(), and storageName().

12.41.3.7 short Key::width

Definition at line 17 of file keymap.h.

Referenced by getMap(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), patchABNT2(), patchANSI(), patchJP106(), and patchJP106fn().

12.41.3.8 short Key::x

Definition at line 16 of file keymap.h.

Referenced by AnimScript::begin(), KeyMap::byPosition(), getMap(), AnimScript::keypress(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), patchABNT2(), patchANSI(), patchJP106(), and patchJP106fn().

12.41.3.9 short Key::y

Definition at line 16 of file keymap.h.

Referenced by AnimScript::begin(), KeyMap::byPosition(), getMap(), AnimScript::keypress(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and patchANSI().

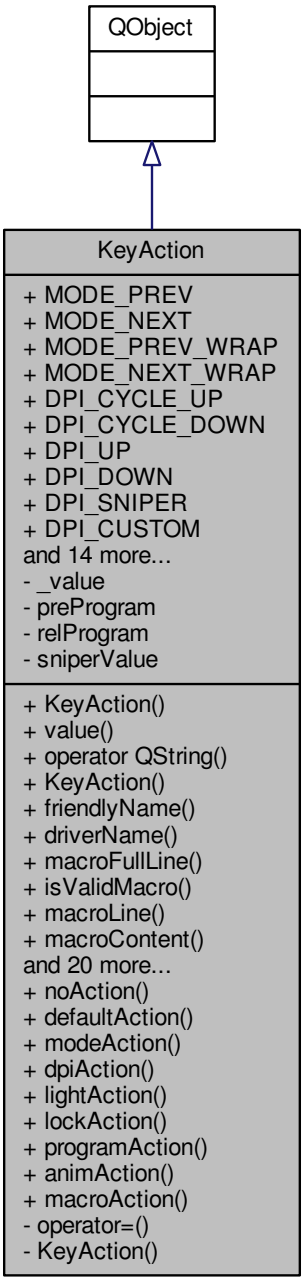
The documentation for this struct was generated from the following file:

- src/ckb/keymap.h

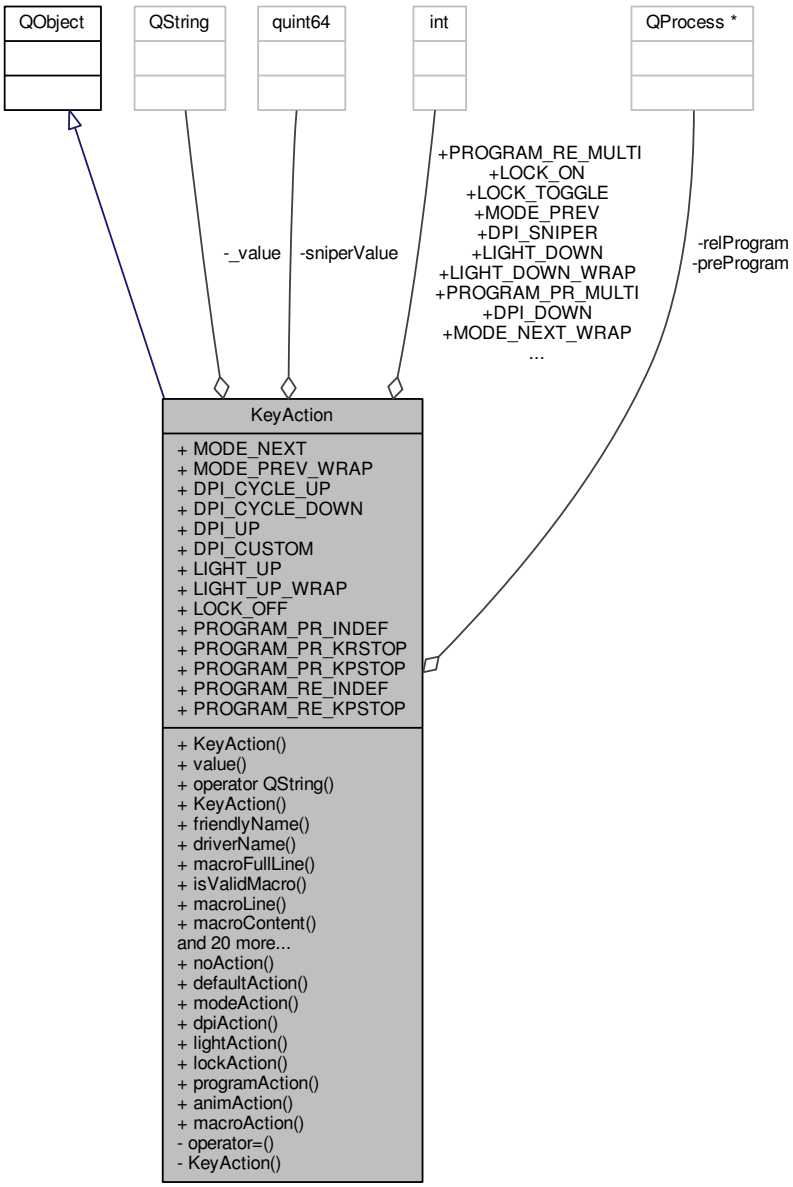
12.42 KeyAction Class Reference

`#include <src/ckb/keyaction.h>`

Inheritance diagram for KeyAction:



Collaboration diagram for KeyAction:



Public Types

- enum Type { UNBOUND, NORMAL, SPECIAL }

Public Member Functions

- KeyAction (const QString &action, QObject *parent=0)
- QString value () const
- operator QString () const
- KeyAction (QObject *parent=0)
- QString friendlyName (const KeyMap &map) const

- QString driverName () const
- QString macroFullLine () const

  macroFullLine If a macro command and a macro definition exists for the given key, returns the complete string except the leading "$" (the $ may confuse some caller).
- bool isValidMacro () const

  isValidMacro checks whether a keyAction contains a valid macro. This is done easily: If the macro action starts with $macro: and has five elements, delimited by ":", we may assume, that is a structural correct macro action. If it has 4 entries only, it is an older definition and ok also.
- QStringList macroLine () const

  macroLine returns all interresting content for a macro definition.
- QString macroContent () const

  macroContent returns the macro key definition only (the second part of the macro action).
- QString macroTiming () const

  macroTiming returns the macro key definition with original timing infos (the fifth and up to now last part of the macro action). If the implementation does not know anything about delays and has no 5th part, return first part.
- void macroDisplay ()

  Debug output for invalid macro Definitions.
- Type type () const
- bool isUnbound () const
- bool isNormal () const
- bool isSpecial () const
- bool isMedia () const
- bool isProgram () const
- bool isAnim () const
- bool isMacro () const
- bool isDPI () const
- bool isMouse () const
- QString specialInfo (int &parameter) const
- int programInfo (QString &onPress, QString &onRelease) const
- int dpiInfo (QPoint &custom) const
- QUuid animInfo (bool &onlyOnce, bool &stopOnRelease) const
- void keyEvent (KbBind *bind, bool down)
- void keyRelease (KbBind *bind)
- void adjustDisplay ()
- ~KeyAction ()

Static Public Member Functions

- static QString noAction ()
- static QString defaultAction (const QString &key, KeyMap::Model model)
- static QString modeAction (int mode)
- static QString dpiAction (int level, int customX=0, int customY=0)
- static QString lightAction (int type=LIGHT_UP_WRAP)
- static QString lockAction (int type=LOCK_TOGGLE)
- static QString programAction (const QString &onPress, const QString &onRelease, int stop)
- static QString animAction (const QUuid &guid, bool onlyOnce, bool stopOnRelease)
- static QString macroAction (QString macroDef)

  well documented in cpp file

Static Public Attributes

- static const int MODE_PREV = -2
- static const int MODE_NEXT = -1
- static const int MODE_PREV_WRAP = -4
- static const int MODE_NEXT_WRAP = -3
- static const int DPI_CYCLE_UP = -4
- static const int DPI_CYCLE_DOWN = -3
- static const int DPI_UP = -2
- static const int DPI_DOWN = -1
- static const int DPI_SNIPER = 0
- static const int DPI_CUSTOM = 6
- static const int LIGHT_UP = 0
- static const int LIGHT_DOWN = 1
- static const int LIGHT_UP_WRAP = 2
- static const int LIGHT_DOWN_WRAP = 3
- static const int LOCK_TOGGLE = 0
- static const int LOCK_ON = 1
- static const int LOCK_OFF = 2
- static const int PROGRAM_PR_MULTI = 0x04
- static const int PROGRAM_PR_INDEF = 0x00
- static const int PROGRAM_PR_KRSTOP = 0x01
- static const int PROGRAM_PR_KPSTOP = 0x02
- static const int PROGRAM_RE_MULTI = 0x40
- static const int PROGRAM_RE_INDEF = 0x00
- static const int PROGRAM_RE_KPSTOP = 0x20

Private Member Functions

- void operator= (const KeyAction &rhs)

  ccMSC: Don't copy key actions (the old one needs to be deleted first) frickler24: statement left as described, but copying is done in KbBind copy constructor
- KeyAction (const KeyAction &rhs)

Private Attributes

- QString _value
- QProcess * preProgram
- QProcess * relProgram
- quint64 sniperValue

12.42.1 Detailed Description

Definition at line 13 of file keyaction.h.

12.42.2 Member Enumeration Documentation

12.42.2.1 enum KeyAction::Type

Enumerator

UNBOUND

NORMAL

SPECIAL

Definition at line 146 of file keyaction.h.

```
146                 {
147         UNBOUND,
148         NORMAL,
149         SPECIAL,
150     };
```

12.42.3 Constructor & Destructor Documentation

12.42.3.1 KeyAction::KeyAction (const QString & *action,* QObject * *parent* = 0)

Definition at line 22 of file keyaction.cpp.

```
23     : QObject(parent), _value(action), preProgram(0),
      relProgram(0), sniperValue(0)
24 {
25 }
```

12.42.3.2 KeyAction::KeyAction (QObject * *parent* = 0) [explicit]

Definition at line 27 of file keyaction.cpp.

```
28     : QObject(parent), _value(""), preProgram(0),
      relProgram(0), sniperValue(0)
29 {
30 }
```

12.42.3.3 KeyAction::~KeyAction ( )

Definition at line 32 of file keyaction.cpp.

References preProgram, and relProgram.

```
32                      {
33     // Clean up processes
34     if(preProgram){
35         preProgram->kill();
36         delete preProgram;
37     }
38     if(relProgram){
39         relProgram->kill();
40         delete relProgram;
41     }
42 }
```

12.42.3.4 KeyAction::KeyAction (const KeyAction & *rhs*) [inline], [private]

Definition at line 187 of file keyaction.h.

```
187 : QObject() {}
```

12.42.4 Member Function Documentation

12.42.4.1 void KeyAction::adjustDisplay ( )

Definition at line 464 of file keyaction.cpp.

Referenced by keyEvent().

```
464                             {
465 #ifdef USE_LIBX11
466     // Try to get the current display from the X server
467     char* display_name = XDisplayName(NULL);
468     if(!display_name)
469         return;
470     Display* display = XOpenDisplay(display_name);
471     if(!display)
472         return;
473     char* display_string = DisplayString(display);
474     if(!display_string || strlen(display_string) == 0){
475         XCloseDisplay(display);
476         return;
477     }
478     size_t envstr_size = strlen(display_string) + 4;
479     char* envstr = new char[envstr_size];
480     strncpy(envstr, display_string, envstr_size);
481     envstr[envstr_size - 1] = 0;
482 
483     Window root_window = XRootWindow(display, DefaultScreen(display));
484     Window root_window_ret, child_window_ret, window;
485     XWindowAttributes attr;
486     int root_x, root_y, win_x, win_y;
487     unsigned int mask_ret;
488 
489     // Find the screen which currently has the mouse
490     XQueryPointer(display, root_window, &root_window_ret, &child_window_ret, &root_x, &root_y, &win_x, &
    win_y, &mask_ret);
491     if(child_window_ret == (Window)NULL)
492         window = root_window_ret;
493     else
494         window = child_window_ret;
495     XGetWindowAttributes(display, window,  &attr);
496 
497     char* ptr = strchr(envstr, ':');
498     if(ptr){
499         ptr = strchr(ptr, '.');
500         if(ptr)
501             *ptr = '\0';
502         char buf[16];
503         snprintf(buf, sizeof(buf), ".%i", XScreenNumberOfScreen(attr.screen));
504         strncat(envstr, buf, envstr_size - 1 - strlen(envstr));
505 
506         // Update environment variable
507         setenv("DISPLAY", envstr, 1);
508     }
509 
510     delete[] envstr;
511     XCloseDisplay(display);
512 #endif
513 }
```

Here is the caller graph for this function:

12.42.4.2 QString KeyAction::animAction (const QUuid & *guid*, bool *onlyOnce*, bool *stopOnRelease*) [static]

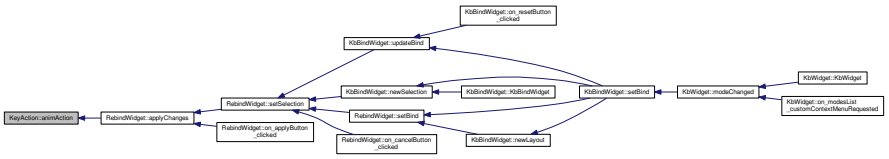
Definition at line 184 of file keyaction.cpp.

References ANIM_KRSTOP, and ANIM_ONCE.

Referenced by RebindWidget::applyChanges().

```
184                                                                                 {
185     int flags = (onlyOnce ? ANIM_ONCE : 0) | (stopOnRelease ?
    ANIM_KRSTOP : 0);
186     return "$anim:" + guid.toString() + QString("+%1").arg(flags);
187 }
```

Here is the caller graph for this function:



12.42.4.3 QUuid KeyAction::animInfo (bool & *onlyOnce,* bool & *stopOnRelease*) const

Definition at line 225 of file keyaction.cpp.

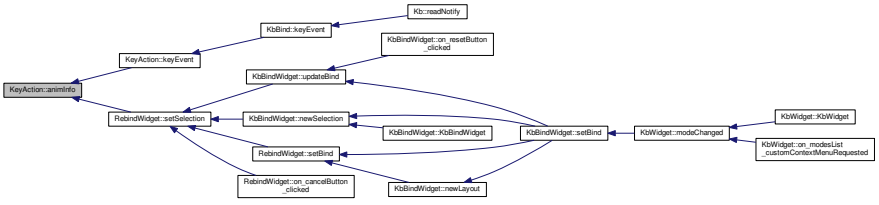
References _value, ANIM_KRSTOP, ANIM_ONCE, and isAnim().

Referenced by keyEvent(), and RebindWidget::setSelection().

```
225                                                                 {
226     if(!isAnim())
227         return QUuid();
228     QString param = _value.mid(6);
229     QStringList split = param.split("+");
230     if(split.length() < 2)
231         return QUuid();
232     QUuid id = split[0];
233     int flags = split[1].toInt();
234     onlyOnce = !!(flags & ANIM_ONCE);
235     stopOnRelease = !!(flags & ANIM_KRSTOP);
236     return id;
237 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.42.4.4 QString KeyAction::defaultAction (const QString & *key,* KeyMap::Model *model*) [static]

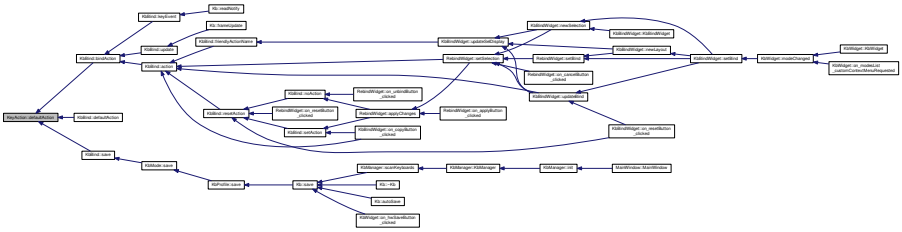
Definition at line 44 of file keyaction.cpp.

References KeyMap::GLAIVE, and KeyMap::HARPOON.

Referenced by KbBind::bindAction(), KbBind::defaultAction(), and KbBind::save().

```
44                                                                          {
45      // G1-G18 are unbound by default
46      if(key.length() >= 2 && key[0] == 'g'
47          && ((key.length() == 2 && key[1] >= '0' && key[1] <= '9')
48          || (key.length() == 3 && key[1] == '1' && key[2] >= '0' && key[2] <= '8')))
49          return "";
50      // So are thumbgrid buttons
51      if(key.startsWith("thumb"))
52          return "";
53      // TODO: default action for MR
54      if(key == "mr")
55          return "";
56      // M1-M3 switch modes
57      if(key == "m1")
58          return "$mode:0";
59      if(key == "m2")
60          return "$mode:1";
61      if(key == "m3")
62          return "$mode:2";
63      // Brightness and Win Lock are their own functions
64      if(key == "light")
65          return "$light:2";
66      if(key == "lock")
67          return "$lock:0";
68      // DPI buttons
69      if(key == "dpiup"){
70          if(model == KeyMap::HARPOON ||
71         model == KeyMap::GLAIVE){
72          return "$dpi:-4";
73          }
74      return "$dpi:-2";
75      }
76      if(key == "dpidn")
77          return "$dpi:-1";
78      if(key == "sniper")
79          return "$dpi:0";
80      // Everything else is a standard keypress
81      return key;
82  }
```

Here is the caller graph for this function:



12.42.4.5 QString KeyAction::dpiAction (int *level*, int *customX* = 0, int *customY* = 0) [static]

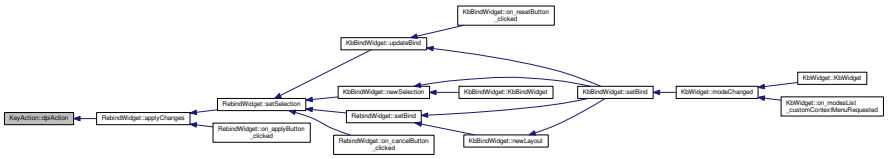
Definition at line 162 of file keyaction.cpp.

References DPI_CUSTOM.

Referenced by RebindWidget::applyChanges().

```
162                                                                         {
163      QString action = tr("$dpi:%1").arg(level);
164      if(level == DPI_CUSTOM)
165          action += tr("+%1+%2").arg(customX).arg(customY);
166      return action;
167  }
```

Here is the caller graph for this function:



12.42.4.6 int KeyAction::dpiInfo (QPoint & *custom*) const

Definition at line 211 of file keyaction.cpp.

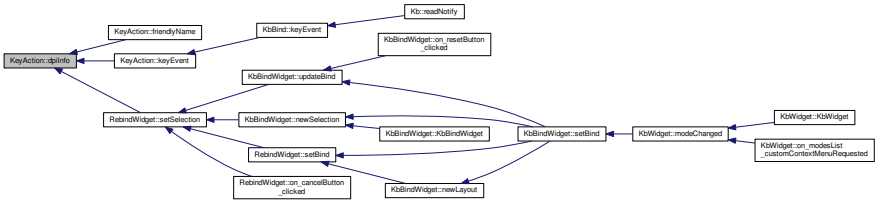
References _value, DPI_CUSTOM, and isDPI().

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

```
211                                         {
212     if(!isDPI())
213         return 0;
214     QString param = _value.mid(5);
215     QStringList lxy = param.split("+");
216     int level = lxy[0].toInt();
217     if(level == DPI_CUSTOM){
218         if(lxy.length() != 3)
219             return 0;
220         custom = QPoint(lxy[1].toInt(), lxy[2].toInt());
221     }
222     return level;
223 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.42.4.7 QString KeyAction::driverName ( ) const

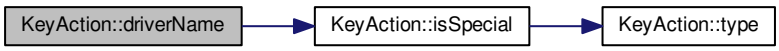
Definition at line 239 of file keyaction.cpp.

References _value, and isSpecial().

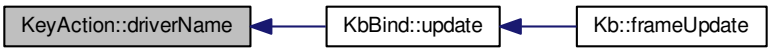
Referenced by KbBind::update().

```
239                                     {
240     if(isSpecial())
241         return "";
242     return _value;
243 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.8 QString KeyAction::friendlyName (const KeyMap & *map*) const

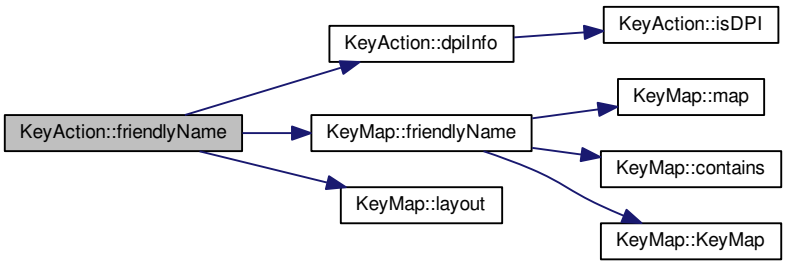
Definition at line 84 of file keyaction.cpp.

References _value, DPI_CUSTOM, DPI_CYCLE_DOWN, DPI_CYCLE_UP, DPI_DOWN, DPI_SNIPER, DPI_UP, dpiInfo(), KeyMap::friendlyName(), KeyMap::layout(), LIGHT_DOWN, LIGHT_DOWN_WRAP, LIGHT_UP, LIGHT-_UP_WRAP, LOCK_OFF, LOCK_ON, LOCK_TOGGLE, MODE_NEXT, MODE_NEXT_WRAP, MODE_PREV, and MODE_PREV_WRAP.

```
84                                                      {
85     if(_value.isEmpty())
86         return "Unbound";
87     QStringList parts = _value.split(":");
88     QString prefix = parts[0];
89     if(parts.length() < 2){
90         KeyMap::Layout layout = map.layout();
91         QString name = KeyMap::friendlyName(_value, layout);
92         if(name.isEmpty())
93             return "(Unknown)";
94         return name;
95     }
96     int suffix = parts[1].toInt();
97     if(prefix == "$mode"){
98         switch(suffix){
99         case MODE_PREV:
100         case MODE_PREV_WRAP:
101             return "Switch to previous mode";
102         case MODE_NEXT:
103         case MODE_NEXT_WRAP:
104             return "Switch to next mode";
105         default:
106             return tr("Switch to mode %1").arg(suffix + 1);
107         }
108     } else if(prefix == "$dpi"){
109         // Split off custom parameters (if any)
110         int level = parts[1].split("+")[0].toInt();
```

```
111         switch(level){
112         case DPI_CYCLE_UP:
113             return "DPI cycle up";
114         case DPI_CYCLE_DOWN:
115             return "DPI cycle down";
116         case DPI_UP:
117             return "DPI up";
118         case DPI_DOWN:
119             return "DPI down";
120         case DPI_SNIPER:
121             return "Sniper";
122         case DPI_CUSTOM:{
123             QPoint xy;
124             dpiInfo(xy);
125             return tr("DPI: %1, %2").arg(xy.x()).arg(xy.y());
126         }
127         default:
128             return tr("DPI stage %1").arg(level);
129         }
130     } else if(prefix == "$light"){
131         switch(suffix){
132         case LIGHT_UP:
133         case LIGHT_UP_WRAP:
134             return "Brightness up";
135         case LIGHT_DOWN:
136         case LIGHT_DOWN_WRAP:
137             return "Brightness down";
138         }
139     } else if(prefix == "$lock"){
140         switch(suffix){
141         case LOCK_TOGGLE:
142             return "Toggle Windows lock";
143         case LOCK_ON:
144             return "Windows lock on";
145         case LOCK_OFF:
146             return "Windows lock off";
147         }
148     } else if(prefix == "$anim"){
149         return "Start animation";
150     } else if(prefix == "$program"){
151         return "Launch program";
152     } else if(prefix == "$macro"){
153         return "Send G-key macro";
154     }
155     return "(Unknown)";
156 }
```

Here is the call graph for this function:



12.42.4.9 bool KeyAction::isAnim ( ) const [inline]

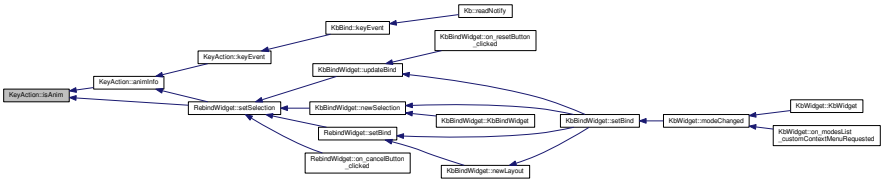
Definition at line 159 of file keyaction.h.

References _value.

Referenced by animInfo(), and RebindWidget::setSelection().

```
159 { return _value.startsWith("$anim:"); }
```

Here is the caller graph for this function:



12.42.4.10 bool KeyAction::isDPI ( ) const [inline]

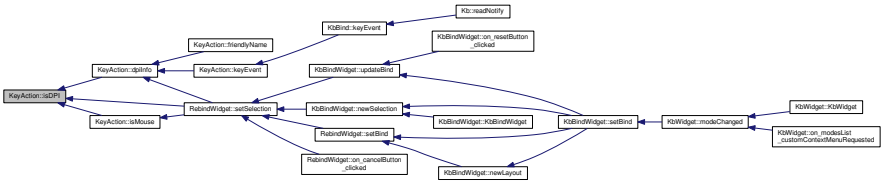
Definition at line 162 of file keyaction.h.

References _value.

Referenced by dpiInfo(), isMouse(), and RebindWidget::setSelection().

```
162 { return _value.startsWith("$dpi:"); }
```

Here is the caller graph for this function:



12.42.4.11 bool KeyAction::isMacro ( ) const [inline]

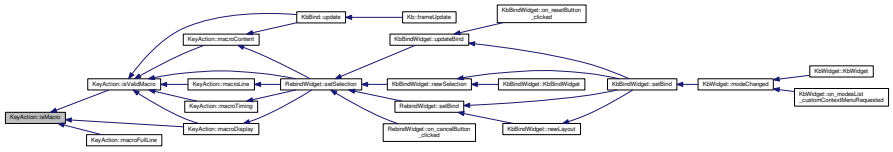
Definition at line 160 of file keyaction.h.

References _value.

Referenced by isValidMacro(), macroDisplay(), and macroFullLine().

```
160 { return _value.startsWith("$macro:"); }
```

Here is the caller graph for this function:



12.42.4.12 bool KeyAction::isMedia ( ) const [inline]

Definition at line 156 of file keyaction.h.

References _value.

```
156 { return _value == "mute" || _value == "volup" || _value == "voldn" ||
      _value == "stop" || _value == "prev" || _value == "play" ||
      _value == "next"; }
```

12.42.4.13 bool KeyAction::isMouse ( ) const [inline]

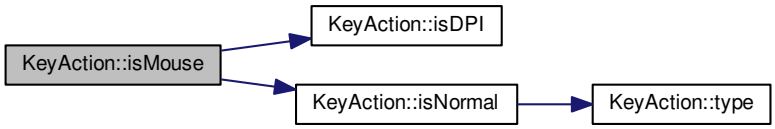
Definition at line 163 of file keyaction.h.

References _value, isDPI(), and isNormal().

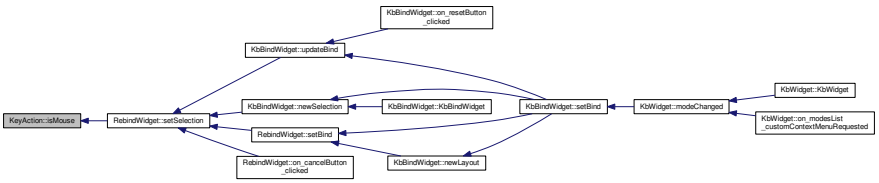
Referenced by RebindWidget::setSelection().

```
163 { return (isNormal() && (_value.startsWith("mouse") || _value.startsWith("wheel"))) ||
      isDPI(); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.14 bool KeyAction::isNormal ( ) const [inline]

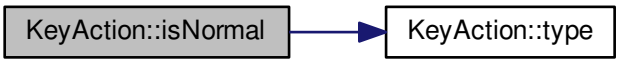
Definition at line 153 of file keyaction.h.

References NORMAL, and type().

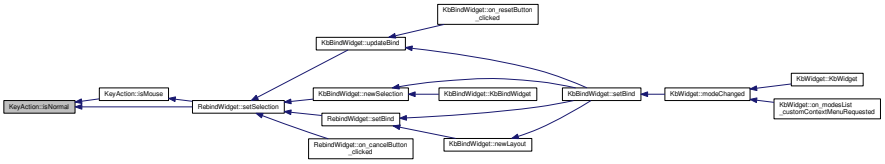
Referenced by isMouse(), and RebindWidget::setSelection().

```
153 { return type() == NORMAL; }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.15 bool KeyAction::isProgram ( ) const [inline]

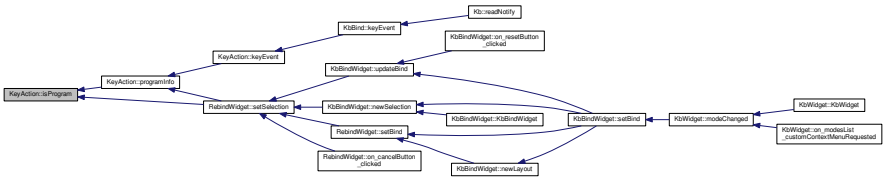
Definition at line 158 of file keyaction.h.

References _value.

Referenced by programInfo(), and RebindWidget::setSelection().

```
158 { return _value.startsWith("$program:"); }
```

Here is the caller graph for this function:



12.42.4.16 bool KeyAction::isSpecial ( ) const [inline]

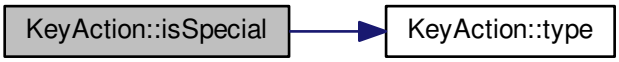
Definition at line 154 of file keyaction.h.

References SPECIAL, and type().

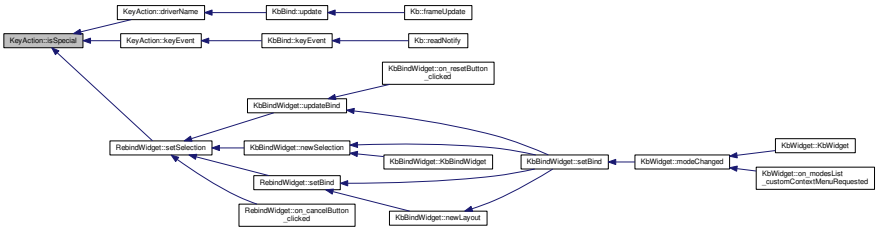
Referenced by driverName(), keyEvent(), and RebindWidget::setSelection().

```
154 { return type() == SPECIAL; }
```

Here is the call graph for this function:



Here is the caller graph for this function:



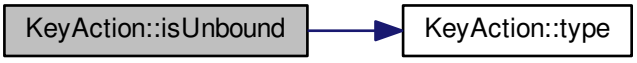
12.42.4.17 bool KeyAction::isUnbound ( ) const [inline]

Definition at line 152 of file keyaction.h.

References type(), and UNBOUND.

```
152 { return type() == UNBOUND; }
```

Here is the call graph for this function:



12.42.4.18 bool KeyAction::isValidMacro ( ) const [inline]

Returns

bool as true iff the macro definition contains all four elements.

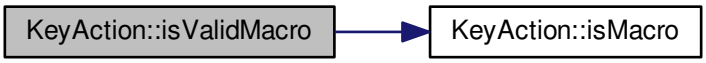
Definition at line 55 of file keyaction.h.

References _value, and isMacro().

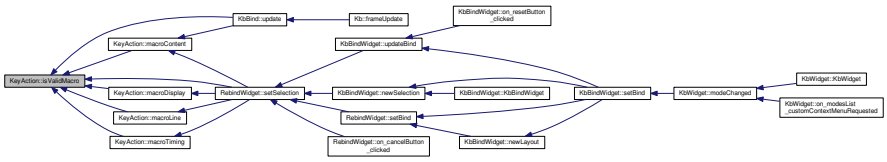
Referenced by macroContent(), macroDisplay(), macroLine(), macroTiming(), RebindWidget::setSelection(), and KbBind::update().

```
55                                      {
56         if (isMacro()) {
57             QStringList ret;
58             ret =_value.split(":");
59             return ((ret.count() >= 4) && (ret.count() <= 5));
60         } else {
61             return false;
62         }
63     }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.19 void KeyAction::keyEvent (KbBind * *bind*, bool *down*)

Definition at line 245 of file keyaction.cpp.

References _value, adjustDisplay(), anim, animInfo(), KbPerf::baseDpiIdx(), KbProfile::currentMode(), Kb::currentProfile(), KbBind::devParent(), KbLight::dimming(), KbPerf::DPI_COUNT, DPI_CUSTOM, DPI_CYCLE_DOWN, DPI_CYCLE_UP, DPI_DOWN, DPI_SNIPER, DPI_UP, KbPerf::dpiCycleDown(), KbPerf::dpiCycleUp(), KbPerf::dpiDown(), dpiInfo(), KbPerf::dpiUp(), KbLight::findAnim(), KbProfile::indexOf(), KbAnim::isActive(), isSpecial(), KbBind::light(), LIGHT_DOWN, LIGHT_DOWN_WRAP, LIGHT_UP, LIGHT_UP_WRAP, LOCK_OFF, LOCK_ON, LOCK_TOGGLE, KbLight::MAX_DIM, MODE_NEXT, MODE_NEXT_WRAP, MODE_PREV, MODE_PREV_WRAP, KbProfile::modeCount(), KbProfile::modes(), KbBind::perf(), KbPerf::popDpi(), preProgram, PROGRAM_PR_KPSTOP, PROGRAM_PR_KRSTOP, PROGRAM_PR_MULTI, PROGRAM_RE_KPSTOP, PROGRAM_RE_MULTI, programInfo(), KbPerf::pushDpi(), KbPerf::pushSniper(), relProgram, Kb::setCurrentMode(), sniperValue, KbAnim::stop(), KbAnim::trigger(), and KbBind::winLock().

Referenced by KbBind::keyEvent().

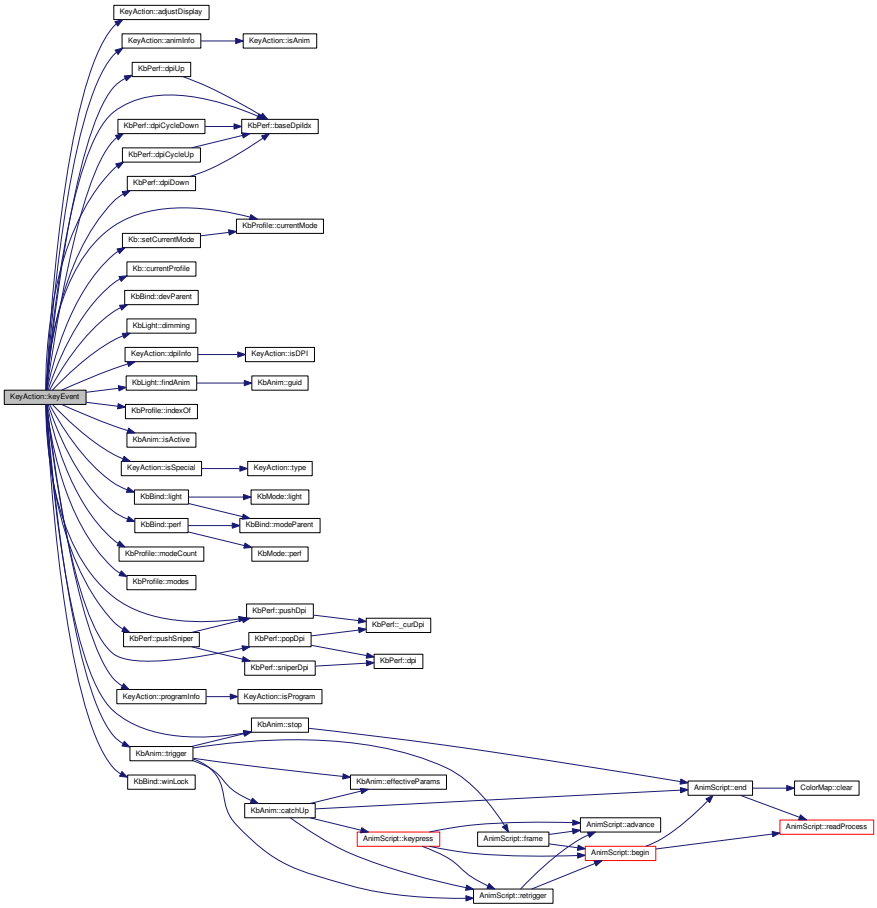
```
245                                                 {
246     // No need to respond to standard actions
247     if(!isSpecial())
248         return;
249     QStringList parts = _value.split(":");
250     if(parts.length() < 2)
251         return;
252     QString prefix = parts[0];
253     int suffix = parts[1].toInt();
254     if(prefix == "$mode"){
255         if(!down)
256             return;
257         // Change mode
258         Kb* device = bind->devParent();
259         KbProfile* currentProfile = device->currentProfile();
```

```
260         int mode = currentProfile->indexOf(currentProfile->currentMode());
261         int modeCount = currentProfile->modeCount();
262         switch(suffix){
263         case MODE_PREV_WRAP:
264             mode--;
265             if(mode < 0)
266                 mode = modeCount - 1;
267             break;
268         case MODE_NEXT_WRAP:
269             mode++;
270             if(mode >= modeCount)
271                 mode = 0;
272             break;
273         case MODE_PREV:
274             mode--;
275             break;
276         case MODE_NEXT:
277             mode++;
278             break;
279         default:
280             // Absolute
281             mode = suffix;
282             break;
283         }
284         if(mode < 0 || mode >= modeCount)
285             return;
286         device->setCurrentMode(currentProfile->modes()[mode]);
287     } else if(prefix == "$dpi"){
288         KbPerf* perf = bind->perf();
289         int level = parts[1].split("+")[0].toInt();
290         switch(level){
291         case DPI_CYCLE_UP:
292             if(!down)
293                 return;
294             perf->dpiCycleUp();
295             break;
296         case DPI_CYCLE_DOWN:
297             if(!down)
298                 return;
299             perf->dpiCycleDown();
300             break;
301         case DPI_UP:
302             if(!down)
303                 return;
304             perf->dpiUp();
305             break;
306         case DPI_DOWN:
307             if(!down)
308                 return;
309             perf->dpiDown();
310             break;
311         case DPI_SNIPER:
312             if(down)
313                 sniperValue = perf->pushSniper();
314             else {
315                 perf->popDpi(sniperValue);
316                 sniperValue = 0;
317             }
318             break;
319         case DPI_CUSTOM:{
320             QPoint xy;
321             dpiInfo(xy);
322             if(xy.x() <= 0 || xy.y() <= 0)
323                 break;
324             if(down)
325                 sniperValue = perf->pushDpi(xy);
326             else {
327                 perf->popDpi(sniperValue);
328                 sniperValue = 0;
329             }
330             break;
331         }
332         default:
333             if(level < 1 || level >= KbPerf::DPI_COUNT
334                     || !down)
335                 return;
336             perf->baseDpiIdx(level);
337             break;
338         }
339     } else if(prefix == "$light"){
340         if(!down)
341             return;
342         // Change brightness
343         KbLight* light = bind->light();
344         int dim = light->dimming();
345         switch(suffix){
346         case LIGHT_UP:
```

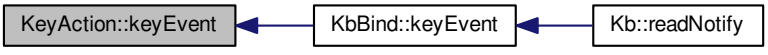
```
347             if(dim > 0)
348                 dim--;
349             break;
350         case LIGHT_DOWN:
351             if(dim < KbLight::MAX_DIM)
352                 dim++;
353             break;
354         case LIGHT_UP_WRAP:
355             dim--;
356             if(dim < 0)
357                 dim = KbLight::MAX_DIM;
358             break;
359         case LIGHT_DOWN_WRAP:
360             dim++;
361             if(dim > KbLight::MAX_DIM)
362                 dim = 0;
363             break;
364         }
365         light->dimming(dim);
366     } else if(prefix == "$lock"){
367         if(!down)
368             return;
369         // Change win lock
370         switch(suffix){
371         case LOCK_TOGGLE:
372             bind->winLock(!bind->winLock());
373             break;
374         case LOCK_ON:
375             bind->winLock(true);
376             break;
377         case LOCK_OFF:
378             bind->winLock(false);
379             break;
380         }
381     } else if(prefix == "$anim"){
382         // Start animation
383         bool onlyOnce = false, stopOnRelease = false;
384         QUuid id = animInfo(onlyOnce, stopOnRelease);
385         KbAnim* anim = bind->light()->findAnim(id);
386         if(!anim)
387             return;
388         if(down){
389             if(!onlyOnce || !anim->isActive())
390                 // If "only once" is enabled, don't start the animation when it's already running
391                 anim->trigger(QDateTime::currentMSecsSinceEpoch(), true);
392         } else if(stopOnRelease){
393             // Key released - stop animation
394             anim->stop();
395         }
396     } else if(prefix == "$program"){
397         // Launch program
398         QString onPress, onRelease;
399         int stop = programInfo(onPress, onRelease);
400         // Stop running programs based on setting
401         QProcess* process = 0;
402         if(down){
403             if(stop & PROGRAM_PR_KPSTOP){
404                 process = preProgram;
405                 if(process)
406                     process->kill();
407                 process = 0;
408             }
409             if(stop & PROGRAM_RE_KPSTOP)
410                 process = relProgram;
411         } else {
412             if(stop & PROGRAM_PR_KRSTOP)
413                 process = preProgram;
414         }
415         if(process)
416             process->kill();
417         // Launch new process if requested
418         QString& program = down ? onPress : onRelease;
419         if(program.isEmpty())
420             return;
421         // Check if the program is running already. If so, don't start it again.
422         process = down ? preProgram : relProgram;
423         if(process){
424             if(process->state() == QProcess::NotRunning)
425                 delete process;
426             else
427                 return;
428         }
429
430         // Adjust the selected display.
431         adjustDisplay();
432
433         // Start the program. Wrap it around sh to parse arguments.
```

```
434         if((down && (stop & PROGRAM_PR_MULTI))
435                 || (!down && (stop & PROGRAM_RE_MULTI))){
436             // Multiple instances allowed? Start detached process
437             QProcess::startDetached("sh", QStringList() << "-c" << program);
438         } else {
439             process = new QProcess(this);
440             process->start("sh", QStringList() << "-c" << program);
441             if(down)
442                 preProgram = process;
443             else
444                 relProgram = process;
445         }
446     } else if (prefix == "$macro") {
447         // Do nothing, because all work is done by the keyboard itself.
448         // For now, there is no reason to react on G-key press or release.
449         // If u find some reason, then here is the place for it.
450     }
451 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.20 void KeyAction::keyRelease (KbBind * *bind*)

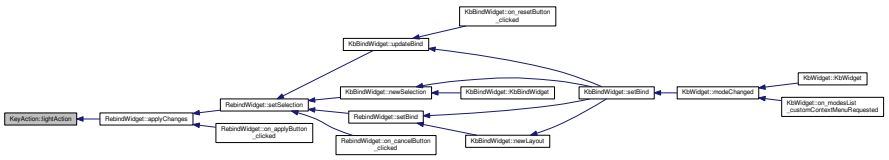
12.42.4.21 QString KeyAction::lightAction (int *type* = LIGHT_UP_WRAP) [static]

Definition at line 169 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
169                                          {
170     return QString("$light:%1").arg(type);
171 }
```

Here is the caller graph for this function:



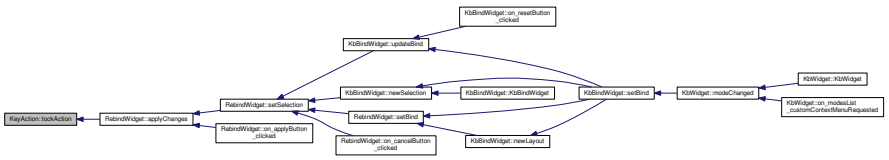
12.42.4.22 QString KeyAction::lockAction (int *type* = LOCK_TOGGLE) [static]

Definition at line 173 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
173                                         {
174     return QString("$lock:%1").arg(type);
175 }
```

Here is the caller graph for this function:



12.42.4.23 QString KeyAction::macroAction (QString *macroDef*) [static]

KeyAction::macroAction is called when applying changes on a macro definition. macroAction ist called while being in the macro pane and clicking Apply with something in the Macro Text Box. It tags that input with "$macro:" for further recognition.

Parameters

macroDef	holds the String containing parts 2-5 of a complete macro definition.

Returns

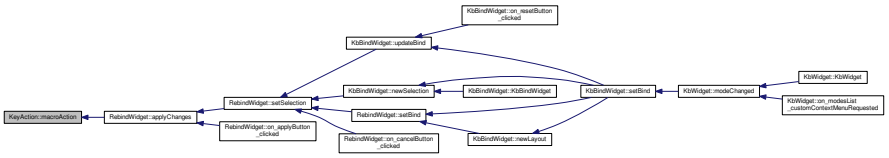
QString holding the complete G-Key macro definition (parts 1-5)

Definition at line 523 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
523                                                 {
524     return QString ("$macro:%1").arg(macroDef);
525 }
```

Here is the caller graph for this function:



12.42.4.24 QString KeyAction::macroContent ( ) const [inline]

Returns

QString macroContent

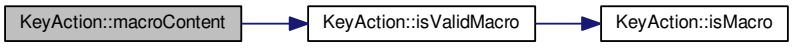
Definition at line 86 of file keyaction.h.

References _value, and isValidMacro().

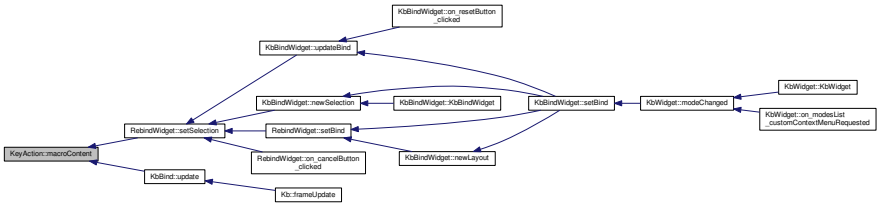
Referenced by RebindWidget::setSelection(), and KbBind::update().

```
86                                          {
87         // return isValidMacro() ? _value.split(":")[1].replace(QRegExp("=\\d+"), "") : ""; ///< Is used if
       we have ckb without delay handling
88         return isValidMacro() ? _value.split(":")[1] : "";
89     }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.25 void KeyAction::macroDisplay ( )

KeyAction::macroDisplay is just for debugging. It shows the content of the key action and some other info.

General Info on KeyAction::_value for macros: That string consists of 4 elements, all delimited by ":".

1. Macro command indicator "$macro:"
2. Macro Key Definition (coming from pteMacroBox): This sequence will program the keyboard and is hardly readable
3. Readable Macro String: This is displayed in pteMacroText
4. Readable Macro Comment:This is displayed in pteMacroComment
5. completely unreadable original macro information with timing values

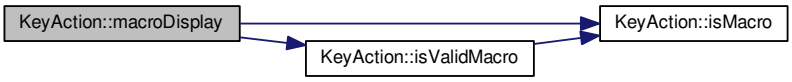
Definition at line 456 of file keyaction.cpp.

References _value, isMacro(), and isValidMacro().

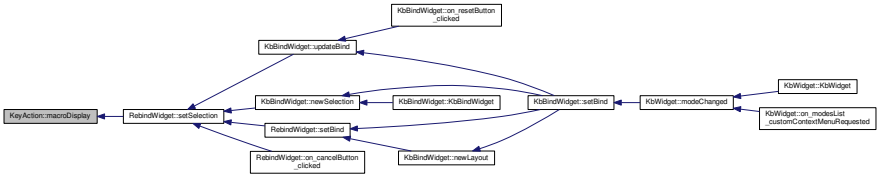
Referenced by RebindWidget::setSelection().

```
456                                {
457     qDebug() << "isMacro returns" << (isMacro() ? "true" : "false");
458     qDebug() << "isValidMacro returns" << (isValidMacro() ? "true" : "false");
459     QStringList ret =_value.split(":");
460     qDebug() << "Macro definition conains" << ret.count() << "elements";
461     qDebug() << "Macro definition is" << _value;
462 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.26 QString KeyAction::macroFullLine ( ) const [inline]

Returns

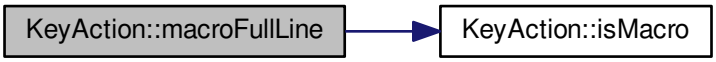
QString All 5 parts are returned in one QString. If no definition exists, return ""

Definition at line 43 of file keyaction.h.

References _value, and isMacro().

```
43                                          {
44          return isMacro() ? _value.right(_value.length()-1) : "";
45      }
```

Here is the call graph for this function:



12.42.4.27 QStringList KeyAction::macroLine ( ) const [inline]

Returns

QStringList returns the Macro Key Definition, Readble Macro String, Readable Macro Comment and the original timing information (if it exists as a 5th part) as QStringList.

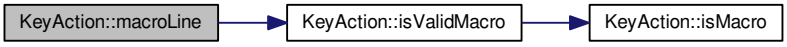
Definition at line 73 of file keyaction.h.

References _value, and isValidMacro().

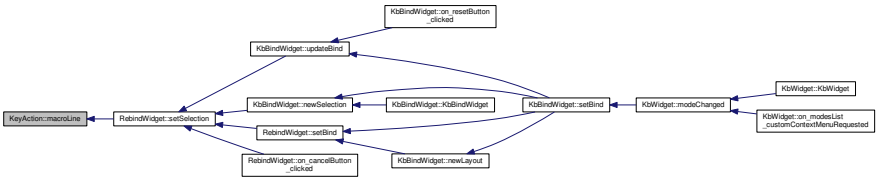
Referenced by RebindWidget::setSelection().

```
73                                          {
74          if (isValidMacro()) {
75              QStringList ret =_value.split(":");
76              ret.removeFirst();
77              return ret;
78          } else return QStringList();
79      }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.42.4.28 QString KeyAction::macroTiming ( ) const [inline]

Returns

QString macroTiming

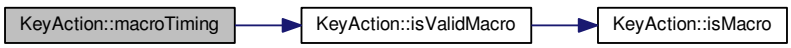
Definition at line 98 of file keyaction.h.

References _value, and isValidMacro().

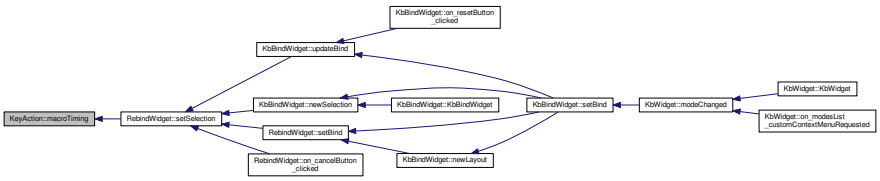
Referenced by RebindWidget::setSelection().

```
98                                      {
99          if (isValidMacro()) {
100             QStringList rval = _value.split(":");
101             return (rval.length() == 4)? rval[1] : rval[4];
102         }
103         return QString("");
104     }
```

Here is the call graph for this function:



Here is the caller graph for this function:



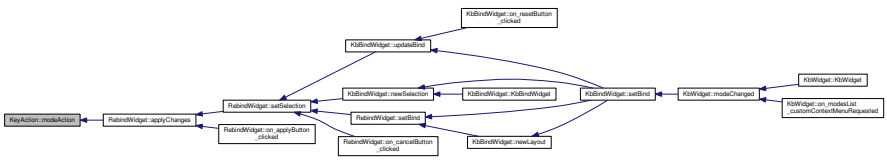
12.42.4.29 QString KeyAction::modeAction (int *mode*) [static]

Definition at line 158 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
158                                          {
159     return QString("$mode:%1").arg(mode);
160 }
```

Here is the caller graph for this function:



12.42.4.30 static QString KeyAction::noAction ( ) [inline], [static]

Definition at line 25 of file keyaction.h.

```
25 { return ""; }
```

12.42.4.31 KeyAction::operator QString ( ) const [inline]

Definition at line 20 of file keyaction.h.

References _value.

```
20 { return _value; }
```

12.42.4.32 void KeyAction::operator= (const KeyAction & *rhs*) [inline], [private]

Definition at line 186 of file keyaction.h.

```
186 {}
```

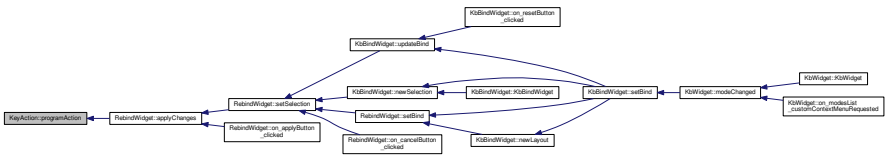
12.42.4.33 QString KeyAction::programAction (const QString & *onPress*, const QString & *onRelease*, int *stop*) [static]

Definition at line 177 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
177                                                                                                 {
178     // URL-encode the commands and place them in the string (":" and "+" are both replaced, so they won't
       interfere)
179     return "$program:" + QString::fromUtf8(QUrl::toPercentEncoding(onPress.trimmed())) + "+" +
       QString::fromUtf8(QUrl::toPercentEncoding(onRelease.trimmed())) + QString("+%1").arg(stop);
180 }
```

Here is the caller graph for this function:



12.42.4.34 int KeyAction::programInfo (QString & *onPress*, QString & *onRelease*) const

Definition at line 199 of file keyaction.cpp.

References _value, and isProgram().

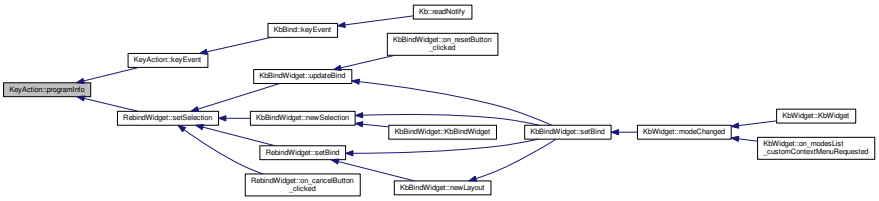
Referenced by keyEvent(), and RebindWidget::setSelection().

```
199                                                                                 {
200     if(!isProgram())
201         return 0;
202     QString param = _value.mid(9);
203     QStringList programs = param.split("+");
```

```
204     if(programs.length() != 3)
205         return 0;
206     onPress = QUrl::fromPercentEncoding(programs[0].toUtf8());
207     onRelease = QUrl::fromPercentEncoding(programs[1].toUtf8());
208     return programs[2].toInt();
209 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.42.4.35 QString KeyAction::specialInfo (int & *parameter*) const

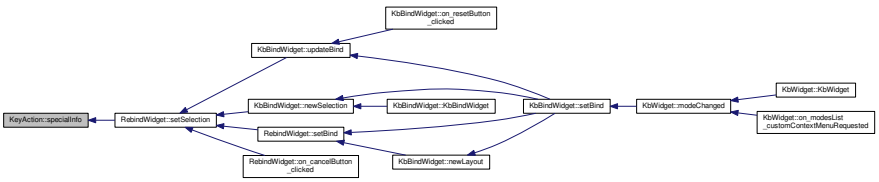
Definition at line 189 of file keyaction.cpp.

References _value.

Referenced by RebindWidget::setSelection().

```
189                                                 {
190     QStringList list = _value.split(":");
191     if(list.length() < 2){
192         parameter = INT_MIN;
193         return "";
194     }
195     parameter = list[1].toInt();
196     return list[0].replace("$", "");
197 }
```

Here is the caller graph for this function:



12.42.4.36 KeyAction::Type KeyAction::type ( ) const

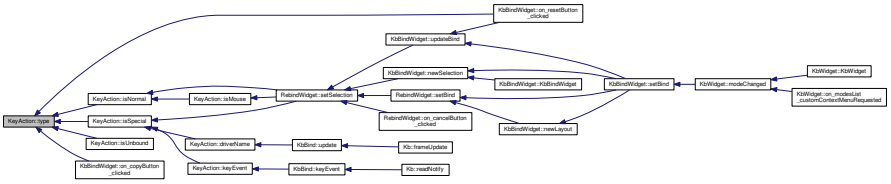
Definition at line 14 of file keyaction.cpp.

References _value, NORMAL, SPECIAL, and UNBOUND.

Referenced by isNormal(), isSpecial(), isUnbound(), KbBindWidget::on_copyButton_clicked(), and KbBindWidget::on_resetButton_clicked().

```
14                                  {
15     if(_value.isEmpty())
16         return UNBOUND;
17     if(_value.at(0) == '$')
18         return SPECIAL;
19     return NORMAL;
20 }
```

Here is the caller graph for this function:



12.42.4.37 QString KeyAction::value ( ) const [inline]

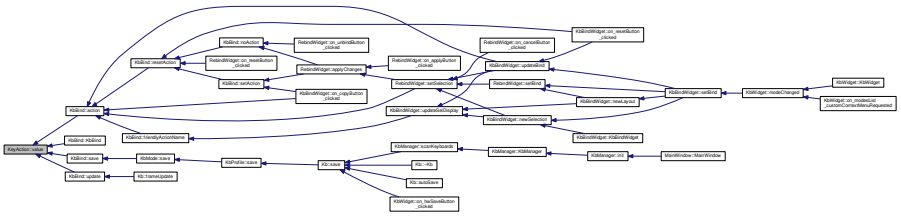
Definition at line 19 of file keyaction.h.

References _value.

Referenced by KbBind::action(), KbBind::KbBind(), KbBind::save(), and KbBind::update().

```
19 { return _value; }
```

Here is the caller graph for this function:



12.42.5 Field Documentation

12.42.5.1 QString KeyAction::_value [private]

Definition at line 189 of file keyaction.h.

Referenced by animInfo(), dpiInfo(), driverName(), friendlyName(), isAnim(), isDPI(), isMacro(), isMedia(), isMouse(), isProgram(), isValidMacro(), keyEvent(), macroContent(), macroDisplay(), macroFullLine(), macroLine(), macroTiming(), operator QString(), programInfo(), specialInfo(), type(), and value().

12.42.5.2 const int KeyAction::DPI_CUSTOM = 6 [static]

Definition at line 128 of file keyaction.h.

Referenced by dpiAction(), dpiInfo(), friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.3 const int KeyAction::DPI_CYCLE_DOWN = -3 [static]

Definition at line 126 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.4 const int KeyAction::DPI_CYCLE_UP = -4 [static]

Definition at line 126 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.5 const int KeyAction::DPI_DOWN = -1 [static]

Definition at line 127 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.6 const int KeyAction::DPI_SNIPER = 0 [static]

Definition at line 128 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.7 const int KeyAction::DPI_UP = -2 [static]

Definition at line 127 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.8 const int KeyAction::LIGHT_DOWN = 1 [static]

Definition at line 131 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.9 const int KeyAction::LIGHT_DOWN_WRAP = 3 [static]

Definition at line 132 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.10 const int KeyAction::LIGHT_UP = 0 [static]

Definition at line 131 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.11 const int KeyAction::LIGHT_UP_WRAP = 2 [static]

Definition at line 132 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.12 const int KeyAction::LOCK_OFF = 2 [static]

Definition at line 135 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.13 const int KeyAction::LOCK_ON = 1 [static]

Definition at line 135 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.14 const int KeyAction::LOCK_TOGGLE = 0 [static]

Definition at line 135 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

12.42.5.15 const int KeyAction::MODE_NEXT = -1 [static]

Definition at line 122 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.16 const int KeyAction::MODE_NEXT_WRAP = -3 [static]

Definition at line 123 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.17 const int KeyAction::MODE_PREV = -2 [static]

Definition at line 122 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.18 const int KeyAction::MODE_PREV_WRAP = -4 [static]

Definition at line 123 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

12.42.5.19 QProcess* KeyAction::preProgram [private]

Definition at line 192 of file keyaction.h.

Referenced by keyEvent(), and ~KeyAction().

12.42.5.20 const int KeyAction::PROGRAM_PR_INDEF = 0x00 [static]

Definition at line 138 of file keyaction.h.

12.42.5.21 const int KeyAction::PROGRAM_PR_KPSTOP = 0x02 [static]

Definition at line 138 of file keyaction.h.

Referenced by keyEvent().

12.42.5.22 const int KeyAction::PROGRAM_PR_KRSTOP = 0x01 [static]

Definition at line 138 of file keyaction.h.

Referenced by keyEvent().

12.42.5.23 const int KeyAction::PROGRAM_PR_MULTI = 0x04 [static]

Definition at line 138 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), keyEvent(), and RebindWidget::setSelection().

12.42.5.24 const int KeyAction::PROGRAM_RE_INDEF = 0x00 [static]

Definition at line 139 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), and RebindWidget::setSelection().

12.42.5.25 const int KeyAction::PROGRAM_RE_KPSTOP = 0x20 [static]

Definition at line 139 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), keyEvent(), and RebindWidget::setSelection().

12.42.5.26 const int KeyAction::PROGRAM_RE_MULTI = 0x40 [static]

Definition at line 139 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), keyEvent(), and RebindWidget::setSelection().

12.42.5.27 QProcess* KeyAction::relProgram [private]

Definition at line 193 of file keyaction.h.

Referenced by keyEvent(), and ~KeyAction().

12.42.5.28 quint64 KeyAction::sniperValue [private]

Definition at line 196 of file keyaction.h.

Referenced by keyEvent().

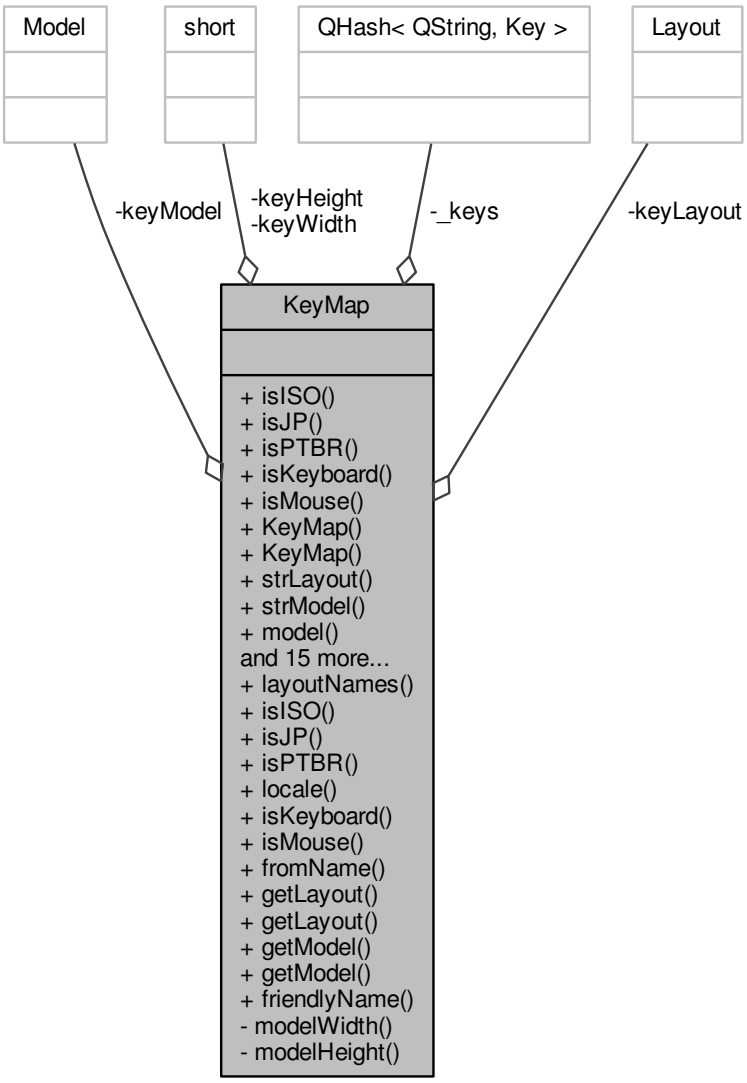
The documentation for this class was generated from the following files:

- src/ckb/keyaction.h
- src/ckb/keyaction.cpp

12.43 KeyMap Class Reference

`#include <src/ckb/keymap.h>`

Collaboration diagram for KeyMap:



Public Types

- enum Model {
  NO_MODEL = -1, K63, K65, K68,
  K70, K95, K95P, STRAFE,
  M65, SABRE, SCIMITAR, HARPOON,
  GLAIVE, _MODEL_MAX }
- enum Layout {

NO_LAYOUT = -1, DK, EU, EU_DVORAK,
GB, GB_DVORAK, US, US_DVORAK,
FR, DE, IT, JP,
NO, PL, PT_BR, MX,
ES, SE, _LAYOUT_MAX }

Public Member Functions

- bool isISO () const
- bool isJP () const
- bool isPTBR () const
- bool isKeyboard () const
- bool isMouse () const
- KeyMap ()
- KeyMap (Model _keyModel, Layout _keyLayout)
- QString strLayout () const
- QString strModel () const
- Model model () const
- Layout layout () const
- QString name () const
- uint count () const
- uint width () const
- uint height () const
- Key key (const QString &name) const
- Key operator[] (const QString &name) const
- bool contains (const QString &name) const
- const QHash< QString, Key > & map () const
- operator const QHash< QString, Key > & () const
- QStringList keys () const
- QList< Key > positions () const
- QString toStorage (const QString &name)
- QString fromStorage (const QString &storage)
- QStringList byPosition () const

Static Public Member Functions

- static QStringList layoutNames ()
- static bool isISO (Layout layout)
- static bool isJP (Layout layout)
- static bool isPTBR (Layout layout)
- static Layout locale ()
- static bool isKeyboard (Model model)
- static bool isMouse (Model model)
- static KeyMap fromName (const QString &name)
- static Layout getLayout (const QString &name)
- static QString getLayout (Layout layout)
- static Model getModel (const QString &name)
- static QString getModel (Model model)
- static QString friendlyName (const QString &key, Layout layout=US)

Static Private Member Functions

- static int modelWidth (Model model)
- static int modelHeight (Model model)

Private Attributes

- QHash< QString, Key > _keys
- short keyWidth
- short keyHeight
- Model keyModel:8
- Layout keyLayout:8

12.43.1 Detailed Description

Definition at line 49 of file keymap.h.

12.43.2 Member Enumeration Documentation

12.43.2.1 enum KeyMap::Layout

Enumerator

NO_LAYOUT
DK
EU
EU_DVORAK
GB
GB_DVORAK
US
US_DVORAK
FR
DE
IT
JP
NO
PL
PT_BR
MX
ES
SE
_LAYOUT_MAX

Definition at line 70 of file keymap.h.

```
70                 {
71         NO_LAYOUT = -1,
72         DK,                     // Danish
73         EU,                     // English (EU)
74         EU_DVORAK,
75         GB,                     // English (UK)
76         GB_DVORAK,
77         US,                     // English (US)
78         US_DVORAK,
79         FR,                     // French
80         DE,                     // German
81         IT,                     // Italian
82         JP,                     // Japanese
83         NO,                     // Norwegian
84         PL,                     // Polish (identical to US)
85         PT_BR,                  // Portuguese (Brazil)
86         MX,                     // Spanish (Mexico/Latin America)
87         ES,                     // Spanish (Spain)
88         SE,                     // Swedish
89         _LAYOUT_MAX
90     };
```

12.43.2.2 enum KeyMap::Model

Enumerator

NO_MODEL

K63

K65

K68

K70

K95

K95P

STRAFE

M65

SABRE

SCIMITAR

HARPOON

GLAIVE

_MODEL_MAX

Definition at line 51 of file keymap.h.

```
51                 {
52         NO_MODEL = -1,
53         // Keyboard models
54         K63,
55         K65,
56         K68,
57         K70,
58         K95,
59         K95P,
60         STRAFE,
61         // Mouse models
62         M65,
63         SABRE,
64         SCIMITAR,
65         HARPOON,
66         GLAIVE,
67         _MODEL_MAX
68     };
```

12.43.3 Constructor & Destructor Documentation

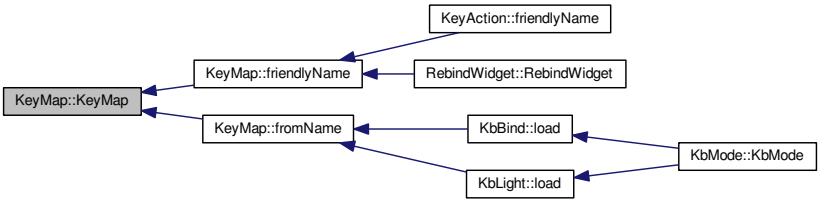
12.43.3.1 KeyMap::KeyMap ( )

Definition at line 897 of file keymap.cpp.

Referenced by friendlyName(), and fromName().

```
897                 :
898       keyWidth(0), keyHeight(0),
899       keyModel(NO_MODEL), keyLayout(NO_LAYOUT)
900 {}
```

Here is the caller graph for this function:



12.43.3.2 KeyMap::KeyMap (Model *_keyModel,* Layout *_keyLayout*)

Definition at line 891 of file keymap.cpp.

```
891                                                 :
892     _keys(getMap(_keyModel, _keyLayout)),
893     keyWidth(modelWidth(_keyModel)), keyHeight(
      modelHeight(_keyModel)),
894     keyModel(_keyModel), keyLayout(_keyLayout)
895 {}
```

12.43.4 Member Function Documentation

12.43.4.1 QStringList KeyMap::byPosition ( ) const

Definition at line 902 of file keymap.cpp.

References Key::x, x, Key::y, and y.

Referenced by RebindWidget::setBind().

```
902                                     {
903     // Use QMaps to order the keys
904     QMap<int, QMap<int, QString> > ordered;
905     QHashIterator<QString, Key> i(*this);
906     while(i.hasNext()){
907         i.next();
908         const Key& key = i.value();
909         ordered[key.y][key.x] = i.key();
910     }
911     // Merge them into a single list
912     QStringList result;
913     QMapIterator<int, QMap<int, QString> > y(ordered);
914     while(y.hasNext()){
915         y.next();
916         QMapIterator<int, QString> x(y.value());
917         while(x.hasNext()){
918             x.next();
919             result << x.value();
920         }
921     }
922     return result;
923 }
```

Here is the caller graph for this function:

12.43.4.2 bool KeyMap::contains (const QString & *name*) const [inline]

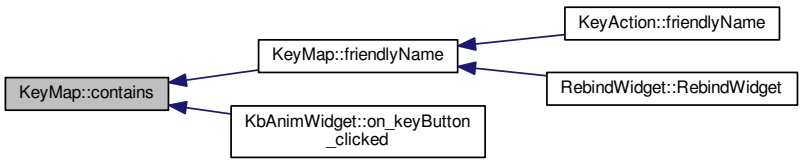
Definition at line 143 of file keymap.h.

References _keys.

Referenced by friendlyName(), and KbAnimWidget::on_keyButton_clicked().

```
143 { return _keys.contains(name); }
```

Here is the caller graph for this function:



12.43.4.3 uint KeyMap::count ( ) const [inline]

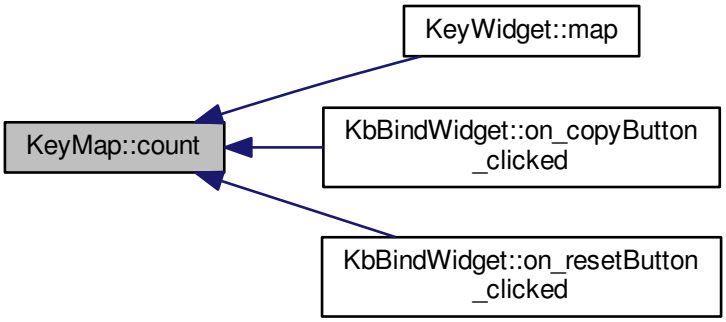
Definition at line 134 of file keymap.h.

References _keys.

Referenced by KeyWidget::map(), KbBindWidget::on_copyButton_clicked(), and KbBindWidget::on_resetButton_-clicked().

```
134 { return _keys.count(); }
```

Here is the caller graph for this function:



12.43.4.4 QString KeyMap::friendlyName (const QString & *key*, Layout *layout* = US) [static]

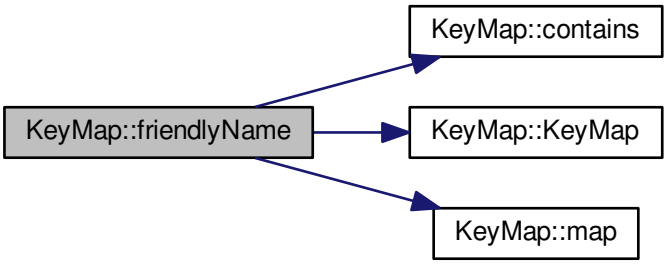
Definition at line 925 of file keymap.cpp.

References contains(), HARPOON, K95, KeyMap(), M65, map(), SCIMITAR, and STRAFE.

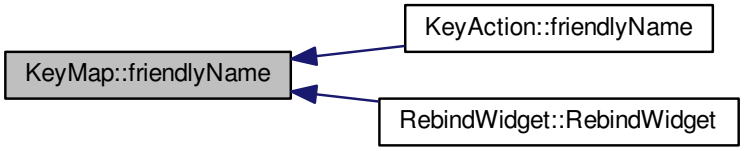
Referenced by KeyAction::friendlyName(), and RebindWidget::RebindWidget().

```
925                                                     {
926     // Try K95 map first
927     // FIXME: This is an odd function and probably should be refactored
928     // it would probably be best to remove the friendly names from the maps and have a completely separate
      name->friendlyName store
929     KeyMap map(K95, layout);
930     if(map.contains(key))
931         return map[key].friendlyName();
932
933     // The only key missing from it should be Fn, which is found on STRAFE
934     map = KeyMap(STRAFE, layout);
935     if(map.contains(key))
936         return map[key].friendlyName();
937
938     // Additionally, there are a handful of keys not present on any physical keyboard, but we need names
      for them
939     if(key == "f13" || key == "f14" || key == "f15" || key == "f16" ||
      key == "f17" || key == "f18" || key == "f19" || key == "f20")
940         return key.toUpper();
941     else if(key == "lightup")
942         return "Screen Brightness Up";
943     else if(key == "lightdn")
944         return "Screen Brightness Down";
945     else if(key == "eject" || key == "power")
946         return key[0].toUpper() + key.mid(1);   // capitalize first letter
947
948     // All other names are found on mice
949     map = KeyMap(SCIMITAR, layout);
950     if(map.contains(key))
951         return map[key].friendlyName();
952     map = KeyMap(M65, layout);
953     if(map.contains(key))
954         return map[key].friendlyName();
955     map = KeyMap(HARPOON, layout);
956     if(map.contains(key))
957         return map[key].friendlyName();
958
959     // Not found at all
960     return "";
961 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.43.4.5 KeyMap KeyMap::fromName (const QString & *name*) [static]

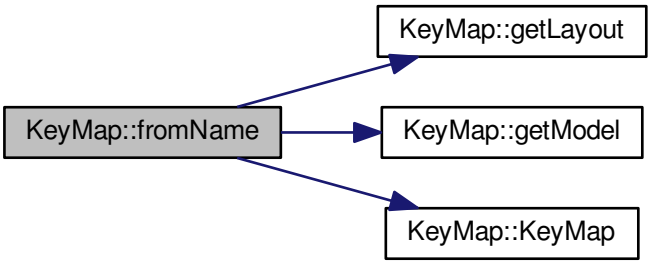
Definition at line 835 of file keymap.cpp.

References getLayout(), getModel(), and KeyMap().

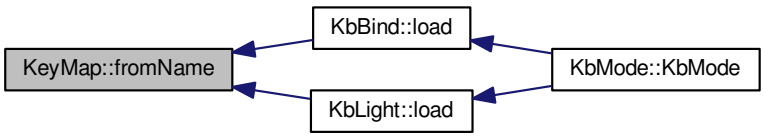
Referenced by KbBind::load(), and KbLight::load().

```
835                                          {
836     QStringList list = name.trimmed().split(" ");
837     if(list.length() != 2)
838         return KeyMap();
839     return KeyMap(getModel(list[0]), getLayout(list[1]));
840 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



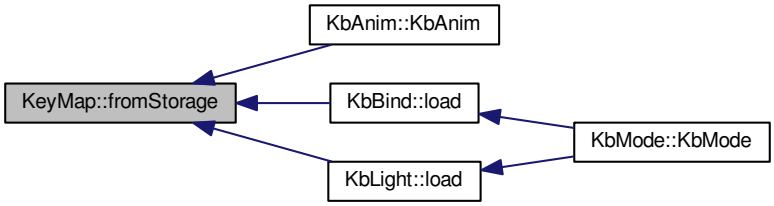
12.43.4.6 QString KeyMap::fromStorage (const QString & *storage*) [inline]

Definition at line 151 of file keymap.h.

Referenced by KbAnim::KbAnim(), KbBind::load(), and KbLight::load().

```
151 { QHashIterator<QString, Key> i(*this); while(i.hasNext()) { i.next(); const char* s = i.value().
      storageName(); if(s == storage) return i.value().name; } return storage; }
```

Here is the caller graph for this function:



12.43.4.7 KeyMap::Layout KeyMap::getLayout (const QString & *name*) [static]

Definition at line 677 of file keymap.cpp.

References DE, DK, ES, EU, EU_DVORAK, FR, GB, GB_DVORAK, IT, JP, MX, NO, NO_LAYOUT, PL, PT_BR, SE, US, and US_DVORAK.

Referenced by fromName(), SettingsWidget::on_layoutBox_activated(), SettingsWidget::SettingsWidget(), and strLayout().

```
677                                                 {
678     QString lower = name.toLower();
679     if(lower == "dk")
680         return DK;
681     if(lower == "eu")
682         return EU;
683     if(lower == "eu_dvorak")
684         return EU_DVORAK;
685     if(lower == "gb_dvorak")
686         return GB_DVORAK;
687     if(lower == "us")
688         return US;
689     if(lower == "us_dvorak")
```

```
690         return US_DVORAK;
691     if(lower == "fr")
692         return FR;
693     if(lower == "de")
694         return DE;
695     if(lower == "it")
696         return IT;
697     if(lower == "no")
698         return NO;
699     if(lower == "jp")
700         return JP;
701     if(lower == "pl")
702         return PL;
703     if(lower == "pt_br")
704         return PT_BR;
705     if(lower == "mx")
706         return MX;
707     if(lower == "es")
708         return ES;
709     if(lower == "se")
710         return SE;
711     if(lower == "gb")
712         return GB;
713     return NO_LAYOUT;
714 }
```

Here is the caller graph for this function:

12.43.4.8 QString KeyMap::getLayout (KeyMap::Layout *layout*) [static]

Definition at line 716 of file keymap.cpp.

References DE, DK, ES, EU, EU_DVORAK, FR, GB, GB_DVORAK, IT, JP, MX, NO, PL, PT_BR, SE, US, and US_DVORAK.

```
716                                             {
717     switch(layout){
718     case DK:
719         return "dk";
720     case EU:
721         return "eu";
722     case EU_DVORAK:
723         return "eu_dvorak";
724     case GB:
725         return "gb";
726     case GB_DVORAK:
727         return "gb_dvorak";
728     case US:
729         return "us";
730     case US_DVORAK:
731         return "us_dvorak";
732     case FR:
733         return "fr";
734     case DE:
735         return "de";
736     case IT:
737         return "it";
738     case JP:
739         return "jp";
740     case NO:
741         return "no";
742     case PL:
743         return "pl";
744     case PT_BR:
745         return "pt_br";
746     case MX:
747         return "mx";
748     case ES:
```

```
749         return "es";
750     case SE:
751         return "se";
752     default:
753         return "";
754     }
755 }
```

12.43.4.9 KeyMap::Model KeyMap::getModel (const QString & *name*) [static]

Definition at line 775 of file keymap.cpp.

References GLAIVE, HARPOON, K63, K65, K68, K70, K95, K95P, M65, NO_MODEL, SABRE, SCIMITAR, and STRAFE.

Referenced by fromName(), Kb::Kb(), and strModel().

```
775                                                 {
776     QString lower = name.toLower();
777     if(lower == "k63")
778         return K63;
779     if(lower == "k65")
780         return K65;
781     if(lower == "k68")
782         return K68;
783     if(lower == "k70")
784         return K70;
785     if(lower == "k95")
786         return K95;
787     if(lower == "k95p")
788         return K95P;
789     if(lower == "strafe")
790         return STRAFE;
791     if(lower == "m65")
792         return M65;
793     if(lower == "sabre")
794         return SABRE;
795     if(lower == "scimitar")
796         return SCIMITAR;
797     if(lower == "harpoon")
798         return HARPOON;
799     if(lower == "glaive")
800         return GLAIVE;
801     return NO_MODEL;
802 }
```

Here is the caller graph for this function:

12.43.4.10 QString KeyMap::getModel (KeyMap::Model *model*) [static]

Definition at line 804 of file keymap.cpp.

References GLAIVE, HARPOON, K63, K65, K68, K70, K95, K95P, M65, SABRE, SCIMITAR, and STRAFE.

```
804                                                 {
805     switch(model){
806     case K63:
807         return "k63";
808     case K65:
809         return "k65";
810     case K68:
811         return "k68";
812     case K70:
813         return "k70";
```

```
814     case K95:
815         return "k95";
816     case K95P:
817         return "k95P";
818     case STRAFE:
819         return "strafe";
820     case M65:
821         return "m65";
822     case SABRE:
823         return "sabre";
824     case SCIMITAR:
825         return "scimitar";
826     case HARPOON:
827         return "harpoon";
828     case GLAIVE:
829         return "glaive";
830     default:
831         return "";
832     }
833 }
```

12.43.4.11 uint KeyMap::height ( ) const [inline]

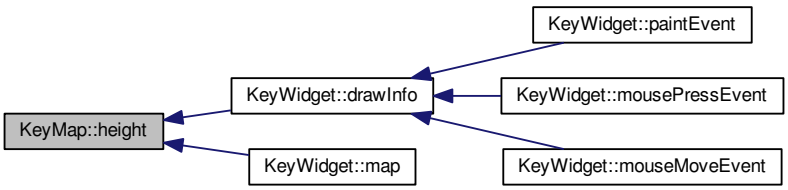
Definition at line 138 of file keymap.h.

References keyHeight.

Referenced by KeyWidget::drawInfo(), and KeyWidget::map().

```
138 { return keyHeight; }
```

Here is the caller graph for this function:



12.43.4.12 static bool KeyMap::isISO (Layout *layout*) [inline], [static]

Definition at line 94 of file keymap.h.

References PL, US, and US_DVORAK.

Referenced by KbBind::isISO(), and RebindWidget::setBind().

```
94 { return layout != US && layout != US_DVORAK && layout !=
     PL; }
```

Here is the caller graph for this function:

12.43.4.13 bool KeyMap::isISO ( ) const [inline]

Definition at line 95 of file keymap.h.

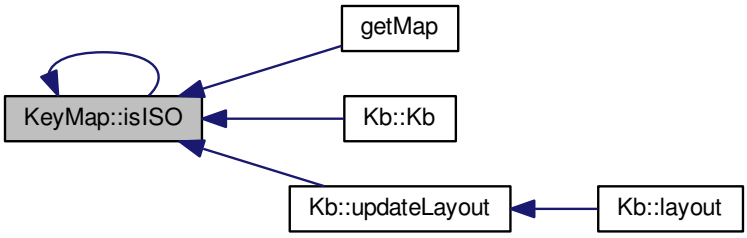
References isISO(), and keyLayout.

Referenced by getMap(), isISO(), Kb::Kb(), and Kb::updateLayout().

```
95 { return isISO(keyLayout); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.43.4.14 static bool KeyMap::isJP (Layout *layout*) [inline], [static]

Definition at line 98 of file keymap.h.

References JP.

```
98 { return layout == JP; }
```

12.43.4.15 bool KeyMap::isJP ( ) const [inline]

Definition at line 99 of file keymap.h.

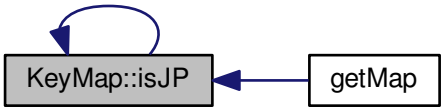
References isJP(), and keyLayout.

Referenced by getMap(), and isJP().

```
99 { return isJP(keyLayout); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.43.4.16 static bool KeyMap::isKeyboard (Model *model*) [inline], [static]

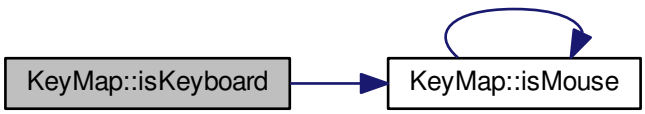
Definition at line 109 of file keymap.h.

References isMouse(), and NO_MODEL.

Referenced by KbBind::isKeyboard().

```
109 { return !isMouse(model) && model != NO_MODEL; }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.43.4.17 bool KeyMap::isKeyboard ( ) const [inline]

Definition at line 110 of file keymap.h.

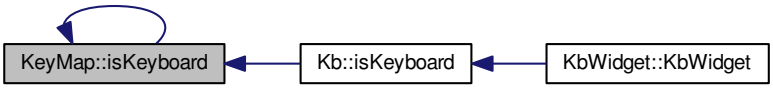
References isKeyboard(), and keyModel.

Referenced by Kb::isKeyboard(), and isKeyboard().

```
110 { return isKeyboard(keyModel); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.43.4.18 static bool KeyMap::isMouse (Model *model*) [inline], [static]

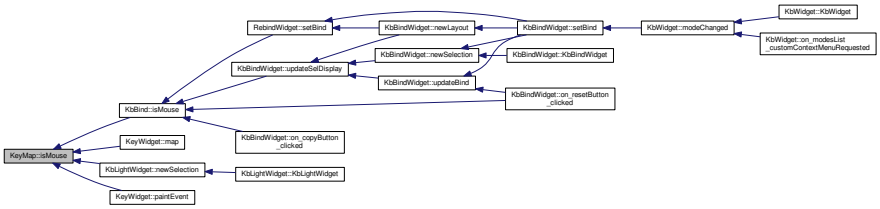
Definition at line 111 of file keymap.h.

References GLAIVE, HARPOON, M65, SABRE, and SCIMITAR.

Referenced by KbBind::isMouse(), KeyWidget::map(), KbLightWidget::newSelection(), and KeyWidget::paintEvent().

```
111 { return model == M65 || model == SABRE || model ==
      SCIMITAR || model == HARPOON || model == GLAIVE; }
```

Here is the caller graph for this function:



12.43.4.19 bool KeyMap::isMouse ( ) const [inline]

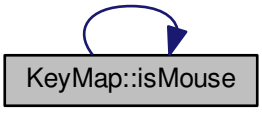
Definition at line 112 of file keymap.h.

References isMouse(), and keyModel.

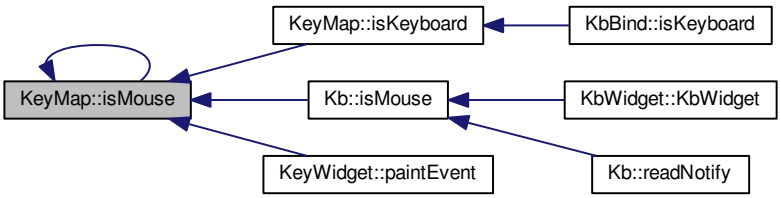
Referenced by isKeyboard(), Kb::isMouse(), isMouse(), and KeyWidget::paintEvent().

```
112 { return isMouse(keyModel); }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.43.4.20 static bool KeyMap::isPTBR (Layout *layout*) [inline], [static]

Definition at line 102 of file keymap.h.

References PT_BR.

```
102 { return layout == PT_BR; }
```

12.43.4.21 bool KeyMap::isPTBR ( ) const [inline]

Definition at line 103 of file keymap.h.

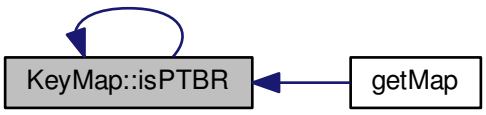
References isPTBR(), and keyLayout.

Referenced by getMap(), and isPTBR().

```
103 { return isPTBR(keyLayout); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.43.4.22 Key KeyMap::key (const QString & *name*) const [inline]

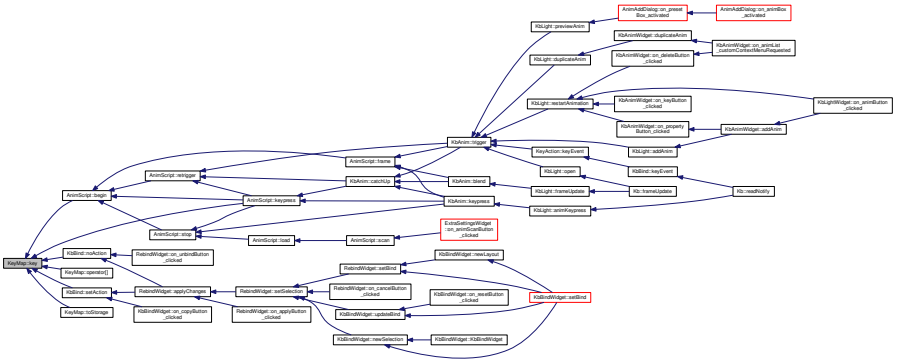
Definition at line 141 of file keymap.h.

References _keys.

Referenced by AnimScript::begin(), AnimScript::keypress(), KbBind::noAction(), operator[](), KbBind::setAction(), and toStorage().

```
141 { Key empty = {0,0,0,0,0,0,0,0,0}; return _keys.value(name, empty); }
```

Here is the caller graph for this function:



12.43.4.23 QStringList KeyMap::keys ( ) const [inline]

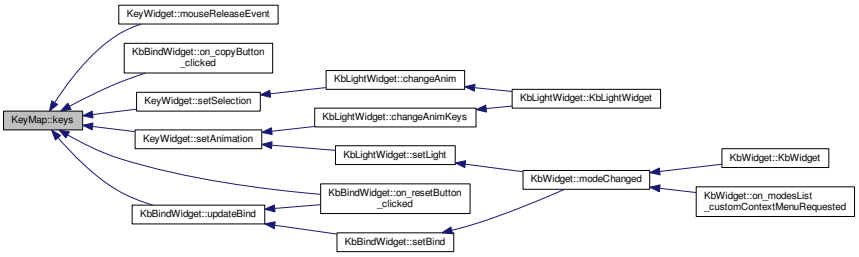
Definition at line 147 of file keymap.h.

References _keys.

Referenced by KeyWidget::mouseReleaseEvent(), KbBindWidget::on_copyButton_clicked(), KbBindWidget::on_-resetButton_clicked(), KeyWidget::setAnimation(), KeyWidget::setSelection(), and KbBindWidget::updateBind().

```
147 { return _keys.keys(); }
```

Here is the caller graph for this function:



12.43.4.24 Layout KeyMap::layout ( ) const [inline]

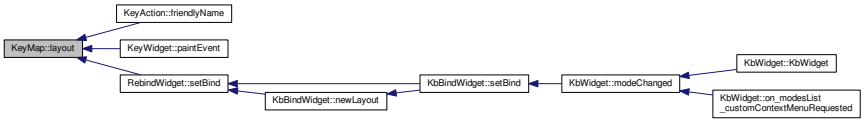
Definition at line 130 of file keymap.h.

References keyLayout.

Referenced by KeyAction::friendlyName(), KeyWidget::paintEvent(), and RebindWidget::setBind().

```
130 { return keyLayout; }
```

Here is the caller graph for this function:



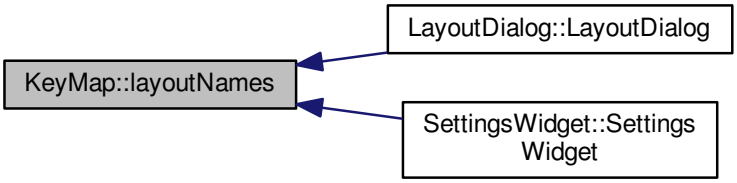
12.43.4.25 QStringList KeyMap::layoutNames ( ) [static]

Definition at line 757 of file keymap.cpp.

Referenced by LayoutDialog::LayoutDialog(), and SettingsWidget::SettingsWidget().

```
757                                {
758     return QStringList()
759             << "Danish"
760             << "English (ISO/European)" << "English (ISO/European, Dvorak)"
761             << "English (United Kingdom)" << "English (United Kingdom, Dvorak)"
762             << "English (United States)" << "English (United States, Dvorak)"
763             << "French"
764             << "German"
765             << "Italian"
766             << "Japanese"
767             << "Norwegian"
768             << "Polish"
769             << "Portuguese (Brazil)"
770             << "Spanish (Latin America)"
771             << "Spanish (Spain)"
772             << "Swedish";
773 }
```

Here is the caller graph for this function:



12.43.4.26 KeyMap::Layout KeyMap::locale ( ) [static]

Definition at line 641 of file keymap.cpp.

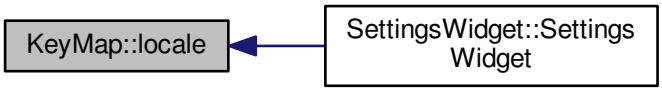
References DE, DK, ES, FR, GB, IT, JP, MX, NO, PL, PT_BR, SE, and US.

Referenced by SettingsWidget::SettingsWidget().

```
641                           {
642     setlocale(LC_ALL, "");
643     QString loc = setlocale(LC_CTYPE, 0);
644     loc = loc.toLower().replace('_', '-');
645     if(loc.startsWith("dk-"))
646         return KeyMap::DK;
```

```
647     else if(loc.startsWith("fr-"))
648         return KeyMap::FR;
649     else if(loc.startsWith("de-"))
650         return KeyMap::DE;
651     else if(loc.startsWith("it-"))
652         return KeyMap::IT;
653     else if(loc.startsWith("ja-"))
654         return KeyMap::JP;
655     else if(loc.startsWith("pl-"))
656         return KeyMap::PL;
657     else if(loc.startsWith("pt-br"))
658         return KeyMap::PT_BR;
659     else if(loc.startsWith("no-"))
660         return KeyMap::NO;
661     else if(loc.startsWith("es-es"))
662         // Spain uses the ES layout
663         return KeyMap::ES;
664     else if(loc.startsWith("es-"))
665         // Other Spanish locales use MX
666         return KeyMap::MX;
667     else if(loc.startsWith("se-"))
668         return KeyMap::SE;
669     else if(loc.startsWith("en-us") || loc.startsWith("en-au") || loc.startsWith("en-ca") || loc.startsWith
    ("en-hk") || loc.startsWith("en-in") || loc.startsWith("en-nz") || loc.startsWith("en-ph") || loc.startsWith
    ("en-sg") || loc.startsWith("en-za"))
670         // Most English locales use US
671         return KeyMap::US;
672     else
673         // Default to GB
674         return KeyMap::GB;
675 }
```

Here is the caller graph for this function:



12.43.4.27 const QHash<QString, Key>& KeyMap::map ( ) const [inline]

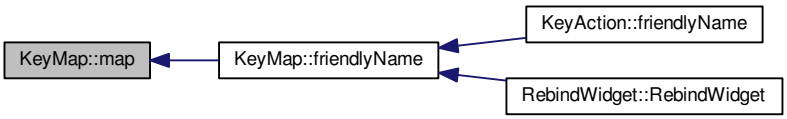
Definition at line 145 of file keymap.h.

References _keys.

Referenced by friendlyName().

```
145 { return _keys; }
```

Here is the caller graph for this function:



12.43.4.28 Model KeyMap::model ( ) const [inline]

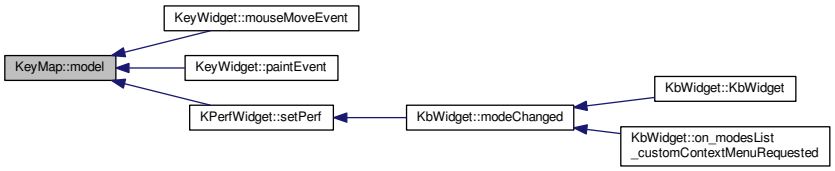
Definition at line 129 of file keymap.h.

References keyModel.

Referenced by KeyWidget::mouseMoveEvent(), KeyWidget::paintEvent(), and KPerfWidget::setPerf().

```
129 { return keyModel; }
```

Here is the caller graph for this function:



12.43.4.29 int KeyMap::modelHeight (Model *model*) [static], [private]

Definition at line 869 of file keymap.cpp.

References GLAIVE, HARPOON, K63, K65, K68, K70, K95, K95_HEIGHT, K95P, K95P_HEIGHT, M65, M65_HEIGHT, SABRE, SCIMITAR, and STRAFE.

```
869                                 {
870     switch(model){
871     case K63:
872     case K65:
873     case K68:
874     case K70:
875     case K95:
876     case STRAFE:
877         return K95_HEIGHT;
878     case K95P:
879         return K95P_HEIGHT;
880     case M65:
881     case SABRE:
882     case SCIMITAR:
883     case HARPOON:
884     case GLAIVE:
885         return M65_HEIGHT;
886     default:
887         return 0;
888     }
889 }
```

12.43.4.30 int KeyMap::modelWidth (Model *model*) [static], [private]

Definition at line 842 of file keymap.cpp.

References GLAIVE, HARPOON, K63, K63_WIDTH, K65, K65_WIDTH, K68, K68_WIDTH, K70, K70_WIDTH, K95, K95_WIDTH, K95P, K95P_WIDTH, KSTRAFE_WIDTH, M65, M65_WIDTH, SABRE, SCIMITAR, and STRAFE.

```
842                                 {
843     switch(model){
844     case K63:
845         return K63_WIDTH;
846     case K65:
847         return K65_WIDTH;
848     case K68:
849         return K68_WIDTH;
```

```
850     case K70:
851         return K70_WIDTH;
852     case K95:
853          return K95_WIDTH;
854     case K95P:
855         return K95P_WIDTH;
856     case STRAFE:
857         return KSTRAFE_WIDTH;
858     case M65:
859     case SABRE:
860     case SCIMITAR:
861     case HARPOON:
862     case GLAIVE:
863         return M65_WIDTH;
864     default:
865         return 0;
866     }
867 }
```

12.43.4.31 QString KeyMap::name ( ) const [inline]

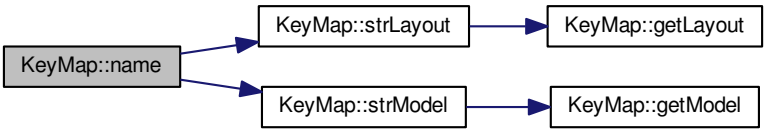
Definition at line 131 of file keymap.h.

References strLayout(), and strModel().

Referenced by KbBind::save(), KbLight::save(), and toStorage().

```
131 { return (strModel() + " " + strLayout()).toUpper(); }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.43.4.32 KeyMap::operator const QHash< QString, Key > & ( ) const [inline]

Definition at line 146 of file keymap.h.

References _keys.

```
146 { return _keys; }
```

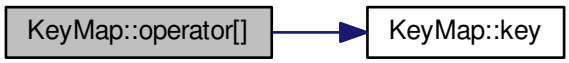
12.43.4.33 Key KeyMap::operator[] (const QString & *name*) const [inline]

Definition at line 142 of file keymap.h.

References key().

```
142 { return key(name); }
```

Here is the call graph for this function:



12.43.4.34 QList<Key> KeyMap::positions ( ) const [inline]

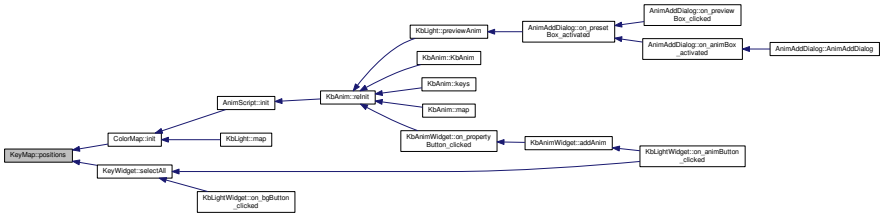
Definition at line 148 of file keymap.h.

References _keys.

Referenced by ColorMap::init(), and KeyWidget::selectAll().

```
148 { return _keys.values(); }
```

Here is the caller graph for this function:



12.43.4.35 QString KeyMap::strLayout ( ) const [inline]

Definition at line 122 of file keymap.h.

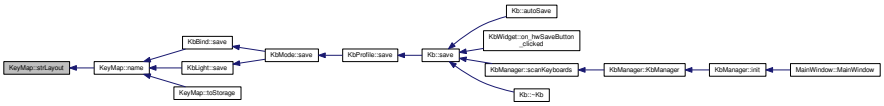
References getLayout(), and keyLayout.

Referenced by name().

```
122 { return getLayout(keyLayout); }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.43.4.36 QString KeyMap::strModel ( ) const [inline]

Definition at line 126 of file keymap.h.

References getModel(), and keyModel.

Referenced by name().

```
126 { return getModel(keyModel); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

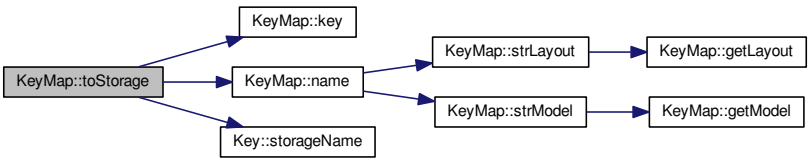
12.43.4.37 QString KeyMap::toStorage (const QString & *name*) [inline]

Definition at line 150 of file keymap.h.

References key(), name(), and Key::storageName().

```
150 { const char* storage = key(name).storageName(); if(!storage) return
      name; return storage; }
```

Here is the call graph for this function:



12.43.4.38 uint KeyMap::width ( ) const [inline]

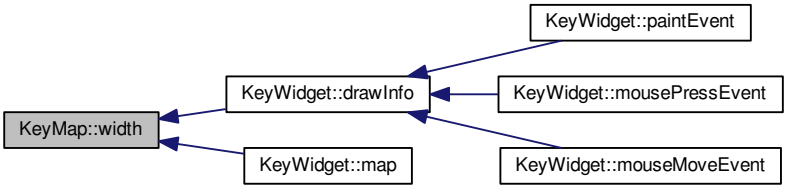
Definition at line 136 of file keymap.h.

References keyWidth.

Referenced by KeyWidget::drawInfo(), and KeyWidget::map().

```
136 { return keyWidth; }
```

Here is the caller graph for this function:



12.43.5 Field Documentation

12.43.5.1 QHash<QString, Key> KeyMap::_keys [private]

Definition at line 163 of file keymap.h.

Referenced by contains(), count(), key(), keys(), map(), operator const QHash< QString, Key > &(), and positions().

12.43.5.2 short KeyMap::keyHeight [private]

Definition at line 164 of file keymap.h.

Referenced by height().

12.43.5.3 Layout KeyMap::keyLayout [private]

Definition at line 166 of file keymap.h.

Referenced by isISO(), isJP(), isPTBR(), layout(), and strLayout().

12.43.5.4 Model KeyMap::keyModel [private]

Definition at line 165 of file keymap.h.

Referenced by isKeyboard(), isMouse(), model(), and strModel().

12.43.5.5 short KeyMap::keyWidth [private]

Definition at line 164 of file keymap.h.

Referenced by width().

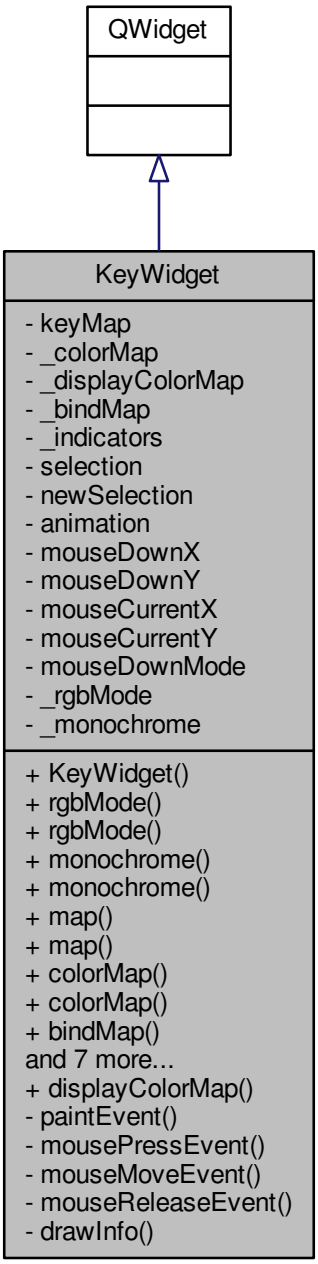
The documentation for this class was generated from the following files:

- src/ckb/keymap.h
- src/ckb/keymap.cpp

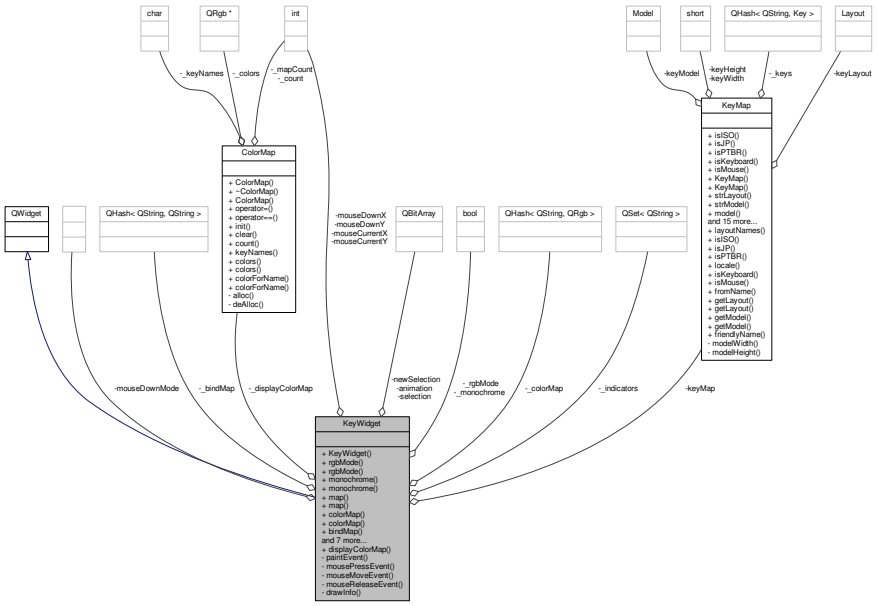
12.44 KeyWidget Class Reference

`#include <src/ckb/keywidget.h>`

Inheritance diagram for KeyWidget:



Collaboration diagram for KeyWidget:



Public Types

- typedef QHash< QString, QString > BindMap

Public Slots

- void displayColorMap (const ColorMap &newDisplayMap, const QSet< QString > &indicators=QSet< QString >())

Signals

- void selectionChanged (QStringList selected)
- void sidelightToggled ()

Public Member Functions

- KeyWidget (QWidget *parent=0, bool rgbMode=true)
- bool rgbMode ()
- void rgbMode (bool newRgbMode)
- bool monochrome ()
- void monochrome (bool newMono)
- const KeyMap & map () const
- void map (const KeyMap &newMap)
- const QColorMap & colorMap () const
- void colorMap (const QColorMap &newColorMap)
- const BindMap & bindMap () const
- void bindMap (const BindMap &newBindMap)
- void setSelection (const QStringList &keys)
- void selectAll ()

- void clearSelection ()
- void setAnimation (const QStringList &keys)
- void setAnimationToSelection ()
- void clearAnimation ()

Private Types

- enum {
  NONE, SET, ADD, SUBTRACT,
  TOGGLE }

Private Member Functions

- void paintEvent (QPaintEvent ∗)
- void mousePressEvent (QMouseEvent ∗event)
- void mouseMoveEvent (QMouseEvent ∗event)
- void mouseReleaseEvent (QMouseEvent ∗event)
- void drawInfo (float &scale, float &offsetX, float &offsetY, int ratio=1)

Private Attributes

- KeyMap keyMap
- QColorMap _colorMap
- ColorMap _displayColorMap
- BindMap _bindMap
- QSet< QString > _indicators
- QBitArray selection
- QBitArray newSelection
- QBitArray animation
- int mouseDownX
- int mouseDownY
- int mouseCurrentX
- int mouseCurrentY
- enum KeyWidget:: { ... } mouseDownMode
- bool _rgbMode
- bool _monochrome

12.44.1 Detailed Description

Definition at line 11 of file keywidget.h.

12.44.2 Member Typedef Documentation

12.44.2.1 typedef QHash<QString, QString> KeyWidget::BindMap

Definition at line 30 of file keywidget.h.

12.44.3 Member Enumeration Documentation

12.44.3.1 anonymous enum [private]

Enumerator

NONE

SET

ADD

SUBTRACT

TOGGLE

Definition at line 66 of file keywidget.h.

```
66          {
67         NONE,
68         SET,
69         ADD,
70         SUBTRACT,
71         TOGGLE,
72     } mouseDownMode;
```

12.44.4 Constructor & Destructor Documentation

12.44.4.1 KeyWidget::KeyWidget (QWidget * *parent* = 0, bool *rgbMode* = true) [explicit]

Definition at line 17 of file keywidget.cpp.

```
17                                                  :
18     QWidget(parent), mouseDownX(-1), mouseDownY(-1),
       mouseCurrentX(-1), mouseCurrentY(-1), mouseDownMode(
       NONE), _rgbMode(rgbMode), _monochrome(false)
19 {
20     setMouseTracking(true);
21     setAutoFillBackground(false);
22 }
```

12.44.5 Member Function Documentation

12.44.5.1 const BindMap& KeyWidget::bindMap ( ) const [inline]

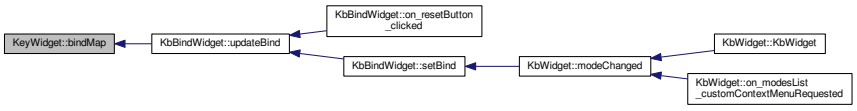
Definition at line 31 of file keywidget.h.

References _bindMap.

Referenced by KbBindWidget::updateBind().

```
31 { return _bindMap; }
```

Here is the caller graph for this function:



12.44.5.2 void KeyWidget::bindMap (const BindMap & *newBindMap*)

Definition at line 65 of file keywidget.cpp.

References _bindMap.

```
65                                                  {
66     _bindMap = newBindMap;
67     update();
68 }
```

12.44.5.3 void KeyWidget::clearAnimation ( )

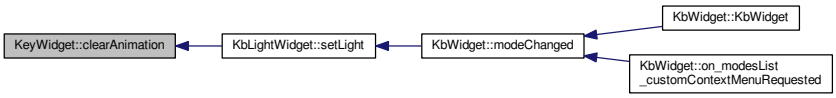
Definition at line 644 of file keywidget.cpp.

References animation.

Referenced by KbLightWidget::setLight().

```
644                                 {
645     animation.fill(false);
646     update();
647 }
```

Here is the caller graph for this function:



12.44.5.4 void KeyWidget::clearSelection ( )

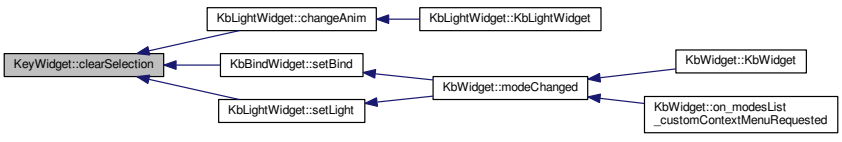
Definition at line 617 of file keywidget.cpp.

References mouseDownMode, newSelection, NONE, selection, and selectionChanged().

Referenced by KbLightWidget::changeAnim(), KbBindWidget::setBind(), and KbLightWidget::setLight().

```
617                                 {
618     selection.fill(false);
619     newSelection.fill(false);
620     mouseDownMode = NONE;
621     update();
622     emit selectionChanged(QStringList());
623 }
```

Here is the caller graph for this function:



12.44.5.5 const QColorMap& KeyWidget::colorMap ( ) const [inline]

Definition at line 27 of file keywidget.h.

References _colorMap.

Referenced by KbLightWidget::changeColor(), KbLightWidget::toggleSidelight(), and KbLightWidget::updateLight().

```
27 { return _colorMap; }
```

Here is the caller graph for this function:

12.44.5.6 void KeyWidget::colorMap (const QColorMap & *newColorMap*)

Definition at line 52 of file keywidget.cpp.

References _colorMap.

```
52                                                  {
53     _colorMap = newColorMap;
54     update();
55 }
```

12.44.5.7 void KeyWidget::displayColorMap (const ColorMap & *newDisplayMap*, const QSet< QString > & *indicators* = QSet<QString>()) [slot]

Definition at line 57 of file keywidget.cpp.

References _displayColorMap, and _indicators.

Referenced by KbLightWidget::on_showAnimBox_clicked().

```
57                                                                                  {
58     if(!isVisible())
59         return;
60     _displayColorMap = newDisplayMap;
61     _indicators = indicators;
62     update();
63 }
```

Here is the caller graph for this function:

12.44.5.8 void KeyWidget::drawInfo (float & *scale*, float & *offsetX*, float & *offsetY*, int *ratio* = 1) [private]

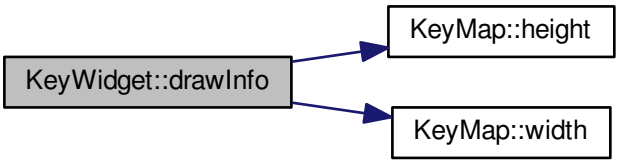
Definition at line 43 of file keywidget.cpp.

References KeyMap::height(), KEY_SIZE, keyMap, width, and KeyMap::width().

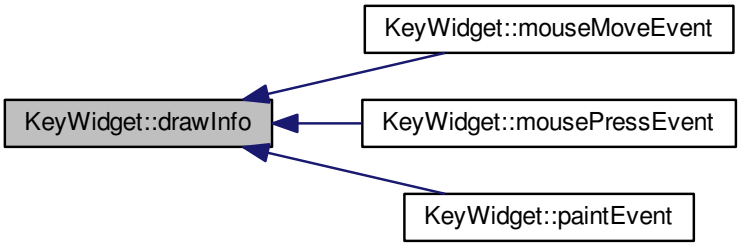
Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

```
43                                                                                  {
44     int w = width() * ratio, h = height() * ratio;
45     float xScale = (float)w / (keyMap.width() + KEY_SIZE);
46     float yScale = (float)h / (keyMap.height() + KEY_SIZE);
47     scale = fmin(xScale, yScale);
48     offsetX = (w / scale - keyMap.width()) / 2.f;
49     offsetY = (h / scale - keyMap.height()) / 2.f;
50 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.44.5.9 const KeyMap& KeyWidget::map ( ) const [inline]

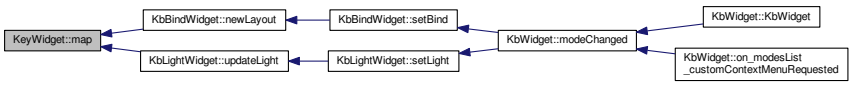
Definition at line 24 of file keywidget.h.

References keyMap.

Referenced by KbBindWidget::newLayout(), and KbLightWidget::updateLight().

```
24 { return keyMap; }
```

Here is the caller graph for this function:



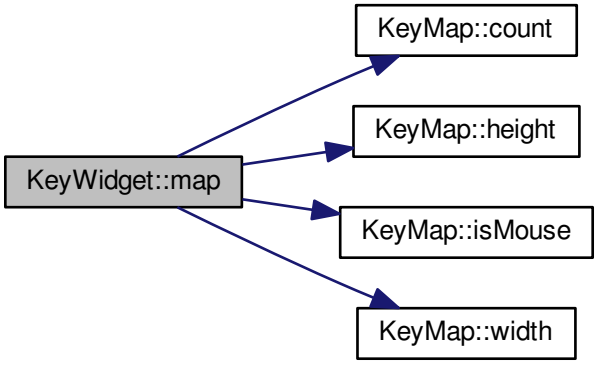
12.44.5.10 void KeyWidget::map (const KeyMap & *newMap*)

Definition at line 24 of file keywidget.cpp.

References animation, KeyMap::count(), KeyMap::height(), KeyMap::isMouse(), KEY_SIZE, keyMap, newSelection, selection, width, and KeyMap::width().

```
24                                      {
25     keyMap = newMap;
26     selection = QBitArray(keyMap.count());
27     newSelection = QBitArray(keyMap.count());
28     animation = QBitArray(keyMap.count());
29     int width, height;
30     if(keyMap.isMouse()){
31         width = (keyMap.width() + KEY_SIZE) * 2.6;
32         height = (keyMap.height() + KEY_SIZE) * 2.6;
33     } else {
34         width = (keyMap.width() + KEY_SIZE) * 2.3;
35         height = (keyMap.height() + KEY_SIZE) * 2.3;
36     }
37     if(width < 500)
38         width = 500;
39     setFixedSize(width, height);
40     update();
41 }
```

Here is the call graph for this function:



12.44.5.11 bool KeyWidget::monochrome ( ) [inline]

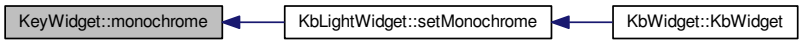
Definition at line 20 of file keywidget.h.

References _monochrome.

Referenced by KbLightWidget::setMonochrome().

```
20 { return _monochrome; }
```

Here is the caller graph for this function:



12.44.5.12 void KeyWidget::monochrome (bool *newMono*) [inline]

Definition at line 21 of file keywidget.h.

References _monochrome.

```
21 { _monochrome = newMono; update(); }
```

12.44.5.13 void KeyWidget::mouseMoveEvent (QMouseEvent * *event*) [private]

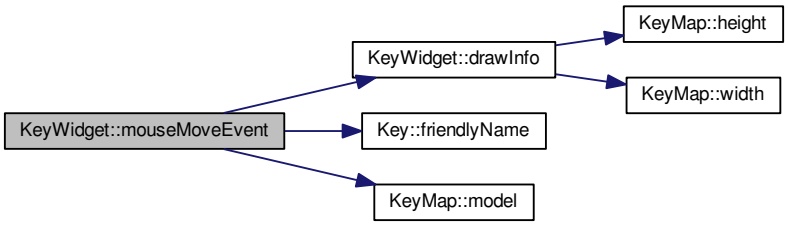
Definition at line 487 of file keywidget.cpp.

References _indicators, _rgbMode, drawInfo(), Key::friendlyName(), Key::hasLed, Key::hasScan, Key::height, keyMap, KeyMap::model(), mouseCurrentX, mouseCurrentY, mouseDownMode, mouseDownX, mouseDownY, Key::name, newSelection, NONE, KeyMap::STRAFE, Key::width, Key::x, and Key::y.

```
487                                                 {
488     event->accept();
489     QString tooltip;
490
491     // Find selection rectangle
492     mouseCurrentX = event->x();
493     mouseCurrentY = event->y();
494     float scale, offX, offY;
495     drawInfo(scale, offX, offY);
496     float mx = mouseCurrentX / scale - offX, my = mouseCurrentY / scale - offY;
497     float mx1, mx2, my1, my2;
498     if(mouseCurrentX >= mouseDownX){
499         mx1 = mouseDownX / scale - offX;
500         mx2 = mouseCurrentX / scale - offX;
501     } else {
502         mx1 = mouseCurrentX / scale - offX;
503         mx2 = mouseDownX / scale - offX;
504     }
505     if(mouseCurrentY >= mouseDownY){
506         my1 = mouseDownY / scale - offY;
507         my2 = mouseCurrentY / scale - offY;
508     } else {
509         my1 = mouseCurrentY / scale - offY;
510         my2 = mouseDownY / scale - offY;
511     }
512     // Clear new selection
513     if(mouseDownMode != NONE)
514         newSelection.fill(false);
515     // See if the event hit any keys
516     QHashIterator<QString, Key> k(keyMap);
517     uint i = -1;
518     while(k.hasNext()){
519         k.next();
520         i++;
521         const Key& key = k.value();
522         if((_rgbMode && !key.hasLed)
523                 || (!_rgbMode && !key.hasScan))
524             continue;
525         // Update tooltip with the mouse hover (if any), even if it's not selectable
526         if(fabs(key.x - mx) <= key.width / 2.f - 1.f && fabs(key.y - my) <= key.
```

```
    height / 2.f - 1.f
527                 && tooltip.isEmpty())
528             tooltip = key.friendlyName(false);
529         // on STRAFE Sidelights and indicators can't be assigned color the way other keys are colored
530         if(keyMap.model() == KeyMap::STRAFE && (!strcmp(key.
    name, "lsidel") || !strcmp(key.name, "rsidel") || _indicators.contains(key.
    name))) // FIX: _indicators check fails whenever _indicators is empty because "show animated" is
     unchecked
531             continue;
532         float kx1 = key.x - key.width / 2.f + 1.f;
533         float ky1 = key.y - key.height / 2.f + 1.f;
534         float kx2 = kx1 + key.width - 2.f;
535         float ky2 = ky1 + key.height - 2.f;
536         // If they overlap, add the key to the selection
537         if(!(mx1 >= kx2 || kx1 >= mx2)
538                 && !(my1 >= ky2 || ky1 >= my2)
539                 && mouseDownMode != NONE)
540             newSelection.setBit(i);
541     }
542
543     if(mouseDownMode != NONE)
544         update();
545     setToolTip(tooltip);
546 }
```

Here is the call graph for this function:



12.44.5.14 void KeyWidget::mousePressEvent (QMouseEvent * *event*) [private]

Definition at line 455 of file keywidget.cpp.

References _rgbMode, ADD, drawInfo(), Key::hasLed, Key::hasScan, Key::height, keyMap, mouseCurrentX, mouseCurrentY, mouseDownMode, mouseDownX, mouseDownY, Key::name, newSelection, SET, sidelightToggled(), SUBTRACT, TOGGLE, Key::width, Key::x, and Key::y.

```
455                                                 {
456     event->accept();
457     mouseDownMode = (event->modifiers() & Qt::AltModifier) ?
    SUBTRACT : (event->modifiers() & Qt::ShiftModifier) ? ADD : (event->modifiers() &
    Qt::ControlModifier) ? TOGGLE : SET;
458     mouseDownX = mouseCurrentX = event->x();
459     mouseDownY = mouseCurrentY = event->y();
460     // See if the event hit a key
461     float scale, offX, offY;
462     drawInfo(scale, offX, offY);
463     float mx = mouseCurrentX / scale - offX, my = mouseCurrentY / scale - offY;
464     QHashIterator<QString, Key> k(keyMap);
465     uint i = -1;
466     while(k.hasNext()){
467         k.next();
468         i++;
469         const Key& key = k.value();
470         if((_rgbMode && !key.hasLed)
471                 || (!_rgbMode && !key.hasScan))
472             continue;
473         if(fabs(key.x - mx) <= key.width / 2.f - 1.f && fabs(key.y - my) <= key.
    height / 2.f - 1.f){
474             // Sidelights can't have a color, but they can be toggled
475             if(!strcmp(key.name, "lsidel") || !strcmp(key.name, "rsidel")){
```

```
476                 emit sidelightToggled(); // get the kblightwidget to record it
477                 update();
478                 break;
479             }
480             newSelection.setBit(i);
481             update();
482             break;
483         }
484     }
485 }
```

Here is the call graph for this function:

12.44.5.15 void KeyWidget::mouseReleaseEvent (QMouseEvent * *event*) [private]

Definition at line 548 of file keywidget.cpp.

References ADD, keyMap, KeyMap::keys(), mouseDownMode, newSelection, NONE, selection, selectionChanged(), SET, SUBTRACT, and TOGGLE.

```
548                                                 {
549     event->accept();
550     if(mouseDownMode == NONE)
551         return;
552     // Apply the new selection
553     switch(mouseDownMode){
554     case SET:
555         selection = newSelection;
556         break;
557     case ADD:
558         selection |= newSelection;
559         break;
560     case SUBTRACT:
561         selection &= ~newSelection;
562         break;
563     case TOGGLE:
564         selection ^= newSelection;
565         break;
566     default:;
567     }
568     // Clear mousedown state.
569     newSelection.fill(false);
570     mouseDownMode = NONE;
571     // Emit signal with the names of the keys
572     QStringList selectedNames;
573     uint i = 0;
574     foreach(const QString& key, keyMap.keys()){
575         if(selection.testBit(i++))
576             selectedNames << key;
577     }
578     emit selectionChanged(selectedNames);
579     update();
580 }
```

Here is the call graph for this function:

12.44.5.16 void KeyWidget::paintEvent (QPaintEvent *) [private]

Definition at line 70 of file keywidget.cpp.

References _bindMap, _colorMap, _displayColorMap, _indicators, _monochrome, _rgbMode, ADD, animation, ColorMap::colorForName(), KbBind::defaultAction(), drawInfo(), KeyMap::EU, KeyMap::EU_DVORAK, Key::friendlyName(), KeyMap::GLAIVE, glaiveOverlay, KbBind::globalRemap(), KeyMap::HARPOON, harpOverlay, Key::hasLed, Key::hasScan, Key::height, KeyMap::isMouse(), KeyMap::K65, KeyMap::K95P, keyMap, KeyMap::layout(), KeyMap::M65, m65Overlay, KeyMap::model(), monoRgb(), mouseCurrentX, mouseCurrentY, mouseDownMode, mouseDownX, mouseDownY, Key::name, newSelection, NONE, sabOverlay, KeyMap::SABRE, KeyMap::SCIMITAR, scimOverlay, selection, SET, KeyMap::STRAFE, SUBTRACT, TOGGLE, Key::width, width, Key::x, x, Key::y, and y.

```
70                                      {
71     const QColor bgColor(68, 64, 64);
72     const QColor keyColor(112, 110, 110);
73     const QColor sniperColor(130, 90, 90);
74     const QColor thumbColor(34, 32, 32);
75     const QColor transparentColor(0, 0, 0, 0);
76     const QColor highlightColor(136, 176, 240);
77     const QColor highlightAnimColor(136, 200, 240);
78     const QColor animColor(112, 200, 110);
79
80     // Determine which keys to highlight
81     QBitArray highlight;
82     switch(mouseDownMode){
83     case SET:
84         highlight = newSelection;
85         break;
86     case ADD:
87         highlight = selection | newSelection;
88         break;
89     case SUBTRACT:
90         highlight = selection & ~newSelection;
91         break;
92     case TOGGLE:
93         highlight = selection ^ newSelection;
94         break;
95     default:
96         highlight = selection;
97     }
98
99     QPainter painter(this);
100 #if QT_VERSION >= QT_VERSION_CHECK(5, 3, 0)
101     int ratio = painter.device()->devicePixelRatio();
102 #else
103     int ratio = 1;
104 #endif
105     int wWidth = width(), wHeight = height();
106     KeyMap::Model model = keyMap.model();
107     KeyMap::Layout layout = keyMap.layout();
108     float scale, offX, offY;
109     drawInfo(scale, offX, offY, ratio);
110     // Draw background
111     painter.setPen(Qt::NoPen);
112     painter.setRenderHint(QPainter::Antialiasing, true);
113
114     if(keyMap.isMouse()){
115         // Draw mouse overlays
116         const QImage* overlay = 0;
117         float xpos = 0.f, ypos = 0.f;
```

```
118         if(model == KeyMap::M65){
119             if(!m65Overlay)
120                 m65Overlay = new QImage(":/img/overlay_m65.png");
121             overlay = m65Overlay;
122             xpos = 2.f;
123             ypos = -2.f;
124         } else if(model == KeyMap::SABRE){
125             if(!sabOverlay)
126                 sabOverlay = new QImage(":/img/overlay_sabre.png");
127             overlay = sabOverlay;
128             xpos = 1.f;
129             ypos = -2.f;
130         } else if(model == KeyMap::SCIMITAR){
131             if(!scimOverlay)
132                 scimOverlay = new QImage(":/img/overlay_scimitar.png");
133             overlay = scimOverlay;
134             xpos = 3.5f;
135             ypos = -2.f;
136         } else if(model == KeyMap::HARPOON){
137             if(!harpOverlay)
138                 harpOverlay = new QImage(":/img/overlay_harpoon.png");
139             overlay = harpOverlay;
140             xpos = 3.5f;
141             ypos = -2.f;
142         } else if(model == KeyMap::GLAIVE){
143             if(!glaiveOverlay)
144                 glaiveOverlay = new QImage(":/img/overlay_glaive.png");
145             overlay = glaiveOverlay;
146             xpos = 3.5f;
147             ypos = -2.f;
148         }
149         if(overlay){
150             painter.setBrush(palette().brush(QPalette::Window));
151             painter.drawRect(0, 0, width(), height());
152             float oXScale = scale / 9.f, oYScale = scale / 9.f;             // The overlay has a resolution
     of 9px per keymap unit
153             float x = (xpos + offX) * scale, y = (ypos + offY) * scale;
154             int w = overlay->width() * oXScale, h = overlay->height() * oYScale;
155             // We need to transform the image with QImage::scaled() because painter.drawImage() will
     butcher it, even with smoothing enabled
156             // However, the width/height need to be rounded to integers
157             int iW = round(w), iH = round(h);
158             painter.drawImage(QRectF((x - (iW - w) / 2.f) / ratio, (y - (iH - h) / 2.f) / ratio, iW /
    ratio, iH / ratio), overlay->scaled(iW, iH, Qt::IgnoreAspectRatio, Qt::SmoothTransformation));
159         }
160     } else {
161         // Otherwise, draw a solid background
162         painter.setBrush(QBrush(bgColor));
163         painter.drawRect(0, 0, width(), height());
164     }
165
166     // Draw mouse highlight (if any)
167     if(mouseDownMode != NONE && (mouseDownX !=
    mouseCurrentX || mouseDownY != mouseCurrentY)){
168         int x1 = (mouseDownX > mouseCurrentX) ?
    mouseCurrentX : mouseDownX;
169         int x2 = (mouseDownX > mouseCurrentX) ?
    mouseDownX : mouseCurrentX;
170         int y1 = (mouseDownY > mouseCurrentY) ?
    mouseCurrentY : mouseDownY;
171         int y2 = (mouseDownY > mouseCurrentY) ?
    mouseDownY : mouseCurrentY;
172         painter.setPen(QPen(highlightColor, 0.5));
173         QColor bColor = highlightColor;
174         bColor.setAlpha(128);
175         painter.setBrush(QBrush(bColor));
176         painter.drawRect(x1, y1, x2 - x1, y2 - y1);
177     }
178
179     // Draw key backgrounds on a separate pixmap so that a drop shadow can be applied to them.
180     QPixmap keyBG(wWidth * ratio, wHeight * ratio);
181     keyBG.fill(QColor(0, 0, 0, 0));
182     QPainter bgPainter(&keyBG);
183     bgPainter.setRenderHints(QPainter::Antialiasing | QPainter::TextAntialiasing);
184     bgPainter.setPen(Qt::NoPen);
185     QHashIterator<QString, Key> k(keyMap);
186     uint i = -1;
187     while(k.hasNext()){
188         k.next();
189         i++;
190         const Key& key = k.value();
191         float x = key.x + offX - key.width / 2.f + 1.f;
192         float y = key.y + offY - key.height / 2.f + 1.f;
193         float w = key.width - 2.f;
194         float h = key.height - 2.f;
195         // In RGB mode, ignore keys without LEDs
196         if((_rgbMode && !key.hasLed)
```

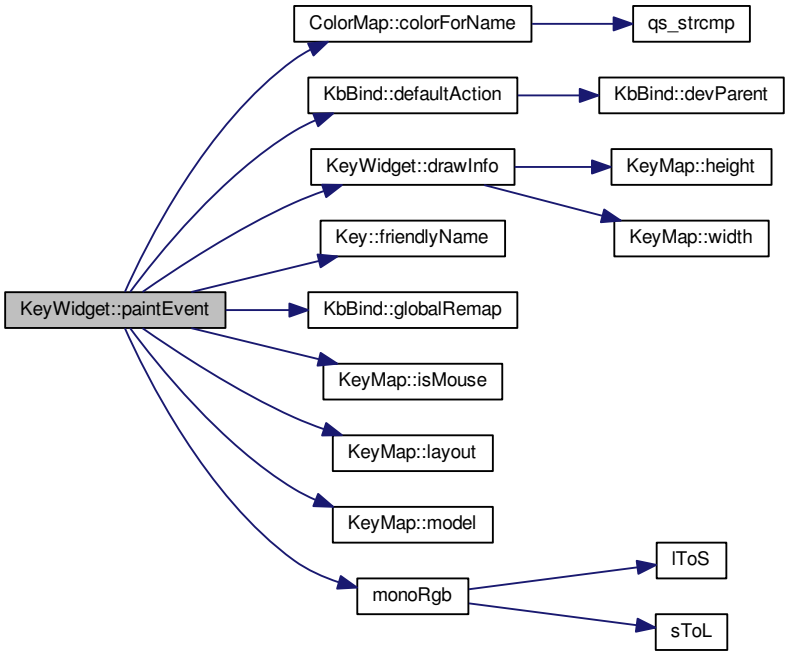
```
197                 || (!_rgbMode && !key.hasScan))
198             continue;
199         // Set color based on key highlight
200         bgPainter.setOpacity(1.);
201         if(highlight.testBit(i)){
202             if(animation.testBit(i))
203                 bgPainter.setBrush(QBrush(highlightAnimColor));
204             else
205                 bgPainter.setBrush(QBrush(highlightColor));
206         } else if(animation.testBit(i)){
207             bgPainter.setBrush(QBrush(animColor));
208         } else {
209             if(!strcmp(key.name, "sniper"))
210                 // Sniper key uses a reddish base color instead of the usual grey
211                 bgPainter.setBrush(QBrush(sniperColor));
212             else if(model == KeyMap::SCIMITAR && !strncmp(key.
    name, "thumb", 5) && strcmp(key.name, "thumb"))
213                 // Thumbgrid keys use a black color
214                 bgPainter.setBrush(QBrush(thumbColor));
215             else if(!strcmp(key.name, "lsidel") || !strcmp(key.name, "rsidel") || !strcmp(key.
    name, "logo"))
216                 // Strafe side lights have different background
217                 bgPainter.setBrush(QBrush(transparentColor));
218             else {
219                 bgPainter.setBrush(QBrush(keyColor));
220                 if(KeyMap::isMouse(model))
221                     bgPainter.setOpacity(0.7);
222             }
223         }
224         if((model != KeyMap::STRAFE && model != KeyMap::K95P) && (!strcmp(key.
    name, "mr") || !strcmp(key.name, "m1") || !strcmp(key.name, "m2") || !strcmp(key.
    name, "m3")
225                 || !strcmp(key.name, "light") || !strcmp(key.name, "lock") || (model ==
    KeyMap::K65 && !strcmp(key.name, "mute")))){
226             // Not all devices have circular buttons
227             x += w / 8.f;
228             y += h / 8.f;
229             w *= 0.75f;
230             h *= 0.75f;
231             bgPainter.drawEllipse(QRectF(x * scale, y * scale, w * scale, h * scale));
232         } else {
233             if(!strcmp(key.name, "enter")){
234                 if(key.height == 24){
235                     // ISO enter key isn't rectangular
236                     y = key.y + 1.f;
237                     h = 10.f;
238                     bgPainter.drawRect(QRectF((x + w - 13.f) * scale, y * scale, 13.f * scale, 22.f * scale
    ));
239                 } else {
240                     // US enter key isn't perfectly centered, needs an extra pixel on the left to appear
     correctly
241                     x -= 1.f;
242                     w += 1.f;
243                 }
244             } else if(!strcmp(key.name, "rshift") || !strcmp(key.name, "stop")){
245                 // A few other keys also need extra pixels
246                 x -= 1.f;
247                 w += 1.f;
248             } else if(!strcmp(key.name, "caps") || !strcmp(key.name, "lshift") || !strcmp(key.
    name, "next")){
249                 w += 1.f;
250             }
251             bgPainter.drawRect(QRectF(x * scale, y * scale, w * scale, h * scale));
252         }
253     }
254
255     // Render the key decorations (RGB -> light circles, binding -> key names) on yet another layer
256     QPixmap decoration(wWidth * ratio, wHeight * ratio);
257     decoration.fill(QColor(0, 0, 0, 0));
258     QPainter decPainter(&decoration);
259     decPainter.setRenderHints(QPainter::Antialiasing | QPainter::TextAntialiasing);
260     if(_rgbMode){
261         // Draw key colors (RGB mode)
262         QHashIterator<QString, Key> k(keyMap);
263         uint i = -1;
264         while(k.hasNext()){
265             k.next();
266             i++;
267             const Key& key = k.value();
268             if(!key.hasLed)
269                 continue;
270             float x = key.x + offX - 1.8f;
271             float y = key.y + offY - 1.8f;
272             float w = 3.6f;
273             float h = 3.6f;
274             // Display a white circle around regular keys, red circle around indicators
275             if(_indicators.contains(key.name))
```

```
276                 decPainter.setPen(QPen(QColor(255, 248, 136), 1.5));
277             else
278                 decPainter.setPen(QPen(QColor(255, 255, 255), 1.5));
279             QRgb color;
280             const QRgb* inDisplay = _displayColorMap.
    colorForName(key.name);
281             if(inDisplay)
282                 // Color in display map? Grab it from there
283                 // (monochrome conversion not necessary as this would have been done by the animation)
284                 color = *inDisplay;
285             else {
286                 // Otherwise, read from base map
287                 color = _colorMap.value(k.key());
288                 if(_monochrome)
289                     color = monoRgb(qRed(color), qGreen(color), qBlue(color));
290             }
291             decPainter.setBrush(QBrush(color));
292             if (model == KeyMap::STRAFE) { // STRAFE custom design and special keys
293                 float kx = key.x + offX - key.width / 2.f + 1.f;
294                 float ky = key.y + offY - key.height / 2.f + 1.f;
295                 float kw = key.width - 2.f;
296                 float kh = key.height - 2.f;
297                 decPainter.setPen(QPen(QColor(255, 255, 255), 1.2)); // less invasive outline to show the
     key color better
298                 if(!strcmp(key.name, "logo")) { // stylized logo
299                     float lx = key.x + offX - key.width / 2.f + 2.f;
300                     float ly = key.y + offY - key.height / 2.f + 2.f;
301                     float lw = key.width - 4.f;
302                     float lh = key.height - 4.f;
303                     QPainterPath logo;
304                     logo.moveTo(lx*scale,(ly+lh)*scale);
305                     logo.quadTo((lx+2.f)*scale,(ly+lh/2.f)*scale,lx*scale,ly*scale);
306                     logo.quadTo((lx+lw)*scale,ly*scale,(lx+lw)*scale,(ly+lh)*scale);
307                     logo.quadTo((lx+lw/2.f)*scale,(ly+lh-4.f)*scale,lx*scale,(ly+lh)*scale);
308                     decPainter.drawPath(logo);
309                     //decPainter.setPen(QPen(Qt::green, 1.2)); //QColor(125,125,125)
310                     //decPainter.drawRect(QRectF(lx * scale, ly * scale, lw * scale, lh * scale)); // don't
     really know why the 12 and 24 make it work here, but they do
311                 } else if(!strcmp(key.name, "lsidel") || !strcmp(key.name, "rsidel")) { // Strafe
     side lights (toggle lights with no animation)
312                     QRadialGradient gradient(QPointF(wWidth/2.f * ratio, wHeight/2.f * ratio), wWidth/2.f *
     ratio);//,QPointF(10, 5));
313                     gradient.setColorAt(0, color);
314                     gradient.setColorAt(0.9, color); // bring up intensity
315                     gradient.setColorAt(1, bgColor);
316                     decPainter.setBrush(QBrush(gradient));
317                     decPainter.setPen(QPen(keyColor, 1.2)); //QColor(125,125,125)
318                     decPainter.drawRect(QRectF(kx * scale, ky * scale - 12 , kw * scale, kh * scale+24));
    // don't really know why the 12 and 24 make it work here, but they do
319                 } else if(_indicators.contains(key.name)) { // FIX: This check fails
     whenever _indicators is empty because "show animated" is unchecked
320                     decPainter.setPen(QPen(QColor(0,0,0,0), 1));    // no outline for the indicators, you
     can't change their color the standard way
321                     decPainter.drawRect(QRectF((kx+2.f) * scale, (ky+2.f) * scale, (kw-4.f) * scale, (kh-4.
    f) * scale)); // square indicators
322                 } else //everything else is a circle, just a tad bigger to show the key color better
323                     decPainter.drawEllipse(QRectF((x-1.f) * scale, (y-1.f) * scale, (w+2.f) * scale, (h+2.f
    ) * scale));
324             } else
325                 decPainter.drawEllipse(QRectF(x * scale, y * scale, w * scale, h * scale));
326         }
327       } else {
328         // Draw key names
329         decPainter.setBrush(Qt::NoBrush);
330         QFont font = painter.font();
331         font.setBold(true);
332         font.setPixelSize(5.25f * scale);
333         QFont font0 = font;
334         QHashIterator<QString, Key> k(keyMap);
335         uint i = -1;
336         while(k.hasNext()){
337             k.next();
338             i++;
339             const Key& key = k.value();
340             if(!key.hasScan)
341                 continue;
342             float x = key.x + offX - key.width / 2.f + 1.f;
343             float y = key.y + offY - key.height / 2.f;
344             float w = key.width - 2.f;
345             float h = key.height;
346             // Print the key's friendly name (with some exceptions)
347             QString keyName = KbBind::globalRemap(key.name);
348             QString name = key.friendlyName(false);
349             name = name.split(" ").last();
350             struct {
351                 const char* keyName, *displayName;
352             } names[] = {
```

```
353                 {"light", ""}, {"lock", ""}, {"mute", ""}, {"volup", keyMap.
    model() == KeyMap::K65 ? "))" : ""}, {"voldn", keyMap.
    model() == KeyMap::K65 ? ")" : ""},
354                 {"prtscn",  "PrtScn\nSysRq"}, {"scroll", "Scroll\nLock"}, {"pause", "Pause\nBreak"}, {"stop
    ", ""}, {"prev", "|"}, {"play", "||"}, {"next", "|"},
355                 {"pgup", "Page\nUp"}, {"pgdn", "Page\nDown"}, {"numlock", "Num\nLock"},
356                 {"caps", "Caps"}, {"lshift", "Shift"}, {"rshift", "Shift"},
357 #ifdef Q_OS_MACX
358                 {"lctrl", ""}, {"rctrl", ""}, {"lwin", ""}, {"rwin", ""}, {"lalt", ""}, {"ralt", ""},
359 #else
360                 {"lctrl", "Ctrl"}, {"rctrl", "Ctrl"}, {"lwin", ""}, {"rwin", ""}, {"lalt", "Alt"}, {"ralt
    ", "Alt"},
361 #endif
362                 {"rmenu", ""}, {"up", ""}, {"left", ""}, {"down", ""}, {"right", ""}, {"fn","Fn"},
363                 {"mouse1", ""}, {"mouse2", ""}, {"mouse3", ""}, {"dpiup", ""}, {"dpidn", ""}, {"wheelup"
    , ""}, {"wheeldn", ""}, {"dpi", ""}, {"mouse5", ""}, {"mouse4", ""}, {"sniper", ""}
364             };
365             for(uint k = 0; k < sizeof(names) / sizeof(names[0]); k++){
366                 if(keyName == names[k].keyName){
367                     name = names[k].displayName;
368                     break;
369                 }
370             }
371             if(keyName == "thumb1" && model == KeyMap::SABRE)
372                 name = "";
373             if(keyName == "mr" || keyName == "m1" || keyName == "m2" || keyName == "m3" || keyName == "up"
    || keyName == "down" || keyName == "left" || keyName == "right")
374                 // Use a smaller size for MR, M1 - M3, and arrow keys
375                 font.setPixelSize(font.pixelSize() * 0.75);
376             else if(keyName == "end")
377                 // Use a smaller size for "End" to match everything else in that area
378                 font.setPixelSize(font.pixelSize() * 0.65);
379             else if(keyName == "light"
380 #ifndef Q_OS_MACX
381                     || keyName == "lwin" || keyName == "rwin"
382 #endif
383                     )
384                 // Use a larger font size for Super (Linux only) and Brightness to compensate for the
    unicode symbols looking smaller
385                 font.setPixelSize(font.pixelSize() * 1.3);
386             if((layout == KeyMap::EU || layout == KeyMap::EU_DVORAK) && (keyName
    == "hash" || keyName == "bslash_iso"))
387                 // Don't differentiate backslashes on the EU layout
388                 name = "\\";
389             // Determine the appropriate size to draw the text at
390             decPainter.setFont(font);
391             QRectF rect(x * scale, y * scale - 1, w * scale, h * scale);
392             int flags = Qt::AlignHCenter | Qt::AlignVCenter | Qt::TextWordWrap;
393             QRectF bounds = decPainter.boundingRect(rect, flags, name);
394             while((bounds.height() >= rect.height() - 8. || bounds.width() >= rect.width() - 2.) && font.
    pixelSize() >= 5){
395                 // Scale font size down until it fits inside the key
396                 font.setPixelSize(font.pixelSize() - 2);
397                 decPainter.setFont(font);
398                 bounds = decPainter.boundingRect(rect, flags, name);
399             }
400             // Pick color based on key function
401             QString bind = _bindMap.value(key.name);
402             QString def = KbBind::defaultAction(key.name, model);
403             if(bind.isEmpty())
404                 // Unbound - red
405                 decPainter.setPen(QColor(255, 136, 136));
406             else if(KeyAction(bind).isProgram())
407                 // Custom program - orange
408                 decPainter.setPen(QColor(255, 224, 192));
409             else if(KeyAction(bind).isSpecial() && (bind == def || !
    KeyAction(def).isSpecial()))
410                 // Special function - blue (only if not mapped to a different function - if a special
    function is remapped, color it yellow)
411                 decPainter.setPen(QColor(128, 224, 255));
412             else if(KeyAction(bind).isMedia() && (bind == def || !
    KeyAction(def).isMedia()))
413                 // Media key - green
414                 decPainter.setPen(QColor(160, 255, 168));
415             else if(bind == def)
416                 // Standard key - white
417                 decPainter.setPen(QColor(255, 255, 255));
418             else
419                 // Remapped key - yellow
420                 decPainter.setPen(QColor(255, 248, 128));
421             decPainter.drawText(rect, flags, name);
422             font = font0;
423         }
424     }
425     // Create drop shadow effects
426     QGraphicsDropShadowEffect* bgEffect = new QGraphicsDropShadowEffect;  // Have to use "new", creating
    these on the stack causes a crash...
```

```
427     bgEffect->setBlurRadius(2.);
428     bgEffect->setColor(QColor(0, 0, 0, 32));
429     bgEffect->setOffset(0, 1);
430     QGraphicsDropShadowEffect* decEffect = new QGraphicsDropShadowEffect;
431     decEffect->setBlurRadius(4.);
432     decEffect->setColor(QColor(0, 0, 0, 104));
433     decEffect->setOffset(0, 1);
434     // Apply them to the pixmaps
435     QGraphicsPixmapItem* bgItem = new QGraphicsPixmapItem(keyBG);
436     bgItem->setGraphicsEffect(bgEffect);
437     QGraphicsPixmapItem* decItem = new QGraphicsPixmapItem(decoration);
438     decItem->setGraphicsEffect(decEffect);
439     // Render everything
440     QGraphicsScene* scene = new QGraphicsScene;
441     scene->addItem(bgItem);
442     scene->addItem(decItem);
443     // It has to be rendered onto yet another pixmap or else DPI scaling will look terrible...
444     QPixmap final(wWidth * ratio, wHeight * ratio);
445     final.fill(QColor(0, 0, 0, 0));
446     QPainter finalPainter(&final);
447     scene->render(&finalPainter, QRectF(0, 0, wWidth * ratio, wHeight * ratio), QRectF(0, 0, wWidth * ratio
    , wHeight * ratio));
448     delete scene;   // <- Automatically cleans up the rest of the objects
449 #if QT_VERSION >= QT_VERSION_CHECK(5, 3, 0)
450     final.setDevicePixelRatio(ratio);
451 #endif
452     painter.drawPixmap(QPointF(0., 0.), final);
453 }
```

Here is the call graph for this function:



12.44.5.17 bool KeyWidget::rgbMode ( ) [inline]

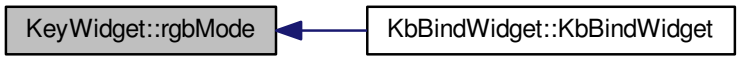
Definition at line 17 of file keywidget.h.

References _rgbMode.

Referenced by KbBindWidget::KbBindWidget().

```
17 { return _rgbMode; }
```

Here is the caller graph for this function:



12.44.5.18 void KeyWidget::rgbMode (bool *newRgbMode*) [inline]

Definition at line 18 of file keywidget.h.

References _rgbMode.

```
18 { _rgbMode = newRgbMode; update(); }
```

12.44.5.19 void KeyWidget::selectAll ( )

Definition at line 596 of file keywidget.cpp.

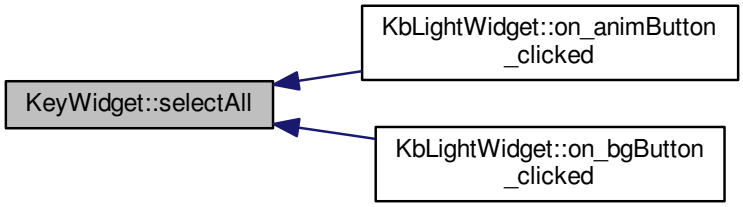
References _rgbMode, Key::hasLed, Key::hasScan, keyMap, mouseDownMode, Key::name, newSelection, NONE, KeyMap::positions(), selection, and selectionChanged().

Referenced by KbLightWidget::on_animButton_clicked(), and KbLightWidget::on_bgButton_clicked().

```
596                          {
597     selection.fill(false);
598     // Fill the selection with all keys that have an LED/scancode (depending on widget mode)
599     int i = 0;
600     QStringList selectedNames;
601     foreach(const Key& key, keyMap.positions()){
602         // Sidelights can't be selected
603         if(strcmp(key.name, "lsidel") && strcmp(key.name, "rsidel")
604            && ((_rgbMode && key.hasLed) || !(_rgbMode && key.
      hasScan))){
605             selection.setBit(i);
606             selectedNames << key.name;
607         }
608         i++;
609     }
610     // Clear mousedown state
611     newSelection.fill(false);
612     mouseDownMode = NONE;
613     update();
614     emit selectionChanged(selectedNames);
615 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



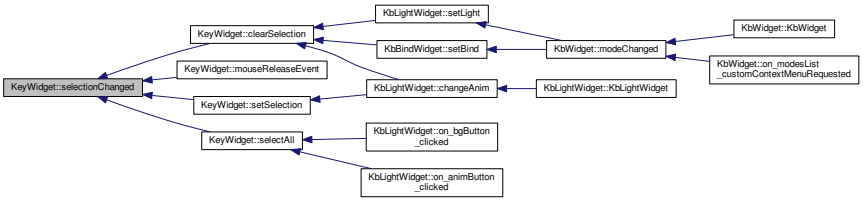
12.44.5.20 void KeyWidget::selectionChanged (QStringList *selected*) [signal]

Definition at line 164 of file moc_keywidget.cpp.

Referenced by clearSelection(), mouseReleaseEvent(), selectAll(), and setSelection().

```
165 {
166     void *_a[] = { Q_NULLPTR, const_cast<void*>(reinterpret_cast<const void*>(&_t1)) };
167     QMetaObject::activate(this, &staticMetaObject, 0, _a);
168 }
```

Here is the caller graph for this function:



12.44.5.21 void KeyWidget::setAnimation (const QStringList & *keys*)

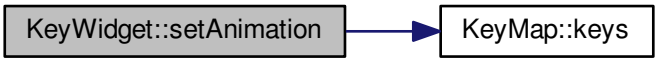
Definition at line 625 of file keywidget.cpp.

References animation, keyMap, and KeyMap::keys().

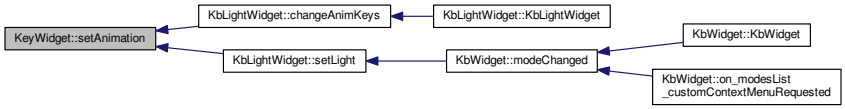
Referenced by KbLightWidget::changeAnimKeys(), and KbLightWidget::setLight().

```
625                                                 {
626     animation.fill(false);
627     QStringList allNames = keyMap.keys();
628     foreach(const QString& key, keys){
629         // Sidelights can't be selected
630         if(!strcmp(key.toLatin1(), "lsidel") || !strcmp(key.toLatin1(), "rsidel"))
631             continue;
632         int index = allNames.indexOf(key);
633         if(index >= 0)
634             animation.setBit(index);
635     }
636     update();
637 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.44.5.22 void KeyWidget::setAnimationToSelection ( )

Definition at line 639 of file keywidget.cpp.

References animation, and selection.

Referenced by KbLightWidget::changeAnim().

```
639                                         {
640     animation = selection;
641     update();
642 }
```

Here is the caller graph for this function:

12.44.5.23 void KeyWidget::setSelection (const QStringList & *keys*)

Definition at line 582 of file keywidget.cpp.

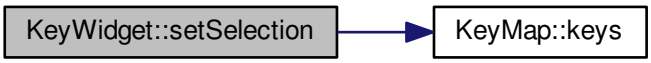
References keyMap, KeyMap::keys(), mouseDownMode, newSelection, NONE, selection, and selectionChanged().

Referenced by KbLightWidget::changeAnim().

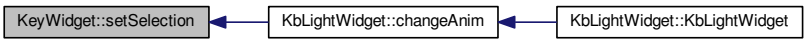
```
582                                                 {
583     selection.fill(false);
584     QStringList allNames = keyMap.keys();
585     foreach(const QString& key, keys){
586         int index = allNames.indexOf(key);
587         if(index >= 0)
```

```
588             selection.setBit(index);
589     }
590     newSelection.fill(false);
591     mouseDownMode = NONE;
592     update();
593     emit selectionChanged(keys);
594 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.44.5.24 void KeyWidget::sidelightToggled ( ) [signal]

Definition at line 171 of file moc_keywidget.cpp.

Referenced by mousePressEvent().

```
172 {
173     QMetaObject::activate(this, &staticMetaObject, 1, Q_NULLPTR);
174 }
```

Here is the caller graph for this function:

12.44.6 Field Documentation

12.44.6.1 BindMap KeyWidget::_bindMap [private]

Definition at line 58 of file keywidget.h.

Referenced by bindMap(), and paintEvent().

12.44.6.2 QColorMap KeyWidget::_colorMap [private]

Definition at line 56 of file keywidget.h.

Referenced by colorMap(), and paintEvent().

12.44.6.3 ColorMap KeyWidget::_displayColorMap [private]

Definition at line 57 of file keywidget.h.

Referenced by displayColorMap(), and paintEvent().

12.44.6.4 QSet<QString> KeyWidget::_indicators [private]

Definition at line 59 of file keywidget.h.

Referenced by displayColorMap(), mouseMoveEvent(), and paintEvent().

12.44.6.5 bool KeyWidget::_monochrome [private]

Definition at line 73 of file keywidget.h.

Referenced by monochrome(), and paintEvent().

12.44.6.6 bool KeyWidget::_rgbMode [private]

Definition at line 73 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), paintEvent(), rgbMode(), and selectAll().

12.44.6.7 QBitArray KeyWidget::animation [private]

Definition at line 63 of file keywidget.h.

Referenced by clearAnimation(), map(), paintEvent(), setAnimation(), and setAnimationToSelection().

12.44.6.8 KeyMap KeyWidget::keyMap [private]

Definition at line 55 of file keywidget.h.

Referenced by drawInfo(), map(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), selectAll(), setAnimation(), and setSelection().

12.44.6.9 int KeyWidget::mouseCurrentX [private]

Definition at line 65 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

12.44.6.10 int KeyWidget::mouseCurrentY [private]

Definition at line 65 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

12.44.6.11 enum { ... } KeyWidget::mouseDownMode [private]

Referenced by clearSelection(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), selectAll(), and setSelection().

12.44.6.12 int KeyWidget::mouseDownX [private]

Definition at line 64 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

12.44.6.13 int KeyWidget::mouseDownY [private]

Definition at line 64 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

12.44.6.14 QBitArray KeyWidget::newSelection [private]

Definition at line 62 of file keywidget.h.

Referenced by clearSelection(), map(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), selectAll(), and setSelection().

12.44.6.15 QBitArray KeyWidget::selection [private]

Definition at line 61 of file keywidget.h.

Referenced by clearSelection(), map(), mouseReleaseEvent(), paintEvent(), selectAll(), setAnimationToSelection(), and setSelection().

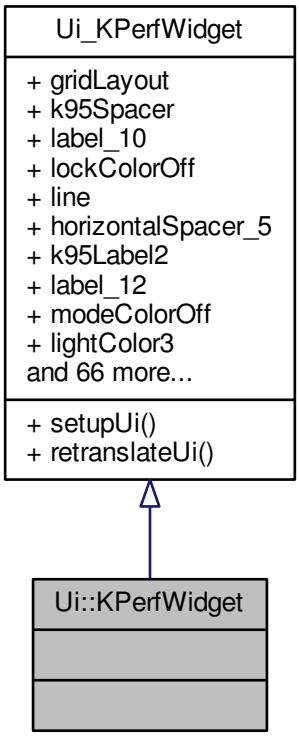
The documentation for this class was generated from the following files:

- src/ckb/keywidget.h
- src/ckb/keywidget.cpp
- src/ckb/release/moc_keywidget.cpp

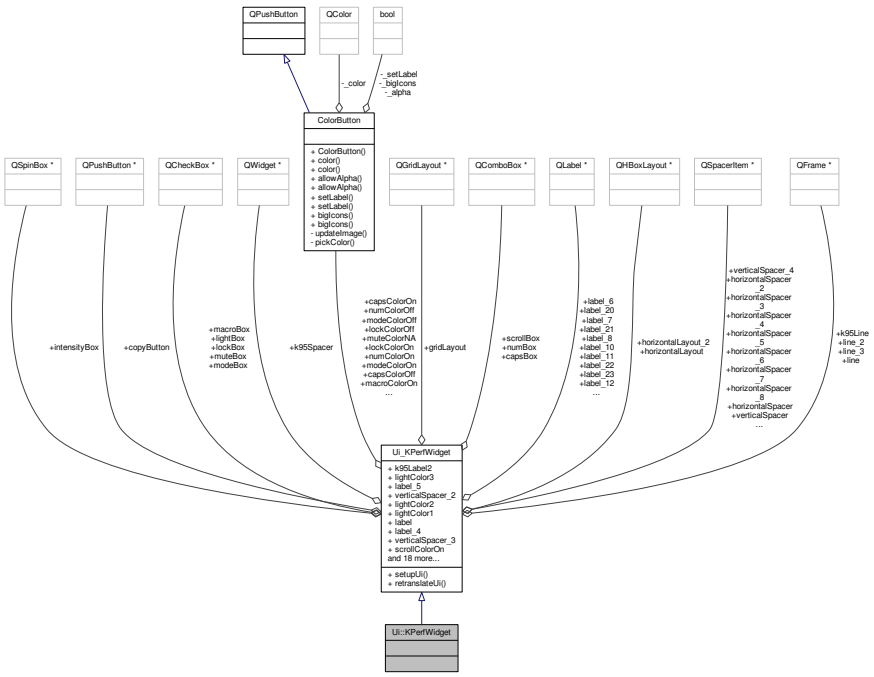
12.45 Ui::KPerfWidget Class Reference

`#include <src/ckb/ui_kperfwidget.h>`

Inheritance diagram for Ui::KPerfWidget:



Collaboration diagram for Ui::KPerfWidget:



Public Member Functions

- void setupUi (QWidget *KPerfWidget)
- void retranslateUi (QWidget *KPerfWidget)

Data Fields

- QGridLayout * gridLayout
- QWidget * k95Spacer
- QLabel * label_10
- ColorButton * lockColorOff
- QFrame * line
- QSpacerItem * horizontalSpacer_5
- QLabel * k95Label2
- QLabel * label_12
- ColorButton * modeColorOff
- ColorButton * lightColor3
- QSpacerItem * horizontalSpacer
- QCheckBox * lockBox
- ColorButton * modeColorOn
- QCheckBox * macroBox
- ColorButton * numColorOn
- QComboBox * numBox
- QLabel * label_5
- QSpacerItem * verticalSpacer_2
- ColorButton * lightColor2
- QSpacerItem * horizontalSpacer_7
- QLabel * label_8

- QCheckBox * lightBox
- ColorButton * lightColor1
- QSpacerItem * verticalSpacer_4
- QLabel * label_6
- QLabel * label
- QSpacerItem * horizontalSpacer_6
- ColorButton * lockColorOn
- QLabel * label_4
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_3
- QCheckBox * modeBox
- ColorButton * capsColorOff
- ColorButton * numColorOff
- ColorButton * capsColorOn
- QFrame * line_3
- QLabel * label_22
- QComboBox * scrollBox
- QLabel * label_7
- ColorButton * scrollColorOn
- QComboBox * capsBox
- ColorButton * scrollColorOff
- QCheckBox * muteBox
- ColorButton * muteColorOn
- ColorButton * macroColorOff
- QLabel * label_3
- ColorButton * muteColorOff
- QLabel * k95Label3
- ColorButton * macroColorOn
- ColorButton * muteColorNA
- QLabel * label_19
- QFrame * k95Line
- QLabel * label_21
- QLabel * label_24
- QLabel * label_20
- QLabel * label_16
- QLabel * label_14
- QLabel * label_18
- QSpacerItem * horizontalSpacer_2
- QLabel * label_15
- QLabel * k95Label4
- QLabel * label_17
- QLabel * label_11
- QLabel * label_23
- QLabel * k95Label1
- QHBoxLayout * horizontalLayout_2
- QPushButton * copyButton
- QSpacerItem * horizontalSpacer_8
- QHBoxLayout * horizontalLayout
- QLabel * label_2
- QSpacerItem * horizontalSpacer_4
- QSpinBox * intensityBox
- QSpacerItem * horizontalSpacer_3
- QLabel * k95Label5
- QFrame * line_2
- QLabel * k95Label6

12.45.1 Detailed Description

Definition at line 704 of file ui_kperfwidget.h.

12.45.2 Member Function Documentation

12.45.2.1 void Ui_KPerfWidget::retranslateUi (QWidget * *KPerfWidget*) [inline],[inherited]

Definition at line 616 of file ui_kperfwidget.h.

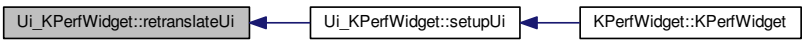
References Ui_KPerfWidget::capsBox, Ui_KPerfWidget::capsColorOff, Ui_KPerfWidget::capsColorOn, Ui_KPerfWidget::copyButton, Ui_KPerfWidget::intensityBox, Ui_KPerfWidget::k95Label1, Ui_KPerfWidget::k95Label2, Ui_KPerfWidget::k95Label3, Ui_KPerfWidget::k95Label4, Ui_KPerfWidget::k95Label5, Ui_KPerfWidget::k95Label6, Ui_KPerfWidget::label, Ui_KPerfWidget::label_10, Ui_KPerfWidget::label_11, Ui_KPerfWidget::label_12, Ui_KPerfWidget::label_14, Ui_KPerfWidget::label_15, Ui_KPerfWidget::label_16, Ui_KPerfWidget::label_17, Ui_KPerfWidget::label_18, Ui_KPerfWidget::label_19, Ui_KPerfWidget::label_2, Ui_KPerfWidget::label_20, Ui_KPerfWidget::label_21, Ui_KPerfWidget::label_22, Ui_KPerfWidget::label_23, Ui_KPerfWidget::label_24, Ui_KPerfWidget::label_3, Ui_KPerfWidget::label_4, Ui_KPerfWidget::label_5, Ui_KPerfWidget::label_6, Ui_KPerfWidget::label_7, Ui_KPerfWidget::label_8, Ui_KPerfWidget::lightBox, Ui_KPerfWidget::lightColor1, Ui_KPerfWidget::lightColor2, Ui_KPerfWidget::lightColor3, Ui_KPerfWidget::lockBox, Ui_KPerfWidget::lockColorOff, Ui_KPerfWidget::lockColorOn, Ui_KPerfWidget::macroBox, Ui_KPerfWidget::macroColorOff, Ui_KPerfWidget::macroColorOn, Ui_KPerfWidget::modeBox, Ui_KPerfWidget::modeColorOff, Ui_KPerfWidget::modeColorOn, Ui_KPerfWidget::muteBox, Ui_KPerfWidget::muteColorNA, Ui_KPerfWidget::muteColorOff, Ui_KPerfWidget::muteColorOn, Ui_KPerfWidget::numBox, Ui_KPerfWidget::numColorOff, Ui_KPerfWidget::numColorOn, Ui_KPerfWidget::scrollBox, Ui_KPerfWidget::scrollColorOff, and Ui_KPerfWidget::scrollColorOn.

Referenced by Ui_KPerfWidget::setupUi().

```
617     {
618         KPerfWidget->setWindowTitle(QApplication::translate("KPerfWidget", "Form", Q_NULLPTR));
619         label_10->setText(QApplication::translate("KPerfWidget", "100%", Q_NULLPTR));
620         lockColorOff->setText(QString());
621         k95Label2->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
622         label_12->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
623         modeColorOff->setText(QString());
624         lightColor3->setText(QString());
625         lockBox->setText(QApplication::translate("KPerfWidget", "Indicate Windows Lock:", Q_NULLPTR)
    );
626         modeColorOn->setText(QString());
627         macroBox->setText(QApplication::translate("KPerfWidget", "Indicate macro recording:",
    Q_NULLPTR));
628         numColorOn->setText(QString());
629         numBox->clear();
630         numBox->insertItems(0, QStringList()
631          << QApplication::translate("KPerfWidget", "Normal", Q_NULLPTR)
632          << QApplication::translate("KPerfWidget", "Always on", Q_NULLPTR)
633          << QApplication::translate("KPerfWidget", "Always off", Q_NULLPTR)
634          << QApplication::translate("KPerfWidget", "RGB indicator", Q_NULLPTR)
635          << QApplication::translate("KPerfWidget", "RGB + normal", Q_NULLPTR)
636         );
637         label_5->setText(QApplication::translate("KPerfWidget", "Num lock:", Q_NULLPTR));
638         lightColor2->setText(QString());
639         label_8->setText(QApplication::translate("KPerfWidget", "Note: many desktop environments do
     not use scroll lock. The indicator may not turn on even when enabled.", Q_NULLPTR));
640         lightBox->setText(QApplication::translate("KPerfWidget", "Indicate brightness:", Q_NULLPTR)
    );
641         lightColor1->setText(QString());
642         label_6->setText(QApplication::translate("KPerfWidget", "Caps lock:", Q_NULLPTR));
643         label->setText(QApplication::translate("KPerfWidget", "Miscellaneous", Q_NULLPTR));
644         lockColorOn->setText(QString());
645         label_4->setText(QApplication::translate("KPerfWidget", "Lock lights", Q_NULLPTR));
646         modeBox->setText(QApplication::translate("KPerfWidget", "Indicate current mode:", Q_NULLPTR)
    );
647         capsColorOff->setText(QString());
648         numColorOff->setText(QString());
649         capsColorOn->setText(QString());
650         label_22->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
651         scrollBox->clear();
652         scrollBox->insertItems(0, QStringList()
653          << QApplication::translate("KPerfWidget", "Normal", Q_NULLPTR)
654          << QApplication::translate("KPerfWidget", "Always on", Q_NULLPTR)
655          << QApplication::translate("KPerfWidget", "Always off", Q_NULLPTR)
656          << QApplication::translate("KPerfWidget", "RGB indicator", Q_NULLPTR)
```

```
657          << QApplication::translate("KPerfWidget", "RGB + normal", Q_NULLPTR)
658         );
659         label_7->setText(QApplication::translate("KPerfWidget", "Scroll lock:", Q_NULLPTR));
660         scrollColorOn->setText(QString());
661         capsBox->clear();
662         capsBox->insertItems(0, QStringList()
663          << QApplication::translate("KPerfWidget", "Normal", Q_NULLPTR)
664          << QApplication::translate("KPerfWidget", "Always on", Q_NULLPTR)
665          << QApplication::translate("KPerfWidget", "Always off", Q_NULLPTR)
666          << QApplication::translate("KPerfWidget", "RGB indicator", Q_NULLPTR)
667          << QApplication::translate("KPerfWidget", "RGB + normal", Q_NULLPTR)
668         );
669         scrollColorOff->setText(QString());
670         muteBox->setText(QApplication::translate("KPerfWidget", "Indicate mute:", Q_NULLPTR));
671         muteColorOn->setText(QString());
672         macroColorOff->setText(QString());
673         label_3->setText(QApplication::translate("KPerfWidget", "Top row", Q_NULLPTR));
674         muteColorOff->setText(QString());
675         k95Label3->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
676         macroColorOn->setText(QString());
677 #ifndef QT_NO_TOOLTIP
678         muteColorNA->setToolTip(QApplication::translate("KPerfWidget", "If the sound device does
       not support muting, or if mute status is unknown, this color will be displayed.", Q_NULLPTR));
679 #endif // QT_NO_TOOLTIP
680         muteColorNA->setText(QString());
681         label_19->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
682         label_21->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
683         label_24->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
684         label_20->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
685         label_16->setText(QApplication::translate("KPerfWidget", "67%", Q_NULLPTR));
686         label_14->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
687         label_18->setText(QApplication::translate("KPerfWidget", "33%", Q_NULLPTR));
688         label_15->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
689         k95Label4->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
690         label_17->setText(QApplication::translate("KPerfWidget", "Unknown", Q_NULLPTR));
691         label_11->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
692         label_23->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
693         k95Label1->setText(QApplication::translate("KPerfWidget", "K95 buttons", Q_NULLPTR));
694         copyButton->setText(QApplication::translate("KPerfWidget", "Copy to mode...", Q_NULLPTR))
    ;
695         label_2->setText(QApplication::translate("KPerfWidget", "Indicator intensity:", Q_NULLPTR));
696         intensityBox->setSuffix(QApplication::translate("KPerfWidget", "%", Q_NULLPTR));
697         k95Label5->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
698         k95Label6->setText(QApplication::translate("KPerfWidget", "Note: macro recording not yet
      implemented. Indicator will always display \"off\".", Q_NULLPTR));
699     } // retranslateUi
```

Here is the caller graph for this function:



12.45.2.2 void Ui_KPerfWidget::setupUi (QWidget * *KPerfWidget*) [inline], [inherited]

Definition at line 111 of file ui_kperfwidget.h.

References Ui_KPerfWidget::capsBox, Ui_KPerfWidget::capsColorOff, Ui_KPerfWidget::capsColorOn, Ui_KPerfWidget::copyButton, Ui_KPerfWidget::gridLayout, Ui_KPerfWidget::horizontalLayout, Ui_KPerfWidget::horizontalLayout_2, Ui_KPerfWidget::horizontalSpacer, Ui_KPerfWidget::horizontalSpacer_2, Ui_KPerfWidget::horizontalSpacer_3, Ui_KPerfWidget::horizontalSpacer_4, Ui_KPerfWidget::horizontalSpacer_5, Ui_KPerfWidget::horizontalSpacer_6, Ui_KPerfWidget::horizontalSpacer_7, Ui_KPerfWidget::horizontalSpacer_8, Ui_KPerfWidget::intensityBox, Ui_KPerfWidget::k95Label1, Ui_KPerfWidget::k95Label2, Ui_KPerfWidget::k95Label3, Ui_KPerfWidget::k95Label4, Ui_KPerfWidget::k95Label5, Ui_KPerfWidget::k95Label6, Ui_KPerfWidget::k95Line, Ui_KPerfWidget::k95Spacer, Ui_KPerfWidget::label, Ui_KPerfWidget::label_10, Ui_KPerfWidget::label_11, Ui_KPerfWidget::label_12, Ui_KPerfWidget::label_14, Ui_KPerfWidget::label_15, Ui_KPerfWidget::label_16, Ui_KPerfWidget::label_17, Ui_KPerfWidget::label_18, Ui_KPerfWidget::label_19, Ui_KPerfWidget::label_2, Ui_KPerfWidget::label_20, Ui_KPerfWidget::label_21, Ui_KPerfWidget::label_22, Ui_KPerfWidget::label_23, Ui_KPerfWidget::label_24, Ui_KPerfWidget::label_3, Ui_KPerfWidget::label_4, Ui_KPerfWidget::label_5, Ui_KPerfWidget::label_6, Ui_KPerfWidget-

::label_7, Ui_KPerfWidget::label_8, Ui_KPerfWidget::lightBox, Ui_KPerfWidget::lightColor1, Ui_KPerfWidget::lightColor2, Ui_KPerfWidget::lightColor3, Ui_KPerfWidget::line, Ui_KPerfWidget::line_2, Ui_KPerfWidget::line_3, Ui_KPerfWidget::lockBox, Ui_KPerfWidget::lockColorOff, Ui_KPerfWidget::lockColorOn, Ui_KPerfWidget::macroBox, Ui_KPerfWidget::macroColorOff, Ui_KPerfWidget::macroColorOn, Ui_KPerfWidget::modeBox, Ui_KPerfWidget::modeColorOff, Ui_KPerfWidget::modeColorOn, Ui_KPerfWidget::muteBox, Ui_KPerfWidget::muteColorNA, Ui_KPerfWidget::muteColorOff, Ui_KPerfWidget::muteColorOn, Ui_KPerfWidget::numBox, Ui_KPerfWidget::numColorOff, Ui_KPerfWidget::numColorOn, Ui_KPerfWidget::retranslateUi(), Ui_KPerfWidget::scrollBox, Ui_KPerfWidget::scrollColorOff, Ui_KPerfWidget::scrollColorOn, Ui_KPerfWidget::verticalSpacer, Ui_KPerfWidget::verticalSpacer_2, Ui_KPerfWidget::verticalSpacer_3, and Ui_KPerfWidget::verticalSpacer_4.

Referenced by KPerfWidget::KPerfWidget().

```
112     {
113         if (KPerfWidget->objectName().isEmpty())
114             KPerfWidget->setObjectName(QStringLiteral("KPerfWidget"));
115         KPerfWidget->resize(840, 679);
116         gridLayout = new QGridLayout(KPerfWidget);
117         gridLayout->setObjectName(QStringLiteral("gridLayout"));
118         k95Spacer = new QWidget(KPerfWidget);
119         k95Spacer->setObjectName(QStringLiteral("k95Spacer"));
120         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
121         sizePolicy.setHorizontalStretch(0);
122         sizePolicy.setVerticalStretch(0);
123         sizePolicy.setHeightForWidth(k95Spacer->sizePolicy().hasHeightForWidth());
124         k95Spacer->setSizePolicy(sizePolicy);
125         k95Spacer->setMinimumSize(QSize(15, 10));
126         k95Spacer->setMaximumSize(QSize(15, 10));
127 
128         gridLayout->addWidget(k95Spacer, 12, 4, 1, 1);
129 
130         label_10 = new QLabel(KPerfWidget);
131         label_10->setObjectName(QStringLiteral("label_10"));
132 
133         gridLayout->addWidget(label_10, 15, 5, 1, 1);
134 
135         lockColorOff = new ColorButton(KPerfWidget);
136         lockColorOff->setObjectName(QStringLiteral("lockColorOff"));
137         sizePolicy.setHeightForWidth(lockColorOff->sizePolicy().hasHeightForWidth());
138         lockColorOff->setSizePolicy(sizePolicy);
139         lockColorOff->setMinimumSize(QSize(0, 40));
140         lockColorOff->setMaximumSize(QSize(16777215, 40));
141 
142         gridLayout->addWidget(lockColorOff, 16, 7, 1, 1);
143 
144         line = new QFrame(KPerfWidget);
145         line->setObjectName(QStringLiteral("line"));
146         line->setFrameShape(QFrame::HLine);
147         line->setFrameShadow(QFrame::Sunken);
148 
149         gridLayout->addWidget(line, 2, 0, 1, 13);
150 
151         horizontalSpacer_5 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
152 
153         gridLayout->addItem(horizontalSpacer_5, 17, 6, 1, 1);
154 
155         k95Label2 = new QLabel(KPerfWidget);
156         k95Label2->setObjectName(QStringLiteral("k95Label2"));
157 
158         gridLayout->addWidget(k95Label2, 9, 5, 1, 1);
159 
160         label_12 = new QLabel(KPerfWidget);
161         label_12->setObjectName(QStringLiteral("label_12"));
162 
163         gridLayout->addWidget(label_12, 17, 5, 1, 1);
164 
165         modeColorOff = new ColorButton(KPerfWidget);
166         modeColorOff->setObjectName(QStringLiteral("modeColorOff"));
167         sizePolicy.setHeightForWidth(modeColorOff->sizePolicy().hasHeightForWidth());
168         modeColorOff->setSizePolicy(sizePolicy);
169         modeColorOff->setMinimumSize(QSize(0, 40));
170         modeColorOff->setMaximumSize(QSize(16777215, 40));
171 
172         gridLayout->addWidget(modeColorOff, 9, 7, 1, 1);
173 
174         lightColor3 = new ColorButton(KPerfWidget);
175         lightColor3->setObjectName(QStringLiteral("lightColor3"));
176         sizePolicy.setHeightForWidth(lightColor3->sizePolicy().hasHeightForWidth());
177         lightColor3->setSizePolicy(sizePolicy);
178         lightColor3->setMinimumSize(QSize(0, 40));
179         lightColor3->setMaximumSize(QSize(16777215, 40));
180 
181         gridLayout->addWidget(lightColor3, 15, 4, 1, 1);
```

```
182
183         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
184
185         gridLayout->addItem(horizontalSpacer, 15, 12, 1, 1);
186
187         lockBox = new QCheckBox(KPerfWidget);
188         lockBox->setObjectName(QStringLiteral("lockBox"));
189
190         gridLayout->addWidget(lockBox, 16, 1, 1, 2);
191
192         modeColorOn = new ColorButton(KPerfWidget);
193         modeColorOn->setObjectName(QStringLiteral("modeColorOn"));
194         sizePolicy.setHeightForWidth(modeColorOn->sizePolicy().hasHeightForWidth());
195         modeColorOn->setSizePolicy(sizePolicy);
196         modeColorOn->setMinimumSize(QSize(0, 40));
197         modeColorOn->setMaximumSize(QSize(16777215, 40));
198
199         gridLayout->addWidget(modeColorOn, 9, 4, 1, 1);
200
201         macroBox = new QCheckBox(KPerfWidget);
202         macroBox->setObjectName(QStringLiteral("macroBox"));
203
204         gridLayout->addWidget(macroBox, 10, 1, 1, 2);
205
206         numColorOn = new ColorButton(KPerfWidget);
207         numColorOn->setObjectName(QStringLiteral("numColorOn"));
208         sizePolicy.setHeightForWidth(numColorOn->sizePolicy().hasHeightForWidth());
209         numColorOn->setSizePolicy(sizePolicy);
210         numColorOn->setMinimumSize(QSize(0, 40));
211         numColorOn->setMaximumSize(QSize(16777215, 40));
212
213         gridLayout->addWidget(numColorOn, 21, 4, 1, 1);
214
215         numBox = new QComboBox(KPerfWidget);
216         numBox->setObjectName(QStringLiteral("numBox"));
217
218         gridLayout->addWidget(numBox, 21, 2, 1, 1);
219
220         label_5 = new QLabel(KPerfWidget);
221         label_5->setObjectName(QStringLiteral("label_5"));
222
223         gridLayout->addWidget(label_5, 21, 1, 1, 1);
224
225         verticalSpacer_2 = new QSpacerItem(20, 20, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
226
227         gridLayout->addItem(verticalSpacer_2, 4, 1, 1, 1);
228
229         lightColor2 = new ColorButton(KPerfWidget);
230         lightColor2->setObjectName(QStringLiteral("lightColor2"));
231         sizePolicy.setHeightForWidth(lightColor2->sizePolicy().hasHeightForWidth());
232         lightColor2->setSizePolicy(sizePolicy);
233         lightColor2->setMinimumSize(QSize(0, 40));
234         lightColor2->setMaximumSize(QSize(16777215, 40));
235
236         gridLayout->addWidget(lightColor2, 15, 7, 1, 1);
237
238         horizontalSpacer_7 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
239
240         gridLayout->addItem(horizontalSpacer_7, 17, 3, 1, 1);
241
242         label_8 = new QLabel(KPerfWidget);
243         label_8->setObjectName(QStringLiteral("label_8"));
244         QFont font;
245         font.setPointSize(8);
246         font.setItalic(true);
247         label_8->setFont(font);
248
249         gridLayout->addWidget(label_8, 25, 1, 1, 12);
250
251         lightBox = new QCheckBox(KPerfWidget);
252         lightBox->setObjectName(QStringLiteral("lightBox"));
253
254         gridLayout->addWidget(lightBox, 15, 1, 1, 2);
255
256         lightColor1 = new ColorButton(KPerfWidget);
257         lightColor1->setObjectName(QStringLiteral("lightColor1"));
258         sizePolicy.setHeightForWidth(lightColor1->sizePolicy().hasHeightForWidth());
259         lightColor1->setSizePolicy(sizePolicy);
260         lightColor1->setMinimumSize(QSize(0, 40));
261         lightColor1->setMaximumSize(QSize(16777215, 40));
262
263         gridLayout->addWidget(lightColor1, 15, 10, 1, 1);
264
265         verticalSpacer_4 = new QSpacerItem(0, 5, QSizePolicy::Minimum, QSizePolicy::Fixed);
```

```
266
267         gridLayout->addItem(verticalSpacer_4, 24, 4, 1, 1);
268
269         label_6 = new QLabel(KPerfWidget);
270         label_6->setObjectName(QStringLiteral("label_6"));
271
272         gridLayout->addWidget(label_6, 22, 1, 1, 1);
273
274         label = new QLabel(KPerfWidget);
275         label->setObjectName(QStringLiteral("label"));
276         QFont font1;
277         font1.setBold(true);
278         font1.setWeight(75);
279         label->setFont(font1);
280
281         gridLayout->addWidget(label, 1, 0, 1, 13);
282
283         horizontalSpacer_6 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
284
285         gridLayout->addItem(horizontalSpacer_6, 17, 9, 1, 1);
286
287         lockColorOn = new ColorButton(KPerfWidget);
288         lockColorOn->setObjectName(QStringLiteral("lockColorOn"));
289         sizePolicy.setHeightForWidth(lockColorOn->sizePolicy().hasHeightForWidth());
290         lockColorOn->setSizePolicy(sizePolicy);
291         lockColorOn->setMinimumSize(QSize(0, 40));
292         lockColorOn->setMaximumSize(QSize(16777215, 40));
293
294         gridLayout->addWidget(lockColorOn, 16, 4, 1, 1);
295
296         label_4 = new QLabel(KPerfWidget);
297         label_4->setObjectName(QStringLiteral("label_4"));
298         label_4->setFont(font1);
299
300         gridLayout->addWidget(label_4, 19, 0, 1, 13);
301
302         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
303
304         gridLayout->addItem(verticalSpacer, 26, 2, 1, 1);
305
306         verticalSpacer_3 = new QSpacerItem(20, 20, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
307
308         gridLayout->addItem(verticalSpacer_3, 18, 4, 1, 1);
309
310         modeBox = new QCheckBox(KPerfWidget);
311         modeBox->setObjectName(QStringLiteral("modeBox"));
312
313         gridLayout->addWidget(modeBox, 9, 1, 1, 2);
314
315         capsColorOff = new ColorButton(KPerfWidget);
316         capsColorOff->setObjectName(QStringLiteral("capsColorOff"));
317         sizePolicy.setHeightForWidth(capsColorOff->sizePolicy().hasHeightForWidth());
318         capsColorOff->setSizePolicy(sizePolicy);
319         capsColorOff->setMinimumSize(QSize(0, 40));
320         capsColorOff->setMaximumSize(QSize(16777215, 40));
321
322         gridLayout->addWidget(capsColorOff, 22, 7, 1, 1);
323
324         numColorOff = new ColorButton(KPerfWidget);
325         numColorOff->setObjectName(QStringLiteral("numColorOff"));
326         QSizePolicy sizePolicy1(QSizePolicy::Fixed, QSizePolicy::Fixed);
327         sizePolicy1.setHorizontalStretch(0);
328         sizePolicy1.setVerticalStretch(40);
329         sizePolicy1.setHeightForWidth(numColorOff->sizePolicy().hasHeightForWidth());
330         numColorOff->setSizePolicy(sizePolicy1);
331         numColorOff->setMaximumSize(QSize(16777215, 40));
332
333         gridLayout->addWidget(numColorOff, 21, 7, 1, 1);
334
335         capsColorOn = new ColorButton(KPerfWidget);
336         capsColorOn->setObjectName(QStringLiteral("capsColorOn"));
337         sizePolicy.setHeightForWidth(capsColorOn->sizePolicy().hasHeightForWidth());
338         capsColorOn->setSizePolicy(sizePolicy);
339         capsColorOn->setMinimumSize(QSize(0, 40));
340         capsColorOn->setMaximumSize(QSize(16777215, 40));
341
342         gridLayout->addWidget(capsColorOn, 22, 4, 1, 1);
343
344         line_3 = new QFrame(KPerfWidget);
345         line_3->setObjectName(QStringLiteral("line_3"));
346         line_3->setFrameShape(QFrame::HLine);
347         line_3->setFrameShadow(QFrame::Sunken);
348
349         gridLayout->addWidget(line_3, 20, 0, 1, 13);
```

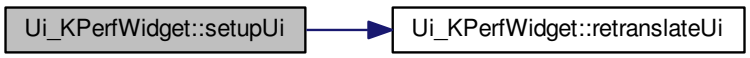
```
350
351         label_22 = new QLabel(KPerfWidget);
352         label_22->setObjectName(QStringLiteral("label_22"));
353
354         gridLayout->addWidget(label_22, 21, 8, 1, 1);
355
356         scrollBox = new QComboBox(KPerfWidget);
357         scrollBox->setObjectName(QStringLiteral("scrollBox"));
358
359         gridLayout->addWidget(scrollBox, 23, 2, 1, 1);
360
361         label_7 = new QLabel(KPerfWidget);
362         label_7->setObjectName(QStringLiteral("label_7"));
363
364         gridLayout->addWidget(label_7, 23, 1, 1, 1);
365
366         scrollColorOn = new ColorButton(KPerfWidget);
367         scrollColorOn->setObjectName(QStringLiteral("scrollColorOn"));
368         sizePolicy.setHeightForWidth(scrollColorOn->sizePolicy().hasHeightForWidth());
369         scrollColorOn->setSizePolicy(sizePolicy);
370         scrollColorOn->setMinimumSize(QSize(0, 40));
371         scrollColorOn->setMaximumSize(QSize(16777215, 40));
372
373         gridLayout->addWidget(scrollColorOn, 23, 4, 1, 1);
374
375         capsBox = new QComboBox(KPerfWidget);
376         capsBox->setObjectName(QStringLiteral("capsBox"));
377
378         gridLayout->addWidget(capsBox, 22, 2, 1, 1);
379
380         scrollColorOff = new ColorButton(KPerfWidget);
381         scrollColorOff->setObjectName(QStringLiteral("scrollColorOff"));
382         sizePolicy.setHeightForWidth(scrollColorOff->sizePolicy().hasHeightForWidth());
383         scrollColorOff->setSizePolicy(sizePolicy);
384         scrollColorOff->setMinimumSize(QSize(0, 40));
385         scrollColorOff->setMaximumSize(QSize(16777215, 40));
386
387         gridLayout->addWidget(scrollColorOff, 23, 7, 1, 1);
388
389         muteBox = new QCheckBox(KPerfWidget);
390         muteBox->setObjectName(QStringLiteral("muteBox"));
391
392         gridLayout->addWidget(muteBox, 17, 1, 1, 2);
393
394         muteColorOn = new ColorButton(KPerfWidget);
395         muteColorOn->setObjectName(QStringLiteral("muteColorOn"));
396         sizePolicy.setHeightForWidth(muteColorOn->sizePolicy().hasHeightForWidth());
397         muteColorOn->setSizePolicy(sizePolicy);
398         muteColorOn->setMinimumSize(QSize(0, 40));
399         muteColorOn->setMaximumSize(QSize(16777215, 40));
400
401         gridLayout->addWidget(muteColorOn, 17, 4, 1, 1);
402
403         macroColorOff = new ColorButton(KPerfWidget);
404         macroColorOff->setObjectName(QStringLiteral("macroColorOff"));
405         QSizePolicy sizePolicy2(QSizePolicy::Minimum, QSizePolicy::Fixed);
406         sizePolicy2.setHorizontalStretch(0);
407         sizePolicy2.setVerticalStretch(40);
408         sizePolicy2.setHeightForWidth(macroColorOff->sizePolicy().hasHeightForWidth());
409         macroColorOff->setSizePolicy(sizePolicy2);
410         macroColorOff->setMinimumSize(QSize(0, 40));
411
412         gridLayout->addWidget(macroColorOff, 10, 7, 1, 1);
413
414         label_3 = new QLabel(KPerfWidget);
415         label_3->setObjectName(QStringLiteral("label_3"));
416         label_3->setFont(font1);
417
418         gridLayout->addWidget(label_3, 13, 0, 1, 3);
419
420         muteColorOff = new ColorButton(KPerfWidget);
421         muteColorOff->setObjectName(QStringLiteral("muteColorOff"));
422         sizePolicy.setHeightForWidth(muteColorOff->sizePolicy().hasHeightForWidth());
423         muteColorOff->setSizePolicy(sizePolicy);
424         muteColorOff->setMinimumSize(QSize(0, 40));
425         muteColorOff->setMaximumSize(QSize(16777215, 40));
426
427         gridLayout->addWidget(muteColorOff, 17, 7, 1, 1);
428
429         k95Label3 = new QLabel(KPerfWidget);
430         k95Label3->setObjectName(QStringLiteral("k95Label3"));
431
432         gridLayout->addWidget(k95Label3, 9, 8, 1, 1);
433
434         macroColorOn = new ColorButton(KPerfWidget);
435         macroColorOn->setObjectName(QStringLiteral("macroColorOn"));
436         sizePolicy2.setHeightForWidth(macroColorOn->sizePolicy().hasHeightForWidth());
```

```
437         macroColorOn->setSizePolicy(sizePolicy2);
438         macroColorOn->setMinimumSize(QSize(0, 40));
439 
440         gridLayout->addWidget(macroColorOn, 10, 4, 1, 1);
441 
442         muteColorNA = new ColorButton(KPerfWidget);
443         muteColorNA->setObjectName(QStringLiteral("muteColorNA"));
444         muteColorNA->setMinimumSize(QSize(0, 40));
445         muteColorNA->setMaximumSize(QSize(16777215, 40));
446 
447         gridLayout->addWidget(muteColorNA, 17, 10, 1, 1);
448 
449         label_19 = new QLabel(KPerfWidget);
450         label_19->setObjectName(QStringLiteral("label_19"));
451 
452         gridLayout->addWidget(label_19, 21, 5, 1, 1);
453 
454         k95Line = new QFrame(KPerfWidget);
455         k95Line->setObjectName(QStringLiteral("k95Line"));
456         k95Line->setFrameShape(QFrame::HLine);
457         k95Line->setFrameShadow(QFrame::Sunken);
458 
459         gridLayout->addWidget(k95Line, 8, 0, 1, 13);
460 
461         label_21 = new QLabel(KPerfWidget);
462         label_21->setObjectName(QStringLiteral("label_21"));
463 
464         gridLayout->addWidget(label_21, 23, 5, 1, 1);
465 
466         label_24 = new QLabel(KPerfWidget);
467         label_24->setObjectName(QStringLiteral("label_24"));
468 
469         gridLayout->addWidget(label_24, 23, 8, 1, 1);
470 
471         label_20 = new QLabel(KPerfWidget);
472         label_20->setObjectName(QStringLiteral("label_20"));
473 
474         gridLayout->addWidget(label_20, 22, 5, 1, 1);
475 
476         label_16 = new QLabel(KPerfWidget);
477         label_16->setObjectName(QStringLiteral("label_16"));
478 
479         gridLayout->addWidget(label_16, 15, 8, 1, 1);
480 
481         label_14 = new QLabel(KPerfWidget);
482         label_14->setObjectName(QStringLiteral("label_14"));
483 
484         gridLayout->addWidget(label_14, 17, 8, 1, 1);
485 
486         label_18 = new QLabel(KPerfWidget);
487         label_18->setObjectName(QStringLiteral("label_18"));
488 
489         gridLayout->addWidget(label_18, 15, 11, 1, 1);
490 
491         horizontalSpacer_2 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
492 
493         gridLayout->addItem(horizontalSpacer_2, 15, 0, 1, 1);
494 
495         label_15 = new QLabel(KPerfWidget);
496         label_15->setObjectName(QStringLiteral("label_15"));
497 
498         gridLayout->addWidget(label_15, 16, 8, 1, 1);
499 
500         k95Label4 = new QLabel(KPerfWidget);
501         k95Label4->setObjectName(QStringLiteral("k95Label4"));
502 
503         gridLayout->addWidget(k95Label4, 10, 5, 1, 1);
504 
505         label_17 = new QLabel(KPerfWidget);
506         label_17->setObjectName(QStringLiteral("label_17"));
507 
508         gridLayout->addWidget(label_17, 17, 11, 1, 1);
509 
510         label_11 = new QLabel(KPerfWidget);
511         label_11->setObjectName(QStringLiteral("label_11"));
512 
513         gridLayout->addWidget(label_11, 16, 5, 1, 1);
514 
515         label_23 = new QLabel(KPerfWidget);
516         label_23->setObjectName(QStringLiteral("label_23"));
517 
518         gridLayout->addWidget(label_23, 22, 8, 1, 1);
519 
520         k95Label1 = new QLabel(KPerfWidget);
521         k95Label1->setObjectName(QStringLiteral("k95Label1"));
522         k95Label1->setFont(font1);
```

```
523
524         gridLayout->addWidget(k95Label1, 7, 0, 1, 5);
525
526         horizontalLayout_2 = new QHBoxLayout();
527         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
528         copyButton = new QPushButton(KPerfWidget);
529         copyButton->setObjectName(QStringLiteral("copyButton"));
530
531         horizontalLayout_2->addWidget(copyButton);
532
533         horizontalSpacer_8 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
534
535         horizontalLayout_2->addItem(horizontalSpacer_8);
536
537
538         gridLayout->addLayout(horizontalLayout_2, 3, 10, 1, 3);
539
540         horizontalLayout = new QHBoxLayout();
541         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
542         label_2 = new QLabel(KPerfWidget);
543         label_2->setObjectName(QStringLiteral("label_2"));
544
545         horizontalLayout->addWidget(label_2);
546
547         horizontalSpacer_4 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
548
549         horizontalLayout->addItem(horizontalSpacer_4);
550
551         intensityBox = new QSpinBox(KPerfWidget);
552         intensityBox->setObjectName(QStringLiteral("intensityBox"));
553         intensityBox->setMaximum(100);
554         intensityBox->setValue(100);
555
556         horizontalLayout->addWidget(intensityBox);
557
558         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
559
560         horizontalLayout->addItem(horizontalSpacer_3);
561
562
563         gridLayout->addLayout(horizontalLayout, 3, 1, 1, 9);
564
565         k95Label5 = new QLabel(KPerfWidget);
566         k95Label5->setObjectName(QStringLiteral("k95Label5"));
567
568         gridLayout->addWidget(k95Label5, 10, 8, 1, 1);
569
570         line_2 = new QFrame(KPerfWidget);
571         line_2->setObjectName(QStringLiteral("line_2"));
572         line_2->setFrameShape(QFrame::HLine);
573         line_2->setFrameShadow(QFrame::Sunken);
574
575         gridLayout->addWidget(line_2, 14, 0, 1, 13);
576
577         k95Label6 = new QLabel(KPerfWidget);
578         k95Label6->setObjectName(QStringLiteral("k95Label6"));
579         k95Label6->setFont(font);
580
581         gridLayout->addWidget(k95Label6, 11, 1, 1, 10);
582
583         QWidget::setTabOrder(intensityBox, copyButton);
584         QWidget::setTabOrder(copyButton, modeBox);
585         QWidget::setTabOrder(modeBox, modeColorOn);
586         QWidget::setTabOrder(modeColorOn, modeColorOff);
587         QWidget::setTabOrder(modeColorOff, macroBox);
588         QWidget::setTabOrder(macroBox, macroColorOn);
589         QWidget::setTabOrder(macroColorOn, macroColorOff);
590         QWidget::setTabOrder(macroColorOff, lightBox);
591         QWidget::setTabOrder(lightBox, lightColor3);
592         QWidget::setTabOrder(lightColor3, lightColor2);
593         QWidget::setTabOrder(lightColor2, lightColor1);
594         QWidget::setTabOrder(lightColor1, lockBox);
595         QWidget::setTabOrder(lockBox, lockColorOn);
596         QWidget::setTabOrder(lockColorOn, lockColorOff);
597         QWidget::setTabOrder(lockColorOff, muteBox);
598         QWidget::setTabOrder(muteBox, muteColorOn);
599         QWidget::setTabOrder(muteColorOn, muteColorOff);
600         QWidget::setTabOrder(muteColorOff, muteColorNA);
601         QWidget::setTabOrder(muteColorNA, numBox);
602         QWidget::setTabOrder(numBox, numColorOn);
603         QWidget::setTabOrder(numColorOn, numColorOff);
604         QWidget::setTabOrder(numColorOff, capsBox);
605         QWidget::setTabOrder(capsBox, capsColorOn);
606         QWidget::setTabOrder(capsColorOn, capsColorOff);
```

```
607         QWidget::setTabOrder(capsColorOff, scrollBox);
608         QWidget::setTabOrder(scrollBox, scrollColorOn);
609         QWidget::setTabOrder(scrollColorOn, scrollColorOff);
610 
611         retranslateUi(KPerfWidget);
612 
613         QMetaObject::connectSlotsByName(KPerfWidget);
614     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.45.3 Field Documentation

12.45.3.1 QComboBox* Ui_KPerfWidget::capsBox [inherited]

Definition at line 74 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.2 ColorButton* Ui_KPerfWidget::capsColorOff [inherited]

Definition at line 66 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.3 ColorButton* Ui_KPerfWidget::capsColorOn [inherited]

Definition at line 68 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.4 QPushButton* Ui_KPerfWidget::copyButton [inherited]

Definition at line 100 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.5 QGridLayout* Ui_KPerfWidget::gridLayout [inherited]

Definition at line 34 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.6 QHBoxLayout* Ui_KPerfWidget::horizontalLayout [inherited]

Definition at line 102 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.7 QHBoxLayout* Ui_KPerfWidget::horizontalLayout_2 [inherited]

Definition at line 99 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.8 QSpacerItem* Ui_KPerfWidget::horizontalSpacer [inherited]

Definition at line 44 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.9 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_2 [inherited]

Definition at line 92 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.10 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_3 [inherited]

Definition at line 106 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.11 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_4 [inherited]

Definition at line 104 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.12 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_5 [inherited]

Definition at line 39 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.13 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_6 [inherited]

Definition at line 60 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.14 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_7 [inherited]

Definition at line 53 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.15 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_8 [inherited]

Definition at line 101 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.16 QSpinBox* Ui_KPerfWidget::intensityBox [inherited]

Definition at line 105 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.17 QLabel* Ui_KPerfWidget::k95Label1 [inherited]

Definition at line 98 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.18 QLabel* Ui_KPerfWidget::k95Label2 [inherited]

Definition at line 40 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.19 QLabel* Ui_KPerfWidget::k95Label3 [inherited]

Definition at line 81 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.20 QLabel* Ui_KPerfWidget::k95Label4 [inherited]

Definition at line 94 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.21 QLabel* Ui_KPerfWidget::k95Label5 [inherited]

Definition at line 107 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.22 QLabel* Ui_KPerfWidget::k95Label6 [inherited]

Definition at line 109 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.23 QFrame* Ui_KPerfWidget::k95Line [inherited]

Definition at line 85 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), and Ui_KPerfWidget::setupUi().

12.45.3.24 QWidget* Ui_KPerfWidget::k95Spacer [inherited]

Definition at line 35 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), and Ui_KPerfWidget::setupUi().

12.45.3.25 QLabel* Ui_KPerfWidget::label [inherited]

Definition at line 59 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.26 QLabel* Ui_KPerfWidget::label_10 [inherited]

Definition at line 36 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.27 QLabel* Ui_KPerfWidget::label_11 [inherited]

Definition at line 96 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.28 QLabel* Ui_KPerfWidget::label_12 [inherited]

Definition at line 41 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.29 QLabel* Ui_KPerfWidget::label_14 [inherited]

Definition at line 90 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.30 QLabel* Ui_KPerfWidget::label_15 [inherited]

Definition at line 93 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.31 QLabel* Ui_KPerfWidget::label_16 [inherited]

Definition at line 89 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.32 QLabel* Ui_KPerfWidget::label_17 [inherited]

Definition at line 95 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.33 QLabel* Ui_KPerfWidget::label_18 [inherited]

Definition at line 91 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.34 QLabel* Ui_KPerfWidget::label_19 [inherited]

Definition at line 84 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.35 QLabel* Ui_KPerfWidget::label_2 [inherited]

Definition at line 103 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.36 QLabel* Ui_KPerfWidget::label_20 [inherited]

Definition at line 88 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.37 QLabel* Ui_KPerfWidget::label_21 [inherited]

Definition at line 86 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.38 QLabel* Ui_KPerfWidget::label_22 [inherited]

Definition at line 70 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.39 QLabel* Ui_KPerfWidget::label_23 [inherited]

Definition at line 97 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.40 QLabel* Ui_KPerfWidget::label_24 [inherited]

Definition at line 87 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.41 QLabel* Ui_KPerfWidget::label_3 [inherited]

Definition at line 79 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.42 QLabel* Ui_KPerfWidget::label_4 [inherited]

Definition at line 62 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.43 QLabel* Ui_KPerfWidget::label_5 [inherited]

Definition at line 50 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.44 QLabel* Ui_KPerfWidget::label_6 [inherited]

Definition at line 58 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.45 QLabel* Ui_KPerfWidget::label_7 [inherited]

Definition at line 72 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.46 QLabel* Ui_KPerfWidget::label_8 [inherited]

Definition at line 54 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.47 QCheckBox* Ui_KPerfWidget::lightBox [inherited]

Definition at line 55 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.48 ColorButton* Ui_KPerfWidget::lightColor1 [inherited]

Definition at line 56 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.49 ColorButton* Ui_KPerfWidget::lightColor2 [inherited]

Definition at line 52 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.50 ColorButton* Ui_KPerfWidget::lightColor3 [inherited]

Definition at line 43 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.51 QFrame* Ui_KPerfWidget::line [inherited]

Definition at line 38 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.52 QFrame* Ui_KPerfWidget::line_2 [inherited]

Definition at line 108 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.53 QFrame* Ui_KPerfWidget::line_3 [inherited]

Definition at line 69 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.54 QCheckBox* Ui_KPerfWidget::lockBox [inherited]

Definition at line 45 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.55 ColorButton* Ui_KPerfWidget::lockColorOff [inherited]

Definition at line 37 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.56 ColorButton* Ui_KPerfWidget::lockColorOn [inherited]

Definition at line 61 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.57 QCheckBox* Ui_KPerfWidget::macroBox [inherited]

Definition at line 47 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.58 ColorButton* Ui_KPerfWidget::macroColorOff [inherited]

Definition at line 78 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.59 ColorButton* Ui_KPerfWidget::macroColorOn [inherited]

Definition at line 82 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.60 QCheckBox* Ui_KPerfWidget::modeBox [inherited]

Definition at line 65 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.61 ColorButton* Ui_KPerfWidget::modeColorOff [inherited]

Definition at line 42 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.62 ColorButton* Ui_KPerfWidget::modeColorOn [inherited]

Definition at line 46 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.63 QCheckBox* Ui_KPerfWidget::muteBox [inherited]

Definition at line 76 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.64 ColorButton* Ui_KPerfWidget::muteColorNA [inherited]

Definition at line 83 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.65 ColorButton* Ui_KPerfWidget::muteColorOff [inherited]

Definition at line 80 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.66 ColorButton* Ui_KPerfWidget::muteColorOn [inherited]

Definition at line 77 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), KPerfWidget::setPerf(), and Ui_KPerfWidget::setupUi().

12.45.3.67 QComboBox* Ui_KPerfWidget::numBox [inherited]

Definition at line 49 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.68 ColorButton* Ui_KPerfWidget::numColorOff [inherited]

Definition at line 67 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.69 ColorButton* Ui_KPerfWidget::numColorOn [inherited]

Definition at line 48 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.70 QComboBox* Ui_KPerfWidget::scrollBox [inherited]

Definition at line 71 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.71 ColorButton* Ui_KPerfWidget::scrollColorOff [inherited]

Definition at line 75 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.72 ColorButton* Ui_KPerfWidget::scrollColorOn [inherited]

Definition at line 73 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), Ui_KPerfWidget::retranslateUi(), and Ui_KPerfWidget::setupUi().

12.45.3.73 QSpacerItem* Ui_KPerfWidget::verticalSpacer [inherited]

Definition at line 63 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.74 QSpacerItem* Ui_KPerfWidget::verticalSpacer_2 [inherited]

Definition at line 51 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.75 QSpacerItem* Ui_KPerfWidget::verticalSpacer_3 [inherited]

Definition at line 64 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

12.45.3.76 QSpacerItem* Ui_KPerfWidget::verticalSpacer_4 [inherited]

Definition at line 57 of file ui_kperfwidget.h.

Referenced by Ui_KPerfWidget::setupUi().

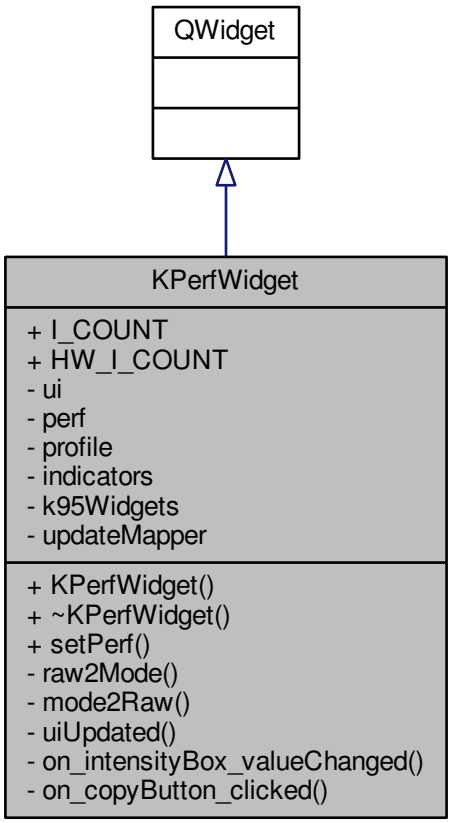
The documentation for this class was generated from the following file:

- src/ckb/ui_kperfwidget.h

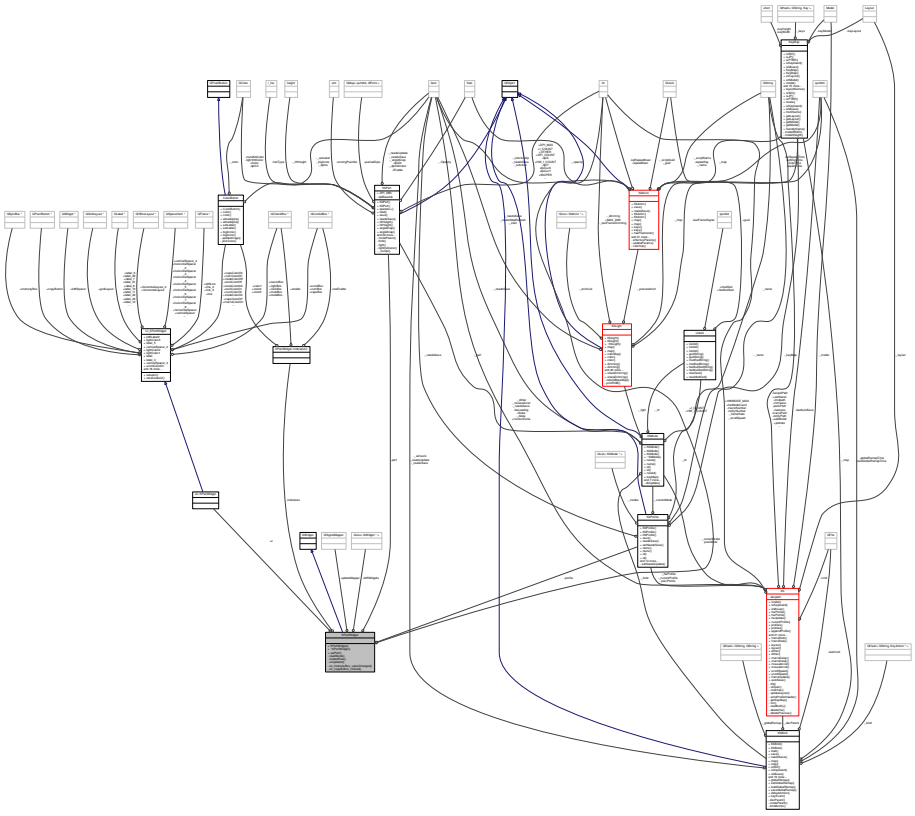
12.46 KPerfWidget Class Reference

`#include <src/ckb/kperfwidget.h>`

Inheritance diagram for KPerfWidget:



Collaboration diagram for KPerfWidget:



Data Structures

- struct IndicatorUi

Public Types

- typedef KbPerf::indicator indicator
- typedef KbPerf::i_hw i_hw

Public Member Functions

- KPerfWidget (QWidget *parent=0)

  KPerfWidget::KPerfWidget sets up the UI for Keyboard Performace panel.
- ~KPerfWidget ()

  KPerfWidget::~KPerfWidget nothing unusual - just delete the ui object.
- void setPerf (KbPerf *newPerf, KbProfile *newProfile)

Static Public Attributes

- static const int I_COUNT = KbPerf::I_COUNT
- static const int HW_I_COUNT = KbPerf::HW_I_COUNT

Private Types

- enum HwMode {
  NORMAL, ALWAYS_ON, ALWAYS_OFF, RGB,
  BOTH }

Private Slots

- void uiUpdated (int index)
- void on_intensityBox_valueChanged (int arg1)
- void on_copyButton_clicked ()

Private Member Functions

- HwMode raw2Mode (bool sw_enable, i_hw hw_enable)

  KPerfWidget::raw2Mode return hardware mode depending on setiings in sw_enable and hw_enable.
- void mode2Raw (HwMode mode, bool &sw_enable, i_hw &hw_enable)

  KPerfWidget::mode2Raw Set values of sw_enable and hw_enable to hte value corresponding to input var mode.

Private Attributes

- Ui::KPerfWidget * ui
- KbPerf * perf
- KbProfile * profile
- IndicatorUi indicators [I_COUNT]
- QList< QWidget * > k95Widgets
- QSignalMapper updateMapper

12.46.1 Detailed Description

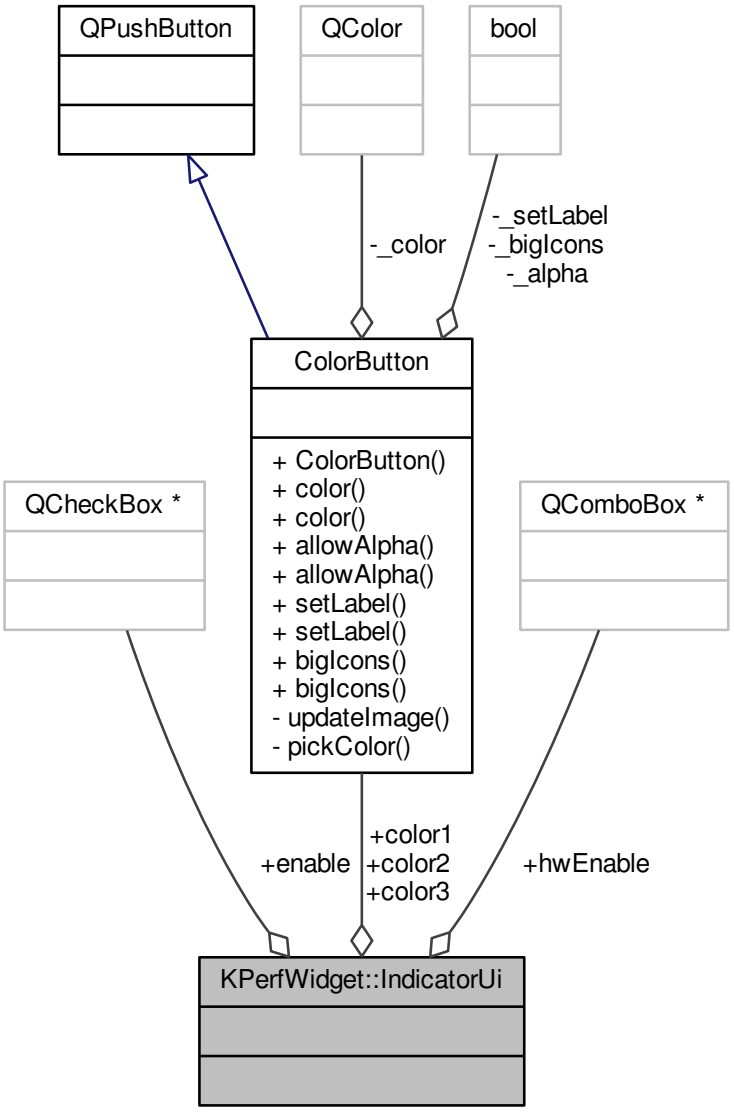
Definition at line 16 of file kperfwidget.h.

12.46.2 Data Structure Documentation

12.46.2.1 struct KPerfWidget::IndicatorUi

Definition at line 54 of file kperfwidget.h.

Collaboration diagram for KPerfWidget::IndicatorUi:



Data Fields

ColorButton *	color1	
ColorButton *	color2	
ColorButton *	color3	
QCheckBox *	enable	
QComboBox *	hwEnable	

12.46.3 Member Typedef Documentation

12.46.3.1 typedef KbPerf::i_hw KPerfWidget::i_hw

Definition at line 29 of file kperfwidget.h.

12.46.3.2 typedef KbPerf::indicator KPerfWidget::indicator

Definition at line 28 of file kperfwidget.h.

12.46.4 Member Enumeration Documentation

12.46.4.1 enum KPerfWidget::HwMode [private]

Enumerator

NORMAL

ALWAYS_ON

ALWAYS_OFF

RGB

BOTH

Definition at line 38 of file kperfwidget.h.

```
38                  {
39          NORMAL,
40          ALWAYS_ON,
41          ALWAYS_OFF,
42          RGB,
43          BOTH
44      };
```

12.46.5 Constructor & Destructor Documentation

12.46.5.1 KPerfWidget::KPerfWidget (QWidget * *parent* = 0) [explicit]

Parameters

parent	

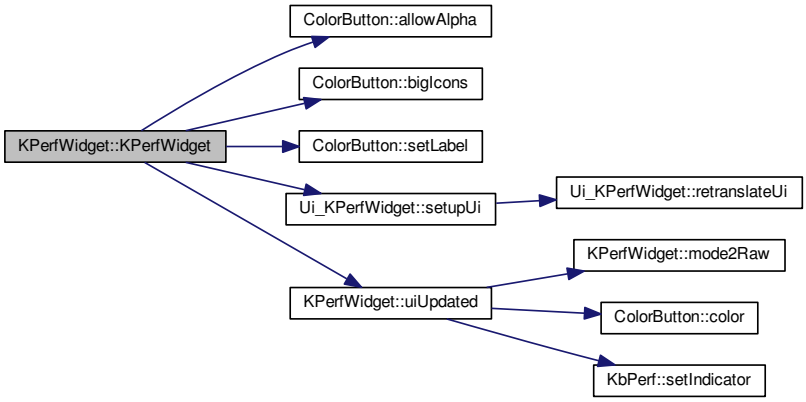
Definition at line 10 of file kperfwidget.cpp.

References ColorButton::allowAlpha(), ColorButton::bigIcons(), Ui_KPerfWidget::capsBox, Ui_KPerfWidget::capsColorOff, Ui_KPerfWidget::capsColorOn, KPerfWidget::IndicatorUi::color1, KPerfWidget::IndicatorUi::color2, KPerfWidget::IndicatorUi::color3, KPerfWidget::IndicatorUi::enable, KPerfWidget::IndicatorUi::hwEnable, I_COUNT, indicators, Ui_KPerfWidget::k95Label1, Ui_KPerfWidget::k95Label2, Ui_KPerfWidget::k95Label3, Ui_KPerfWidget::k95Label4, Ui_KPerfWidget::k95Label5, Ui_KPerfWidget::k95Label6, Ui_KPerfWidget::k95Line, Ui_KPerfWidget::k95Spacer, k95Widgets, Ui_KPerfWidget::lightBox, Ui_KPerfWidget::lightColor1, Ui_KPerfWidget::lightColor2, Ui_KPerfWidget::lightColor3, Ui_KPerfWidget::lockBox, Ui_KPerfWidget::lockColorOff, Ui_KPerfWidget::lockColorOn, Ui_KPerfWidget::macroBox, Ui_KPerfWidget::macroColorOff, Ui_KPerfWidget::macroColorOn, Ui_KPerfWidget::modeBox, Ui_KPerfWidget::modeColorOff, Ui_KPerfWidget::modeColorOn, Ui_KPerfWidget::muteBox, Ui_KPerfWidget::muteColorNA, Ui_KPerfWidget::muteColorOff, Ui_KPerfWidget::muteColorOn, Ui_KPerfWidget::numBox, Ui_KPerfWidget::numColorOff, Ui_KPerfWidget::numColorOn, Ui_KPerfWidget::scrollBox, Ui_KPerfWidget::scrollColorOff, Ui_KPerfWidget::scrollColorOn, ColorButton::setLabel(), Ui_KPerfWidget::setupUi(), ui, uiUpdated(), and updateMapper.

```
10                                      :
11      QWidget(parent),
12      ui(new Ui::KPerfWidget)
13  {
14      ui->setupUi(this);
15      // Set up indicators
```

```
16      indicators[0].enable = 0; indicators[0].hwEnable =
       ui->numBox; indicators[0].color1 = ui->numColorOn;
       indicators[0].color2 = ui->numColorOff; indicators[0].
       color3 = 0;
17      indicators[1].enable = 0; indicators[1].hwEnable =
       ui->capsBox; indicators[1].color1 = ui->capsColorOn;
       indicators[1].color2 = ui->capsColorOff;
       indicators[1].color3 = 0;
18      indicators[2].enable = 0; indicators[2].hwEnable =
       ui->scrollBox; indicators[2].color1 = ui->
       scrollColorOn; indicators[2].color2 = ui->
       scrollColorOff; indicators[2].color3 = 0;
19      indicators[3].enable = ui->modeBox; indicators[3].
       hwEnable = 0; indicators[3].color1 = ui->modeColorOn;
       indicators[3].color2 = ui->modeColorOff;
       indicators[3].color3 = 0;
20      indicators[4].enable = ui->macroBox; indicators[4].
       hwEnable = 0; indicators[4].color1 = ui->macroColorOn;
       indicators[4].color2 = ui->macroColorOff;
       indicators[4].color3 = 0;
21      indicators[5].enable = ui->lightBox; indicators[5].
       hwEnable = 0; indicators[5].color1 = ui->lightColor1;
       indicators[5].color2 = ui->lightColor2; indicators[5].
       color3 = ui->lightColor3;
22      indicators[6].enable = ui->lockBox; indicators[6].
       hwEnable = 0; indicators[6].color1 = ui->lockColorOn;
       indicators[6].color2 = ui->lockColorOff;
       indicators[6].color3 = 0;
23      indicators[7].enable = ui->muteBox; indicators[7].
       hwEnable = 0; indicators[7].color1 = ui->muteColorOn;
       indicators[7].color2 = ui->muteColorOff;
       indicators[7].color3 = ui->muteColorNA;
24      for(int i = 0; i < I_COUNT; i++){
25          indicators[i].color1->setLabel(false);
26          indicators[i].color1->bigIcons(true);
27          indicators[i].color1->allowAlpha(true);
28          indicators[i].color2->setLabel(false);
29          indicators[i].color2->bigIcons(true);
30          indicators[i].color2->allowAlpha(true);
31          if(indicators[i].color3){
32              indicators[i].color3->setLabel(false);
33              indicators[i].color3->bigIcons(true);
34              indicators[i].color3->allowAlpha(true);
35          }
36          // Map signals
37          if(indicators[i].enable){
38              connect(indicators[i].enable, SIGNAL(clicked(bool)), &
       updateMapper, SLOT(map()));
39              updateMapper.setMapping(indicators[i].enable, i);
40          }
41          if(indicators[i].hwEnable){
42              connect(indicators[i].hwEnable, SIGNAL(activated(int)), &
       updateMapper, SLOT(map()));
43              updateMapper.setMapping(indicators[i].hwEnable, i);
44          }
45          connect(indicators[i].color1, SIGNAL(colorChanged(QColor)), &
       updateMapper, SLOT(map()));
46          updateMapper.setMapping(indicators[i].color1, i);
47          connect(indicators[i].color2, SIGNAL(colorChanged(QColor)), &
       updateMapper, SLOT(map()));
48          updateMapper.setMapping(indicators[i].color2, i);
49          if(indicators[i].color3){
50              connect(indicators[i].color3, SIGNAL(colorChanged(QColor)), &
       updateMapper, SLOT(map()));
51              updateMapper.setMapping(indicators[i].color3, i);
52          }
53      }
54      connect(&updateMapper, SIGNAL(mapped(int)), this, SLOT(uiUpdated(int)));
55      k95Widgets << ui->modeBox << ui->modeColorOn <<
       ui->modeColorOff << ui->macroBox << ui->macroColorOn <<
       ui->macroColorOff << ui->k95Label1 << ui->k95Label2 <<
       ui->k95Label3 << ui->k95Label4 << ui->k95Label5 <<
       ui->k95Label6 << ui->k95Line << ui->k95Spacer;
56  }
```

Here is the call graph for this function:



12.46.5.2 KPerfWidget::~KPerfWidget ( )

Definition at line 61 of file kperfwidget.cpp.

References ui.

```
61                              {
62     delete ui;
63 }
```

12.46.6 Member Function Documentation

12.46.6.1 void KPerfWidget::mode2Raw (HwMode *mode*, bool & *sw_enable*, i_hw & *hw_enable*) [private]

Parameters

[IN]	mode
[OUT]	sw_enable
[OUT]	hw_enable
[IN]	mode
[OUT]	sw_enable
[OUT]	hw_enable mode determines how colors at a key and separate indicators should be handled: Mode | sw | hw _ | _ | _ NORMAL | No color change at the key | use the separate indicator depending on key state ALWAYS_ON | No color change at the key | switch on separate indicator ALWAYS_OFF | No color change at the key | switch off separate indicator RGB | use color change at the key depending on color sliders | switch off separate indicator BOTH | use color change at the key depending on color sliders | use the separate indicator depending on key state

Definition at line 100 of file kperfwidget.cpp.

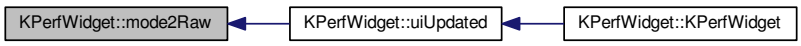
References ALWAYS_OFF, ALWAYS_ON, BOTH, NORMAL, KbPerf::NORMAL, KbPerf::OFF, KbPerf::ON, and RGB.

Referenced by uiUpdated().

```
100                                                                 {
101     switch(mode){
102     case NORMAL:
```

```
103         sw_enable = false;
104         hw_enable = KbPerf::NORMAL;
105         break;
106     case ALWAYS_ON:
107         sw_enable = false;
108         hw_enable = KbPerf::ON;
109         break;
110     case ALWAYS_OFF:
111         sw_enable = false;
112         hw_enable = KbPerf::OFF;
113         break;
114     case RGB:
115         sw_enable = true;
116         hw_enable = KbPerf::OFF;
117         break;
118     case BOTH:
119         sw_enable = true;
120         hw_enable = KbPerf::NORMAL;
121         break;
122     }
123 }
```

Here is the caller graph for this function:



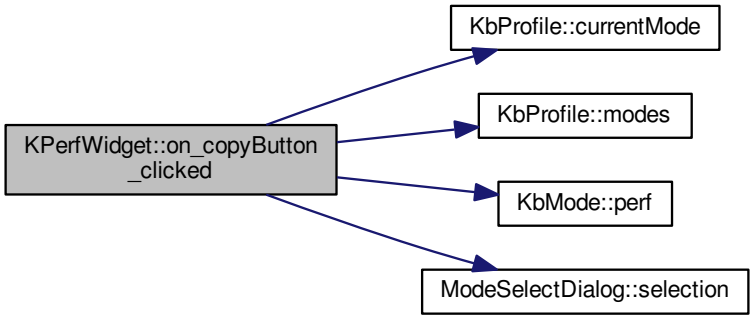
12.46.6.2 void KPerfWidget::on_copyButton_clicked ( ) [private],[slot]

Definition at line 221 of file kperfwidget.cpp.

References KbProfile::currentMode(), KbProfile::modes(), perf, KbMode::perf(), profile, and ModeSelectDialog::selection().

```
221                                     {
222     ModeSelectDialog dialog(this, profile->currentMode(),
      profile->modes(), "Copy performance settings to:");
223     if(dialog.exec() != QDialog::Accepted)
224         return;
225     QList<KbMode*> selectedModes = dialog.selection();
226     foreach(KbMode* mode, selectedModes){
227         *mode->perf() = *perf;
228     }
229 }
```

Here is the call graph for this function:



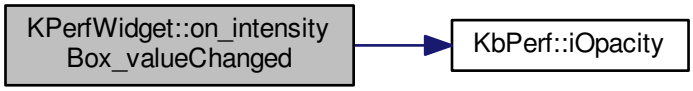
12.46.6.3 void KPerfWidget::on_intensityBox_valueChanged (int *arg1*) [private], [slot]

Definition at line 215 of file kperfwidget.cpp.

References KbPerf::iOpacity(), and perf.

```
215                                                     {
216     if(!perf)
217         return;
218     perf->iOpacity(arg1 / 100.f);
219 }
```

Here is the call graph for this function:



12.46.6.4 KPerfWidget::HwMode KPerfWidget::raw2Mode (bool *sw_enable,* i_hw *hw_enable*) [private]

Parameters

sw_enable	
hw_enable	

Returns

the mode of operation for key-coloring and separate indicators.

See Also

KPerfWidget::mode2Raw for details.

Definition at line 72 of file kperfwidget.cpp.

References ALWAYS_OFF, ALWAYS_ON, BOTH, NORMAL, KbPerf::NORMAL, KbPerf::OFF, KbPerf::ON, and RGB.

Referenced by setPerf().

```
72                                                                  {
73     if(sw_enable){
74         if(hw_enable == KbPerf::NORMAL)
75             return BOTH;
76         return RGB;
77     } else {
78         if(hw_enable == KbPerf::ON)
79             return ALWAYS_ON;
80         else if(hw_enable == KbPerf::OFF)
81             return ALWAYS_OFF;
82         return NORMAL;
83     }
84 }
```

Here is the caller graph for this function:

12.46.6.5 void KPerfWidget::setPerf (KbPerf * *newPerf,* KbProfile * *newProfile*)

Definition at line 125 of file kperfwidget.cpp.

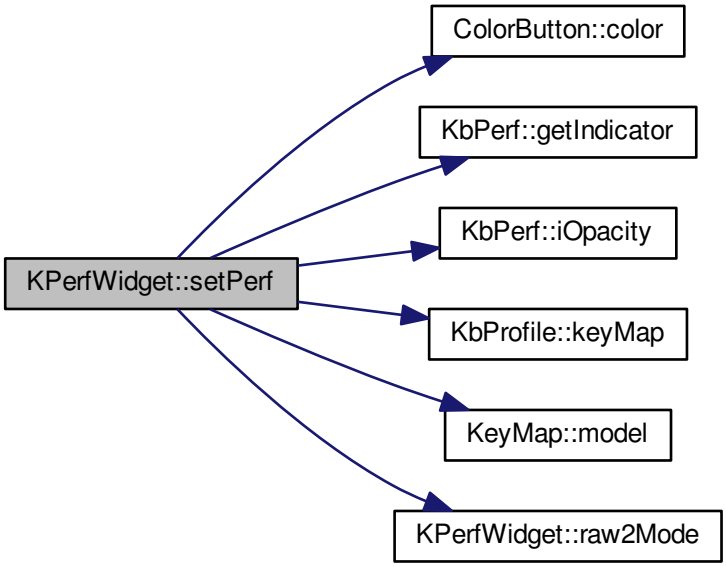
References BOTH, ColorButton::color(), KPerfWidget::IndicatorUi::color1, KPerfWidget::IndicatorUi::color2, KPerfWidget::IndicatorUi::color3, KPerfWidget::IndicatorUi::enable, KbPerf::getIndicator(), HW_I_COUNT, KPerfWidget::IndicatorUi::hwEnable, I_COUNT, indicators, Ui_KPerfWidget::intensityBox, KbPerf::iOpacity(), KeyMap::K95, k95Widgets, KbProfile::keyMap(), Ui_KPerfWidget::label_12, Ui_KPerfWidget::label_14, Ui_KPerfWidget::label_17, KeyMap::model(), Ui_KPerfWidget::muteBox, Ui_KPerfWidget::muteColorNA, Ui_KPerfWidget::muteColorOff, Ui_KPerfWidget::muteColorOn, perf, profile, raw2Mode(), RGB, KeyMap::STRAFE, and ui.

Referenced by KbWidget::modeChanged().

```
125                                                     {
126     perf = newPerf;
127     profile = newProfile;
128     // Set intensity
129     ui->intensityBox->setValue(round(perf->iOpacity() * 100.f));
130     // Set hardware indicator values
131     for(int i = 0; i < HW_I_COUNT; i++){
132         QColor c1, c2, c3;
133         bool software;
134         i_hw hardware;
135         perf->getIndicator((indicator)i, c1, c2, c3, software, hardware);
136         HwMode mode = raw2Mode(software, hardware);
137         indicators[i].hwEnable->setCurrentIndex((int)mode);
138         indicators[i].color1->color(c1);
139         indicators[i].color2->color(c2);
140         if(mode == RGB || mode == BOTH){
141             indicators[i].color1->setEnabled(true);
142             indicators[i].color2->setEnabled(true);
143         } else {
144             indicators[i].color1->setEnabled(false);
145             indicators[i].color2->setEnabled(false);
146         }
147     }
148     // Set software indicators
149     for(int i = HW_I_COUNT; i < I_COUNT; i++){
150         QColor c1, c2, c3;
151         bool software;
152         i_hw hardware;
153         perf->getIndicator((indicator)i, c1, c2, c3, software, hardware);
154         indicators[i].enable->setChecked(software);
155         indicators[i].color1->color(c1);
156         indicators[i].color2->color(c2);
157         if(indicators[i].color3) indicators[i].color3->
    color(c3);
158         if(software){
159             indicators[i].color1->setEnabled(true);
160             indicators[i].color2->setEnabled(true);
161             if(indicators[i].color3) indicators[i].color3->setEnabled(true);
162         } else {
163             indicators[i].color1->setEnabled(false);
164             indicators[i].color2->setEnabled(false);
165             if(indicators[i].color3) indicators[i].color3->setEnabled(false);
166         }
167     }
168     // Hide K95 indicators on non-K95s
169     if(profile->keyMap().model() == KeyMap::K95){
170         foreach(QWidget* w, k95Widgets)
171             w->show();
172     } else {
173         foreach(QWidget* w, k95Widgets)
174             w->hide();
175         if(profile->keyMap().model() == KeyMap::STRAFE) { // hide
```

```
        indicators unsupported by STRAFE
176             ui->muteBox->hide();
177             ui->muteColorOn->hide();
178             ui->muteColorOff->hide();
179             ui->muteColorNA->hide();
180             ui->label_12->hide();
181             ui->label_14->hide();
182             ui->label_17->hide();
183         }
184     }
185 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.46.6.6 void KPerfWidget::uiUpdated (int *index*) [private],[slot]

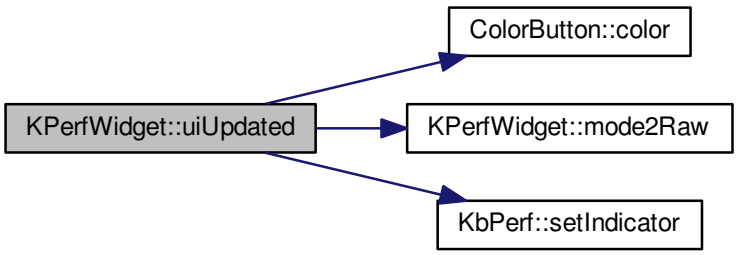
Definition at line 187 of file kperfwidget.cpp.

References ColorButton::color(), KPerfWidget::IndicatorUi::color1, KPerfWidget::IndicatorUi::color2, KPerfWidget::IndicatorUi::color3, KPerfWidget::IndicatorUi::enable, indicators, mode2Raw(), KbPerf::NONE, perf, and KbPerf::setIndicator().

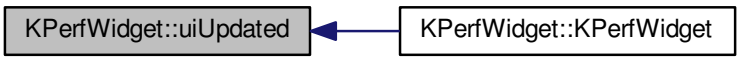
Referenced by KPerfWidget().

```
187                                     {
188     if(!perf)
189         return;
190     // Read HW/SW enable state
191     bool software;
192     i_hw hardware;
193     if(indicators[index].enable){
194         software = indicators[index].enable->isChecked();
195         hardware = KbPerf::NONE;
196     } else {
197         mode2Raw((HwMode)indicators[index].hwEnable->currentIndex(), software,
      hardware);
198     }
199     // Enable/disable color buttons based on whether or not the indicator is turned on
200     if(software){
201         indicators[index].color1->setEnabled(true);
202         indicators[index].color2->setEnabled(true);
203         if(indicators[index].color3) indicators[index].
      color3->setEnabled(true);
204     } else {
205         indicators[index].color1->setEnabled(false);
206         indicators[index].color2->setEnabled(false);
207         if(indicators[index].color3) indicators[index].
      color3->setEnabled(false);
208     }
209     QColor c1 = indicators[index].color1->color();
210     QColor c2 = indicators[index].color2->color();
211     QColor c3 = indicators[index].color3 ? indicators[index].
      color3->color() : QColor();
212     perf->setIndicator((indicator)index, c1, c2, c3, software, hardware);
213 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.46.7 Field Documentation

12.46.7.1 const int KPerfWidget::HW_I_COUNT = KbPerf::HW_I_COUNT [static]

Definition at line 27 of file kperfwidget.h.

Referenced by setPerf().

12.46.7.2 const int KPerfWidget::I_COUNT = KbPerf::I_COUNT [static]

Definition at line 26 of file kperfwidget.h.

Referenced by KPerfWidget(), and setPerf().

12.46.7.3 IndicatorUi KPerfWidget::indicators[I_COUNT] [private]

Definition at line 59 of file kperfwidget.h.

Referenced by KPerfWidget(), setPerf(), and uiUpdated().

12.46.7.4 QList<QWidget*> KPerfWidget::k95Widgets [private]

Definition at line 60 of file kperfwidget.h.

Referenced by KPerfWidget(), and setPerf().

12.46.7.5 KbPerf* KPerfWidget::perf [private]

Definition at line 34 of file kperfwidget.h.

Referenced by on_copyButton_clicked(), on_intensityBox_valueChanged(), setPerf(), and uiUpdated().

12.46.7.6 KbProfile* KPerfWidget::profile [private]

Definition at line 35 of file kperfwidget.h.

Referenced by on_copyButton_clicked(), and setPerf().

12.46.7.7 Ui::KPerfWidget* KPerfWidget::ui [private]

Definition at line 32 of file kperfwidget.h.

Referenced by KPerfWidget(), setPerf(), and ~KPerfWidget().

12.46.7.8 QSignalMapper KPerfWidget::updateMapper [private]

Definition at line 62 of file kperfwidget.h.

Referenced by KPerfWidget().

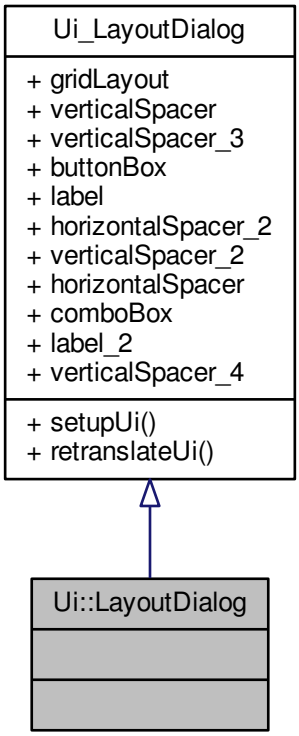
The documentation for this class was generated from the following files:

- src/ckb/kperfwidget.h
- src/ckb/kperfwidget.cpp

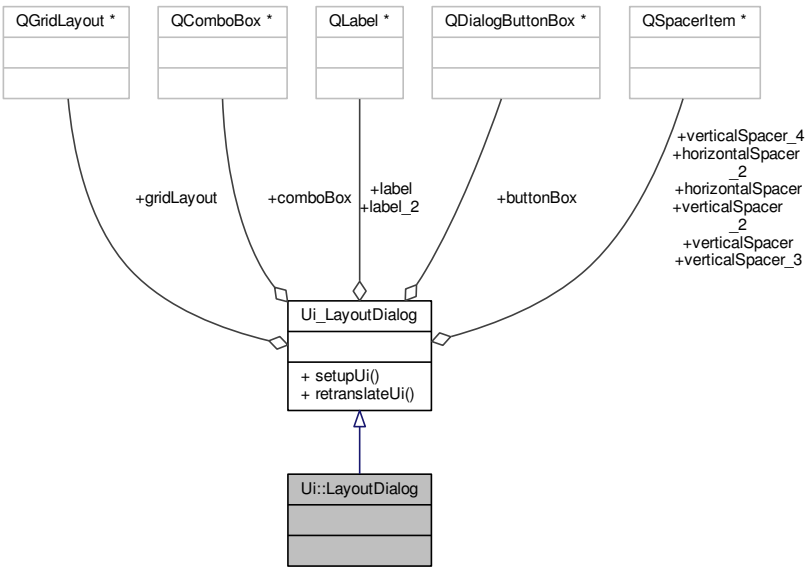
12.47 Ui::LayoutDialog Class Reference

`#include <src/ckb/ui_layoutdialog.h>`

Inheritance diagram for Ui::LayoutDialog:



Collaboration diagram for Ui::LayoutDialog:



Public Member Functions

- void setupUi (QDialog *LayoutDialog)
- void retranslateUi (QDialog *LayoutDialog)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_3
- QDialogButtonBox * buttonBox
- QLabel * label
- QSpacerItem * horizontalSpacer_2
- QSpacerItem * verticalSpacer_2
- QSpacerItem * horizontalSpacer
- QComboBox * comboBox
- QLabel * label_2
- QSpacerItem * verticalSpacer_4

12.47.1 Detailed Description

Definition at line 119 of file ui_layoutdialog.h.

12.47.2 Member Function Documentation

12.47.2.1 void Ui_LayoutDialog::retranslateUi (QDialog * *LayoutDialog*) [inline], [inherited]

Definition at line 109 of file ui_layoutdialog.h.

References Ui_LayoutDialog::label, and Ui_LayoutDialog::label_2.

Referenced by Ui_LayoutDialog::setupUi().

```
110     {
111         LayoutDialog->setWindowTitle(QApplication::translate("LayoutDialog", "Keyboard Layout", Q_NULLPTR))
    ;
112         label->setText(QApplication::translate("LayoutDialog", "Please select your keyboard layout:",
    Q_NULLPTR));
113         label_2->setText(QApplication::translate("LayoutDialog", "Note: this only controls the
     layout within the ckb application. If your keys are typing the wrong characters, check your system preferences.",
     Q_NULLPTR));
114     } // retranslateUi
```

Here is the caller graph for this function:

12.47.2.2 void Ui_LayoutDialog::setupUi (QDialog * *LayoutDialog*) [inline],[inherited]

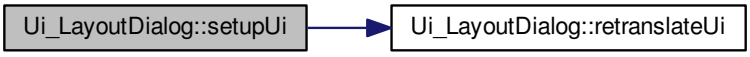
Definition at line 41 of file ui_layoutdialog.h.

References Ui_LayoutDialog::buttonBox, Ui_LayoutDialog::comboBox, Ui_LayoutDialog::gridLayout, Ui_LayoutDialog::horizontalSpacer, Ui_LayoutDialog::horizontalSpacer_2, Ui_LayoutDialog::label, Ui_LayoutDialog::label_2, Ui_LayoutDialog::retranslateUi(), Ui_LayoutDialog::verticalSpacer, Ui_LayoutDialog::verticalSpacer_2, Ui_LayoutDialog::verticalSpacer_3, and Ui_LayoutDialog::verticalSpacer_4.

Referenced by LayoutDialog::LayoutDialog().

```
42     {
43         if (LayoutDialog->objectName().isEmpty())
44             LayoutDialog->setObjectName(QStringLiteral("LayoutDialog"));
45         LayoutDialog->resize(400, 240);
46         gridLayout = new QGridLayout(LayoutDialog);
47         gridLayout->setObjectName(QStringLiteral("gridLayout"));
48         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
49 
50         gridLayout->addItem(verticalSpacer, 6, 0, 1, 3);
51 
52         verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
53 
54         gridLayout->addItem(verticalSpacer_3, 0, 1, 1, 1);
55 
56         buttonBox = new QDialogButtonBox(LayoutDialog);
57         buttonBox->setObjectName(QStringLiteral("buttonBox"));
58         buttonBox->setOrientation(Qt::Horizontal);
59         buttonBox->setStandardButtons(QDialogButtonBox::Ok);
60 
61         gridLayout->addWidget(buttonBox, 7, 0, 1, 3);
62 
63         label = new QLabel(LayoutDialog);
64         label->setObjectName(QStringLiteral("label"));
65         label->setAlignment(Qt::AlignCenter);
66 
67         gridLayout->addWidget(label, 1, 0, 1, 3);
68 
69         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
70 
71         gridLayout->addItem(horizontalSpacer_2, 3, 2, 1, 1);
72 
73         verticalSpacer_2 = new QSpacerItem(20, 15, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
74 
75         gridLayout->addItem(verticalSpacer_2, 4, 1, 1, 1);
76 
77         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
78 
79         gridLayout->addItem(horizontalSpacer, 3, 0, 1, 1);
80 
81         comboBox = new QComboBox(LayoutDialog);
82         comboBox->setObjectName(QStringLiteral("comboBox"));
83 
84         gridLayout->addWidget(comboBox, 3, 1, 1, 1);
85 
86         label_2 = new QLabel(LayoutDialog);
87         label_2->setObjectName(QStringLiteral("label_2"));
88         QFont font;
89         font.setPointSize(8);
90         font.setItalic(true);
91         label_2->setFont(font);
92         label_2->setAlignment(Qt::AlignCenter);
93         label_2->setWordWrap(true);
94 
95         gridLayout->addWidget(label_2, 5, 0, 1, 3);
96 
97         verticalSpacer_4 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
98 
99         gridLayout->addItem(verticalSpacer_4, 2, 1, 1, 1);
100 
101 
102         retranslateUi(LayoutDialog);
103         QObject::connect(buttonBox, SIGNAL(accepted()), LayoutDialog, SLOT(accept()));
104         QObject::connect(buttonBox, SIGNAL(rejected()), LayoutDialog, SLOT(reject()));
105 
106         QMetaObject::connectSlotsByName(LayoutDialog);
107     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.47.3 Field Documentation

12.47.3.1 QDialogButtonBox* Ui_LayoutDialog::buttonBox [inherited]

Definition at line 32 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.2 QComboBox* Ui_LayoutDialog::comboBox [inherited]

Definition at line 37 of file ui_layoutdialog.h.

Referenced by LayoutDialog::LayoutDialog(), LayoutDialog::selected(), and Ui_LayoutDialog::setupUi().

12.47.3.3 QGridLayout* Ui_LayoutDialog::gridLayout [inherited]

Definition at line 29 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.4 QSpacerItem* Ui_LayoutDialog::horizontalSpacer [inherited]

Definition at line 36 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.5 QSpacerItem* Ui_LayoutDialog::horizontalSpacer_2 [inherited]

Definition at line 34 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.6 QLabel* Ui_LayoutDialog::label [inherited]

Definition at line 33 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::retranslateUi(), and Ui_LayoutDialog::setupUi().

12.47.3.7 QLabel* Ui_LayoutDialog::label_2 [inherited]

Definition at line 38 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::retranslateUi(), and Ui_LayoutDialog::setupUi().

12.47.3.8 QSpacerItem* Ui_LayoutDialog::verticalSpacer [inherited]

Definition at line 30 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.9 QSpacerItem* Ui_LayoutDialog::verticalSpacer_2 [inherited]

Definition at line 35 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.10 QSpacerItem* Ui_LayoutDialog::verticalSpacer_3 [inherited]

Definition at line 31 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

12.47.3.11 QSpacerItem* Ui_LayoutDialog::verticalSpacer_4 [inherited]

Definition at line 39 of file ui_layoutdialog.h.

Referenced by Ui_LayoutDialog::setupUi().

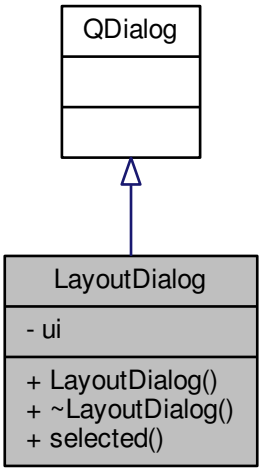
The documentation for this class was generated from the following file:

- src/ckb/ui_layoutdialog.h

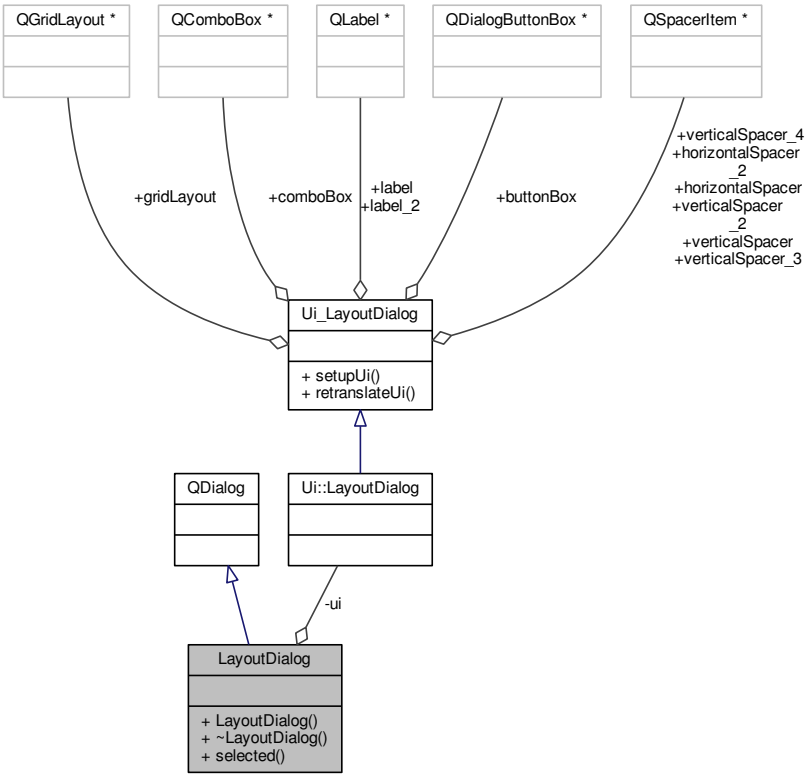
12.48 LayoutDialog Class Reference

`#include <src/ckb/layoutdialog.h>`

Inheritance diagram for LayoutDialog:



Collaboration diagram for LayoutDialog:



Public Member Functions

- LayoutDialog (QWidget *parent=0)
- ~LayoutDialog ()
- KeyMap::Layout selected () const

Private Attributes

- Ui::LayoutDialog * ui

12.48.1 Detailed Description

Definition at line 13 of file layoutdialog.h.

12.48.2 Constructor & Destructor Documentation

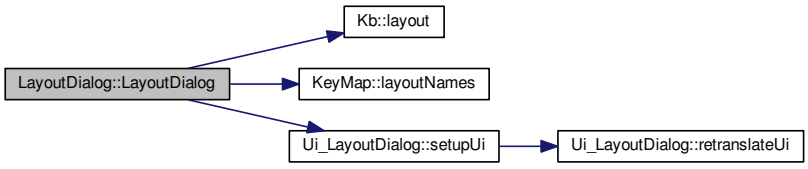
12.48.2.1 LayoutDialog::LayoutDialog (QWidget * *parent* = 0) [explicit]

Definition at line 5 of file layoutdialog.cpp.

References Ui_LayoutDialog::comboBox, Kb::layout(), KeyMap::layoutNames(), Ui_LayoutDialog::setupUi(), and ui.

```
5                                          :
6      QDialog(parent),
7      ui(new Ui::LayoutDialog) {
8      ui->setupUi(this);
9      // Populate layout list
10      ui->comboBox->addItems(KeyMap::layoutNames());
11      ui->comboBox->setCurrentIndex((int)Kb::layout());
12  }
```

Here is the call graph for this function:



12.48.2.2 LayoutDialog::~LayoutDialog ( )

Definition at line 14 of file layoutdialog.cpp.

References ui.

```
14                              {
15      delete ui;
16  }
```

12.48.3 Member Function Documentation

12.48.3.1 KeyMap::Layout LayoutDialog::selected ( ) const

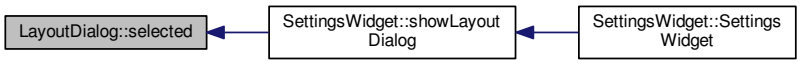
Definition at line 18 of file layoutdialog.cpp.

References Ui_LayoutDialog::comboBox, and ui.

Referenced by SettingsWidget::showLayoutDialog().

```
18                                          {
19     return (KeyMap::Layout)ui->comboBox->currentIndex();
20 }
```

Here is the caller graph for this function:



12.48.4 Field Documentation

12.48.4.1 Ui::LayoutDialog* LayoutDialog::ui [private]

Definition at line 24 of file layoutdialog.h.

Referenced by LayoutDialog(), selected(), and ~LayoutDialog().

The documentation for this class was generated from the following files:

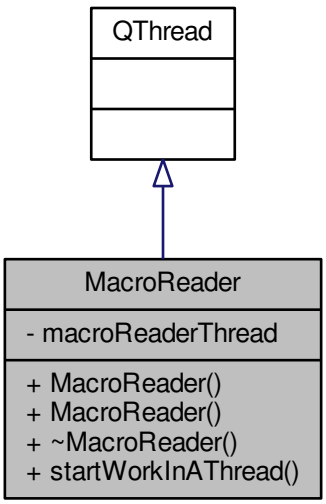
- src/ckb/layoutdialog.h
- src/ckb/layoutdialog.cpp

12.49 MacroReader Class Reference

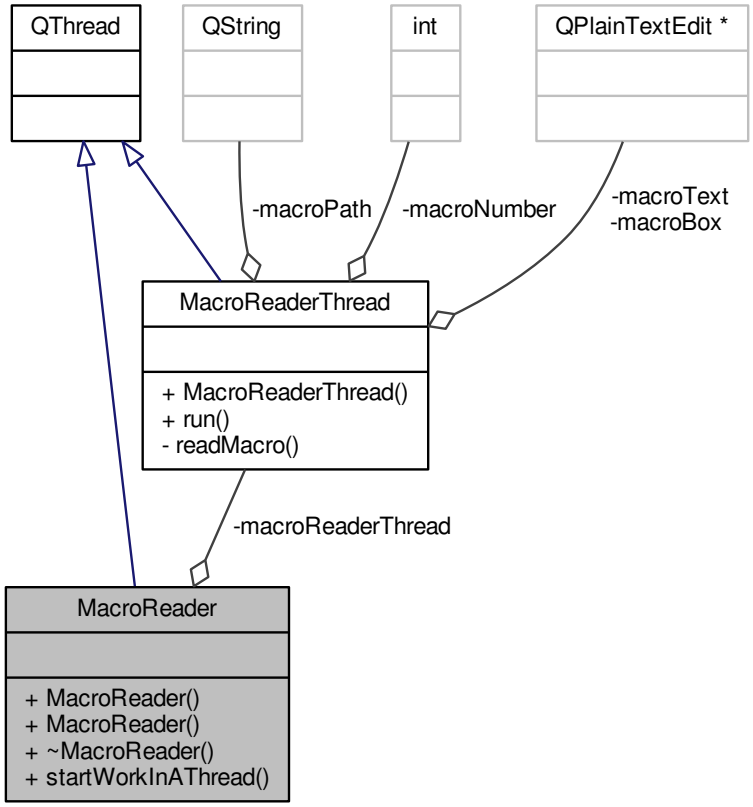
The MacroReader class creates a worker thread object. It does a connect do delayed deletion of thread local variables in the case the worker thread terminates.

`#include <src/ckb/macroreader.h>`

Inheritance diagram for MacroReader:



Collaboration diagram for MacroReader:



Public Member Functions

- MacroReader ()

  MacroReader Calling MacroReader without params is not allowed.
- MacroReader (int macroNumber, QString macroPath, QPlainTextEdit ∗macBox, QPlainTextEdit ∗macText)

  MacroReader This is the only allowed constructor. It only calls startWorkInAThread() with the four params.
- ∼MacroReader ()
- void startWorkInAThread (int macroNumber, QString macroPath, QPlainTextEdit ∗macBox, QPlainTextEdit ∗macText)

  startWorkInAThread This member function creates the new thread object and starts it with the four params.

Private Attributes

- MacroReaderThread ∗ macroReaderThread

12.49.1 Detailed Description

Definition at line 87 of file macroreader.h.

12.49.2 Constructor & Destructor Documentation

12.49.2.1 MacroReader::MacroReader ( )

Definition at line 9 of file macroreader.cpp.

```
9                           {
10     qDebug() << "Calling MacroReader without params is not allowed.";
11 }
```

12.49.2.2 MacroReader::MacroReader (int *macroNumber*, QString *macroPath*, QPlainTextEdit * *macBox*, QPlainTextEdit * *macText*)

Parameters

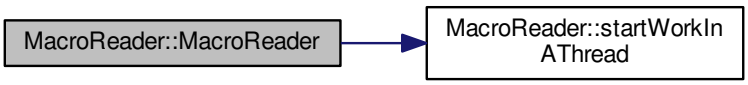
macroNumber	
macroPath	
macBox	
macText	

Definition at line 13 of file macroreader.cpp.

References startWorkInAThread().

```
13
       {
14     startWorkInAThread(macroNumber, macroPath, macBox, macText);
15 }
```

Here is the call graph for this function:



12.49.2.3 MacroReader::~MacroReader ( )

Definition at line 16 of file macroreader.cpp.

```
16 {}
```

12.49.3 Member Function Documentation

12.49.3.1 void MacroReader::startWorkInAThread (int *macroNumber*, QString *macroPath*, QPlainTextEdit * *macBox*, QPlainTextEdit * *macText*)

Parameters

macroNumber	
macroPath	
macBox	
macText	

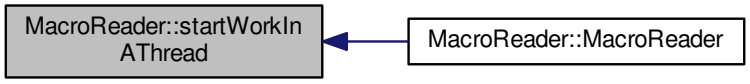
Definition at line 18 of file macroreader.cpp.

References macroReaderThread.

Referenced by MacroReader().

```
18
                {
19     macroReaderThread = new MacroReaderThread(macroNumber, macroPath,
       macBox, macText);
20     connect(macroReaderThread, &MacroReaderThread::finished,
       macroReaderThread, &QObject::deleteLater);
21     macroReaderThread->start();
22 }
```

Here is the caller graph for this function:



12.49.4 Field Documentation

12.49.4.1 MacroReaderThread* MacroReader::macroReaderThread [private]

Definition at line 123 of file macroreader.h.

Referenced by startWorkInAThread().

The documentation for this class was generated from the following files:

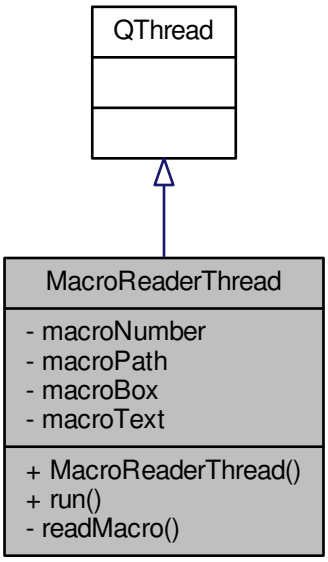
- src/ckb/macroreader.h
- src/ckb/macroreader.cpp

12.50 MacroReaderThread Class Reference

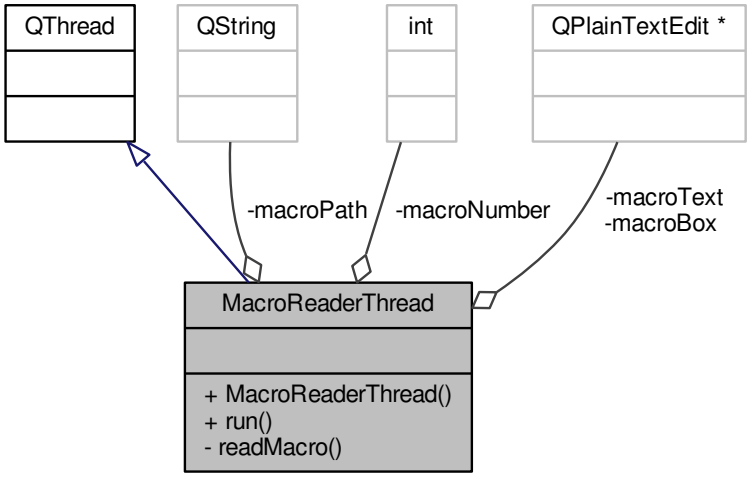
The MacroReaderThread class is responsible for reading Macro Key Values. It is created as a separate thread (worker thread) for reading macro commands from an fresh opened notify channel. Standard notify channel for macro definitions is number 2.

`#include <src/ckb/macroreader.h>`

Inheritance diagram for MacroReaderThread:



Collaboration diagram for MacroReaderThread:



Public Member Functions

- MacroReaderThread (int macNum, QString macPath, QPlainTextEdit ∗macBox, QPlainTextEdit ∗macText)

  MacroReaderThread saves the four params to local vars with similar varNames.

- void run () Q_DECL_OVERRIDE

  run is the notification reader main loop.

Private Slots

- void readMacro (QString line)

  readMacro is called for each line received by the worker thread. The method ist called via signal (metaObject) from the worker thread, which reads the keyboard input. Just display the key code in the macroBox Widget without he trailing newline and reposition the cursor in the macro pane.

Private Attributes

- int macroNumber

  macroNumber Filenames of nofity channels have the structure <input-device-path>/ckb1/notify<number> First part is hold in macroPath, the number is hold in macroNumber. macroNumber may range from 0 to 9.
- QString macroPath

  macroPath holds the path for the notify channel
- QPlainTextEdit * macroBox

  macroBox will receive the Macro Key Values sent from the keyboard while defining a new macro.
- QPlainTextEdit * macroText

  macroText is the other textpane used in the UI while typing new macros. That variable is used for setting the focus to that textpane and to set the cursor at EOT.

12.50.1 Detailed Description

While the worker Thread gets input from the keyboard, the lines are sent via signalling (metaobject) to run a member function in the context of the Qt UI manager.

When the notify channel is closed (that's normally done by pressing "Stop"-Button in the UI), the worker thread closes the channelFile and leaves.

See Also

MacroReaderThread(), ~MacroReaderThread(), readMaco(), run()

Definition at line 22 of file macroreader.h.

12.50.2 Constructor & Destructor Documentation

12.50.2.1 MacroReaderThread::MacroReaderThread (int *macNum*, QString *macPath*, QPlainTextEdit * *macBox*, QPlainTextEdit * *macText*) [inline]

Parameters

macNum	
macPath	
macBox	
macText	

Definition at line 55 of file macroreader.h.

References macroBox, macroNumber, macroPath, and macroText.

```
55                                                                                  {
56          macroNumber = macNum;
57          macroPath = macPath;
58          macroBox = macBox;
59          macroText = macText;
60      }
```

12.50.3 Member Function Documentation

12.50.3.1 void MacroReaderThread::readMacro (QString *line*) [private],[slot]

This is used, because the worker thread shouldn't get access to the UI elements (and normally has none, because the pointers macroBox and macroText remain on the stack). That mechanism guarantees, that the UI does not freeze if it happens something magic to the reading function.

Parameters

line	holds the line just got from keyboard

We want to see the keys as they appear in the macroText Widget.

Because it is possible to change the Focus via keyboard, we must set the focus on each call.

Definition at line 27 of file macroreader.cpp.

References macroBox, and macroText.

```
27                                              {
32     macroText->setFocus();
33     QTextCursor c = macroText->textCursor();
34     c.setPosition(macroText->toPlainText().length());
35     macroText->setTextCursor(c);
36     macroBox->appendPlainText(line.left(line.size()-1));
37 }
```

12.50.3.2 void MacroReaderThread::run ( )

MacroReaderThread::run is the standard main function for a thread. Tries to open a file <macroPath><macroNumber> several times (in this case, it should be possible the first time (the code was recycled from kb.cpp).

While the file is open, read lines an signal them via metaObject() to the main thread. When the file is closed by the sender, close it as reader and terminate.

Definition at line 47 of file macroreader.cpp.

References macroBox, macroNumber, macroPath, and macroText.

```
47                              {
48     qDebug() << "MacroReader::run() started with" << macroNumber << "and" <<
       macroPath << "and" << macroBox << "and" << macroText;
49
50     QFile macroFile(macroPath);
51     // Wait a small amount of time for the node to open (100ms) (code reused from kb.cpp)
52     QThread::usleep(100000);
53     if(!macroFile.open(QIODevice::ReadOnly)){
54         // If it's still not open, try again before giving up (1s at a time, 10s total)
55         QThread::usleep(900000);
56         for(int i = 1; i < 10; i++){
57             if(macroFile.open(QIODevice::ReadOnly))
58                 break;
59             QThread::sleep(1);
60         }
61         if(!macroFile.isOpen()) {
62             qDebug() << QString("unable to open macroFile (%1)").arg(macroPath);
63             return;
64         }
65     }
66     // Read data from notification node macroPath
67     // Count time between lines read from the interface
68     QByteArray line;
69     timeval t;
70     gettimeofday(&t, NULL);
71     double tstart = t.tv_sec+(t.tv_usec/1000000.0);
72     bool firstline = true;
73
74     while(macroFile.isOpen() && (line = macroFile.readLine()).length() > 0){
75         QString text = QString::fromUtf8(line);
76         gettimeofday(&t, NULL);
77         double tnow = t.tv_sec+(t.tv_usec/1000000.0);
78
79         // in the first line, there is only a delay "before start". Don't use it.
80         if (!firstline) {
```

```
81              text.prepend ("\n");
82              text.prepend (QString::number ((tnow - tstart) * 1000000.0, 'f', 0));
83              text.prepend ("=");
84          } else firstline = false;
85          tstart = tnow;
86
87          metaObject()->invokeMethod(this, "readMacro", Qt::QueuedConnection, Q_ARG(QString, text));
88      }
89      qDebug() << "MacroReader::run() ends.";
90      macroFile.close();
91      QThread::exit ();
92  }
```

12.50.4 Field Documentation

12.50.4.1 QPlainTextEdit* MacroReaderThread::macroBox [private]

Definition at line 40 of file macroreader.h.

Referenced by MacroReaderThread(), readMacro(), and run().

12.50.4.2 int MacroReaderThread::macroNumber [private]

Definition at line 31 of file macroreader.h.

Referenced by MacroReaderThread(), and run().

12.50.4.3 QString MacroReaderThread::macroPath [private]

See Also

macroNumber

Definition at line 36 of file macroreader.h.

Referenced by MacroReaderThread(), and run().

12.50.4.4 QPlainTextEdit* MacroReaderThread::macroText [private]

Definition at line 45 of file macroreader.h.

Referenced by MacroReaderThread(), readMacro(), and run().

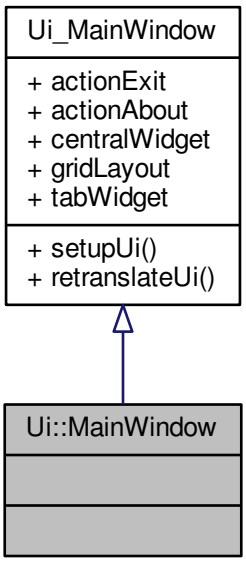
The documentation for this class was generated from the following files:

- src/ckb/macroreader.h
- src/ckb/macroreader.cpp

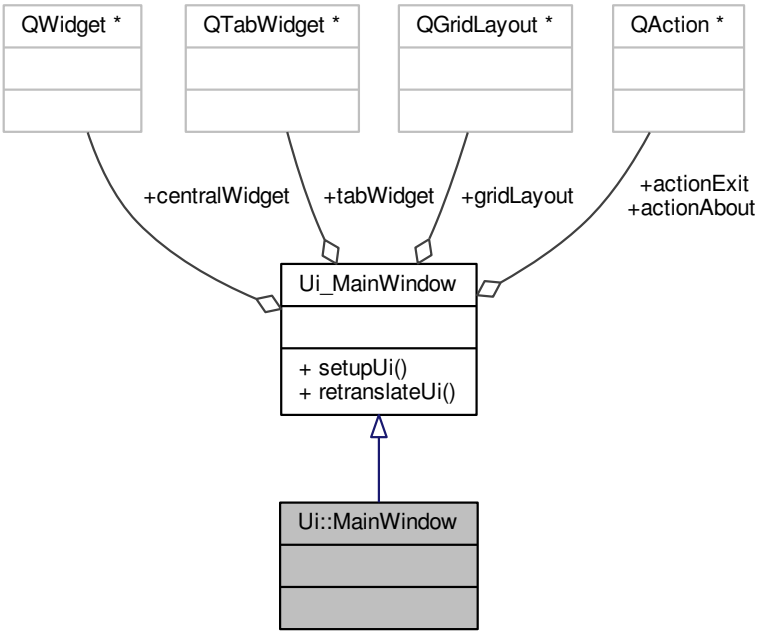
12.51 Ui::MainWindow Class Reference

`#include <src/ckb/ui_mainwindow.h>`

Inheritance diagram for Ui::MainWindow:



Collaboration diagram for Ui::MainWindow:



Public Member Functions

- void setupUi (QMainWindow *MainWindow)
- void retranslateUi (QMainWindow *MainWindow)

Data Fields

- QAction * actionExit
- QAction * actionAbout
- QWidget * centralWidget
- QGridLayout * gridLayout
- QTabWidget * tabWidget

12.51.1 Detailed Description

Definition at line 79 of file ui_mainwindow.h.

12.51.2 Member Function Documentation

12.51.2.1 void Ui_MainWindow::retranslateUi (QMainWindow * *MainWindow*) [inline], [inherited]

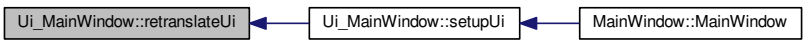
Definition at line 68 of file ui_mainwindow.h.

References Ui_MainWindow::actionAbout, and Ui_MainWindow::actionExit.

Referenced by Ui_MainWindow::setupUi().

```
69     {
70         MainWindow->setWindowTitle(QApplication::translate("MainWindow", "ckb-next", Q_NULLPTR));
71         actionExit->setText(QApplication::translate("MainWindow", "Exit", Q_NULLPTR));
72         actionExit->setShortcut(QApplication::translate("MainWindow", "Ctrl+Q", Q_NULLPTR));
73         actionAbout->setText(QApplication::translate("MainWindow", "About", Q_NULLPTR));
74     } // retranslateUi
```

Here is the caller graph for this function:



12.51.2.2 void Ui_MainWindow::setupUi (QMainWindow * *MainWindow*) [inline], [inherited]

Definition at line 33 of file ui_mainwindow.h.

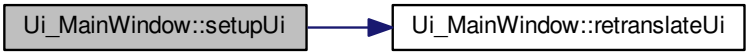
References Ui_MainWindow::actionAbout, Ui_MainWindow::actionExit, Ui_MainWindow::centralWidget, Ui_MainWindow::gridLayout, Ui_MainWindow::retranslateUi(), and Ui_MainWindow::tabWidget.

Referenced by MainWindow::MainWindow().

```
34     {
35         if (MainWindow->objectName().isEmpty())
36             MainWindow->setObjectName(QStringLiteral("MainWindow"));
37         MainWindow->resize(1000, 750);
38         MainWindow->setMinimumSize(QSize(1000, 750));
39         QIcon icon;
40         icon.addFile(QStringLiteral(":/img/ckb-logo.png"), QSize(), QIcon::Normal, QIcon::Off);
```

```
41          MainWindow->setWindowIcon(icon);
42          MainWindow->setUnifiedTitleAndToolBarOnMac(true);
43          actionExit = new QAction(MainWindow);
44          actionExit->setObjectName(QStringLiteral("actionExit"));
45          actionAbout = new QAction(MainWindow);
46          actionAbout->setObjectName(QStringLiteral("actionAbout"));
47          centralWidget = new QWidget(MainWindow);
48          centralWidget->setObjectName(QStringLiteral("centralWidget"));
49          gridLayout = new QGridLayout(centralWidget);
50          gridLayout->setSpacing(6);
51          gridLayout->setContentsMargins(11, 11, 11, 11);
52          gridLayout->setObjectName(QStringLiteral("gridLayout"));
53          tabWidget = new QTabWidget(centralWidget);
54          tabWidget->setObjectName(QStringLiteral("tabWidget"));
55
56          gridLayout->addWidget(tabWidget, 0, 0, 1, 1);
57
58          MainWindow->setCentralWidget(centralWidget);
59
60          retranslateUi(MainWindow);
61
62          tabWidget->setCurrentIndex(-1);
63
64
65          QMetaObject::connectSlotsByName(MainWindow);
66      } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.51.3 Field Documentation

12.51.3.1 QAction* Ui_MainWindow::actionAbout [inherited]

Definition at line 28 of file ui_mainwindow.h.

Referenced by Ui_MainWindow::retranslateUi(), and Ui_MainWindow::setupUi().

12.51.3.2 QAction* Ui_MainWindow::actionExit [inherited]

Definition at line 27 of file ui_mainwindow.h.

Referenced by MainWindow::MainWindow(), Ui_MainWindow::retranslateUi(), and Ui_MainWindow::setupUi().

12.51.3.3 QWidget* Ui_MainWindow::centralWidget [inherited]

Definition at line 29 of file ui_mainwindow.h.

Referenced by Ui_MainWindow::setupUi().

12.51.3.4 QGridLayout* Ui_MainWindow::gridLayout [inherited]

Definition at line 30 of file ui_mainwindow.h.

Referenced by Ui_MainWindow::setupUi().

12.51.3.5 QTabWidget* Ui_MainWindow::tabWidget [inherited]

Definition at line 31 of file ui_mainwindow.h.

Referenced by MainWindow::addDevice(), MainWindow::MainWindow(), MainWindow::removeDevice(), Ui_MainWindow::setupUi(), and MainWindow::showFwUpdateNotification().

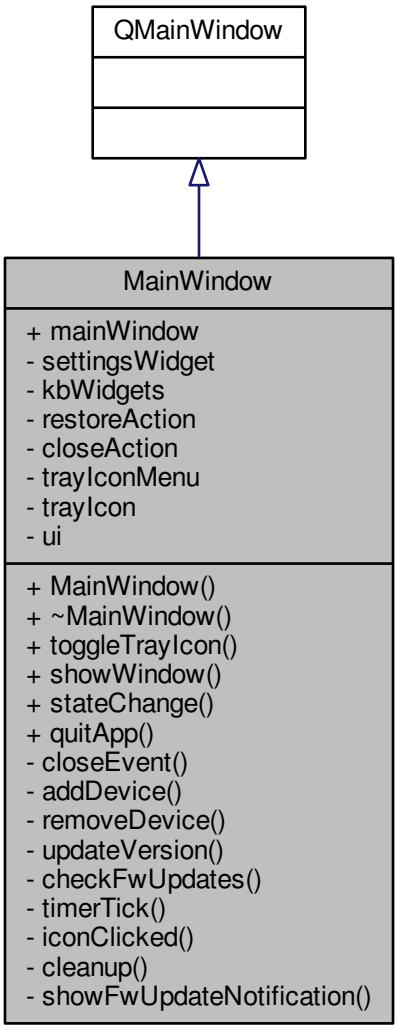
The documentation for this class was generated from the following file:

- src/ckb/ui_mainwindow.h

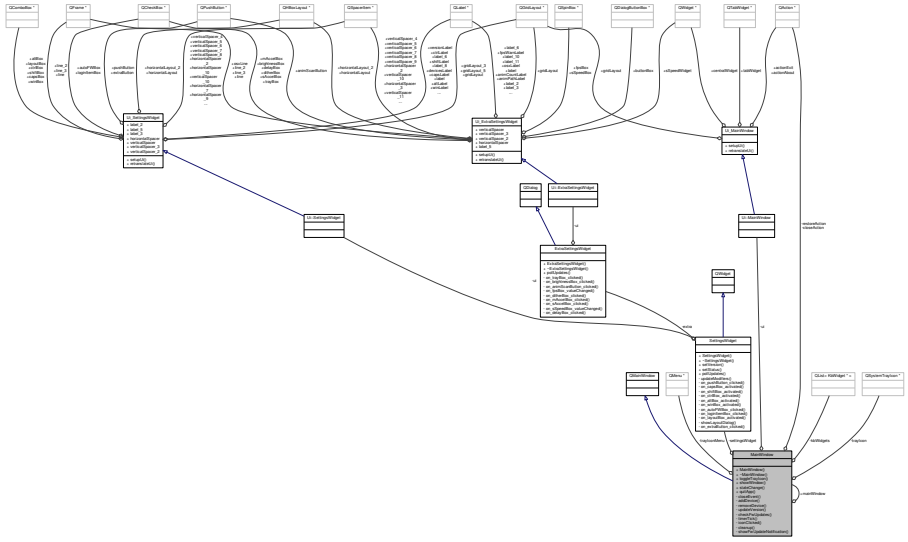
12.52 MainWindow Class Reference

`#include <src/ckb/mainwindow.h>`

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Slots

- void showWindow ()
- void stateChange (Qt::ApplicationState state)
- void quitApp ()

Public Member Functions

- MainWindow (QWidget ∗parent=0)
- ∼MainWindow ()
- void toggleTrayIcon (bool visible)

Static Public Attributes

- static MainWindow ∗ mainWindow = 0

Private Slots

- void addDevice (Kb ∗device)
- void removeDevice (Kb ∗device)
- void updateVersion ()
- void checkFwUpdates ()
- void timerTick ()
- void iconClicked (QSystemTrayIcon::ActivationReason reason)
- void cleanup ()
- void showFwUpdateNotification (QWidget ∗widget, float version)

Private Member Functions

- void closeEvent (QCloseEvent ∗event)

Private Attributes

- SettingsWidget * settingsWidget
- QList< KbWidget * > kbWidgets
- QAction * restoreAction
- QAction * closeAction
- QMenu * trayIconMenu
- QSystemTrayIcon * trayIcon
- Ui::MainWindow * ui

12.52.1 Detailed Description

Definition at line 27 of file mainwindow.h.

12.52.2 Constructor & Destructor Documentation

12.52.2.1 MainWindow::MainWindow (QWidget * *parent* = 0) [explicit]

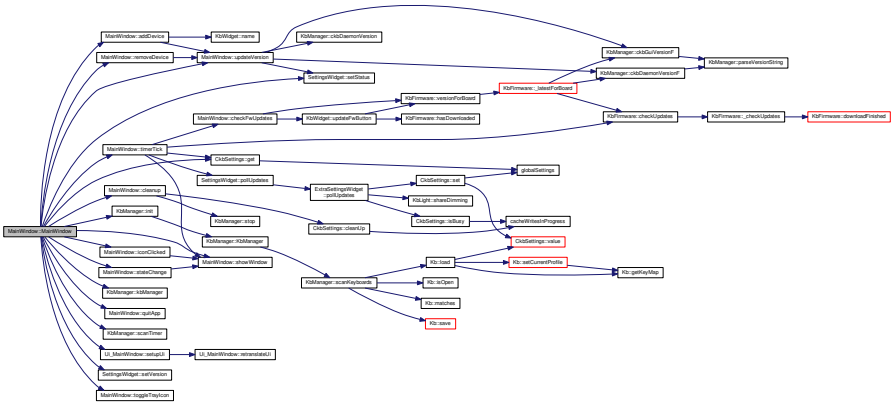
Definition at line 49 of file mainwindow.cpp.

References Ui_MainWindow::actionExit, addDevice(), cleanup(), closeAction, configLabel, daemonDialogText, devpath, CkbSettings::get(), iconClicked(), KbManager::init(), KbManager::kbManager(), mainWindow, quitApp(), removeDevice(), restoreAction, KbManager::scanTimer(), SettingsWidget::setStatus(), settingsWidget, Ui_MainWindow::setupUi(), SettingsWidget::setVersion(), showWindow(), stateChange(), Ui_MainWindow::tabWidget, timerTick(), toggleTrayIcon(), trayIcon, trayIconMenu, ui, and updateVersion().

```
49                                      :
50     QMainWindow(parent),
51     ui(new Ui::MainWindow)
52 {
53     ui->setupUi(this);
54     mainWindow = this;
55
56     // Start device manager
57     KbManager::init(CKB_VERSION_STR);
58     connect(KbManager::kbManager(), SIGNAL(kbConnected(Kb*)), this, SLOT(
       addDevice(Kb*)));
59     connect(KbManager::kbManager(), SIGNAL(kbDisconnected(Kb*)), this, SLOT(
       removeDevice(Kb*)));
60     connect(KbManager::kbManager(), SIGNAL(versionUpdated()), this, SLOT(
       updateVersion()));
61     connect(KbManager::scanTimer(), SIGNAL(timeout()), this, SLOT(
       timerTick()));
62
63     // Set up tray icon
64     restoreAction = new QAction(tr("Restore"), this);
65     closeAction = new QAction(tr("Quit ckb"), this);
66 #ifdef USE_LIBAPPINDICATOR
67     QString desktop = std::getenv("XDG_CURRENT_DESKTOP");
68     unityDesktop = (desktop.toLower() == "unity");
69
70     if(unityDesktop){
71         trayIcon = 0;
72
73         indicatorMenu = gtk_menu_new();
74         indicatorMenuRestoreItem = gtk_menu_item_new_with_label("Restore");
75         indicatorMenuQuitItem = gtk_menu_item_new_with_label("Quit ckb");
76
77         gtk_menu_shell_append(GTK_MENU_SHELL(indicatorMenu), indicatorMenuRestoreItem);
78         gtk_menu_shell_append(GTK_MENU_SHELL(indicatorMenu), indicatorMenuQuitItem);
79
80         g_signal_connect(indicatorMenuQuitItem, "activate",
81             G_CALLBACK(quitIndicator), this);
82         g_signal_connect(indicatorMenuRestoreItem, "activate",
83             G_CALLBACK(restoreIndicator), this);
84
85         gtk_widget_show(indicatorMenuRestoreItem);
86         gtk_widget_show(indicatorMenuQuitItem);
87
88         indicator = app_indicator_new("ckb", "indicator-messages",
       APP_INDICATOR_CATEGORY_APPLICATION_STATUS);
89
```

```
90          app_indicator_set_status(indicator, APP_INDICATOR_STATUS_ACTIVE);
91          app_indicator_set_menu(indicator, GTK_MENU(indicatorMenu));
92          app_indicator_set_icon(indicator, "ckb");
93      } else
94  #endif // USE_LIBAPPINDICATOR
95      {
96          trayIconMenu = new QMenu(this);
97          trayIconMenu->addAction(restoreAction);
98          trayIconMenu->addAction(closeAction);
99          trayIcon = new QSystemTrayIcon(QIcon(":/img/ckb-logo.png"), this);
100          trayIcon->setContextMenu(trayIconMenu);
101          connect(trayIcon, SIGNAL(activated(QSystemTrayIcon::ActivationReason)), this, SLOT(
      iconClicked(QSystemTrayIcon::ActivationReason)));
102      }
103      toggleTrayIcon(!CkbSettings::get("Program/SuppressTrayIcon").toBool());
104
105  #ifdef Q_OS_MACX
106      // Make a custom "Close" menu action for OSX, as the default one brings up the "still running" popup
       unnecessarily
107      QMenuBar* menuBar = new QMenuBar(this);
108      setMenuBar(menuBar);
109      this->menuBar()->addMenu("ckb")->addAction(closeAction);
110  #else
111      // On linux, add a handler for Ctrl+Q
112      new QShortcut(QKeySequence("Ctrl+Q"), this, SLOT(quitApp()));
113  #endif
114
115      connect(ui->actionExit, SIGNAL(triggered()), this, SLOT(quitApp()));
116      connect(closeAction, SIGNAL(triggered()), this, SLOT(quitApp()));
117      connect(restoreAction, SIGNAL(triggered()), this, SLOT(
      showWindow()));
118      connect(qApp, SIGNAL(applicationStateChanged(Qt::ApplicationState)), this, SLOT(
      stateChange(Qt::ApplicationState)));
119
120      connect(qApp, SIGNAL(aboutToQuit()), this, SLOT(cleanup()));
121
122      ui->tabWidget->addTab(settingsWidget = new
      SettingsWidget(this), configLabel);
123      settingsWidget->setVersion("ckb-next " CKB_VERSION_STR);
124
125      // create daemon dialog as a QMessageBox
126      // this will create a focussed dialog, that has to be interacted with,
127      // if the daemon is not running
128      // set the main and informative text to tell the user about the issue
129      QMessageBox dialog;
130      dialog.setText(tr("The ckb-next daemon is not running. This program will <b>not</b> work without it!"))
      ;
131      dialog.setInformativeText(daemonDialogText);
132      dialog.setIcon(QMessageBox::Critical);
133
134      // check, whether daemon is running
135      // the daemon creates the root device path on initialization and thus it
136      // can be assumed, that the daemon is not running if doesn't exist
137      // `.arg(0)` is necessary to interpolate the correct suffix into the path
138      // see `./kbmanager.cpp` for details
139      QFileInfo rootDevPath(devpath.arg(0));
140      if (!rootDevPath.exists()) {
141          // set settings widget's status
142          // show the main window (otherwise only the dialog will be visible)
143          // finally show the dialog
144          settingsWidget->setStatus(tr("The ckb-next daemon is not running."));
145          showWindow();
146          dialog.exec();
147      }
148  }
```

Here is the call graph for this function:



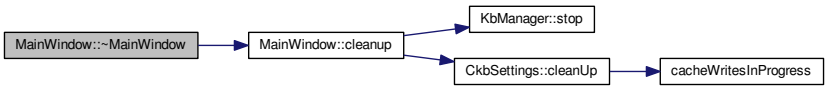
12.52.2.2 MainWindow::~MainWindow ( )

Definition at line 341 of file mainwindow.cpp.

References cleanup(), and ui.

```
341                          {
342     cleanup();
343     delete ui;
344 }
```

Here is the call graph for this function:



12.52.3 Member Function Documentation

12.52.3.1 void MainWindow::addDevice (Kb * *device*) [private], [slot]

Definition at line 159 of file mainwindow.cpp.

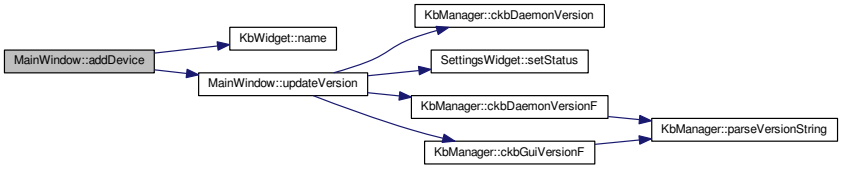
References KbWidget::device, kbWidgets, KbWidget::name(), Ui_MainWindow::tabWidget, ui, and updateVersion().

Referenced by MainWindow().

```
159                                    {
160     // Connected already?
161     foreach(KbWidget* w, kbWidgets){
162         if(w->device == device)
163             return;
164     }
165     // Add the keyboard
166     KbWidget* widget = new KbWidget(this, device);
167     kbWidgets.append(widget);
168     // Add to tabber; switch to this device if the user is on the settings screen
169     int count = ui->tabWidget->count();
170     ui->tabWidget->insertTab(count - 1, widget, widget->name());
171     if(ui->tabWidget->currentIndex() == count)
172         ui->tabWidget->setCurrentIndex(count - 1);
```

```
173     // Update connected device count
174     updateVersion();
175 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.52.3.2 void MainWindow::checkFwUpdates ( ) [private], [slot]

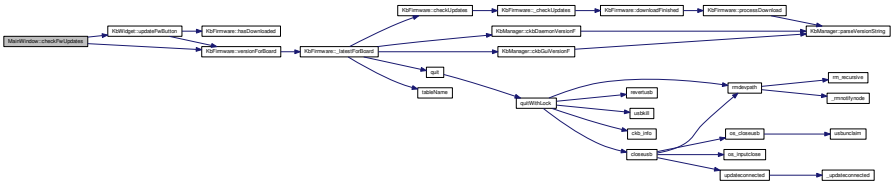
Definition at line 210 of file mainwindow.cpp.

References KbWidget::device, Kb::features, Kb::firmware, KbWidget::hasShownNewFW, kbWidgets, mainWindow, KbWidget::updateFwButton(), and KbFirmware::versionForBoard().

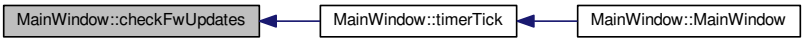
Referenced by timerTick().

```
210                                {
211     if(!mainWindow->isVisible())
212         return;
213     foreach(KbWidget* w, kbWidgets){
214         // Display firmware upgrade notification if a new version is available
215         float version = KbFirmware::versionForBoard(w->
    device->features);
216         if(version > w->device->firmware.toFloat()){
217             if(w->hasShownNewFW)
218                 continue;
219             w->hasShownNewFW = true;
220             w->updateFwButton();
221             // Don't run this method here because it will lock up the timer and prevent devices from
     working properly
222             // Use a queued invocation instead
223             metaObject()->invokeMethod(this, "showFwUpdateNotification", Qt::QueuedConnection, Q_ARG(
    QWidget*, w), Q_ARG(float, version));
224             // Don't display more than one of these at once
225             return;
226         }
227     }
228 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.52.3.3 void MainWindow::cleanup ( ) [private],[slot]

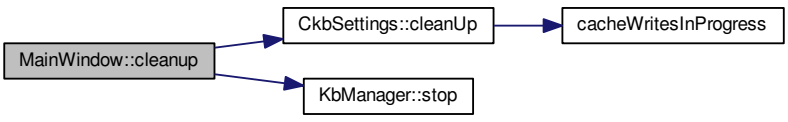
Definition at line 333 of file mainwindow.cpp.

References CkbSettings::cleanUp(), kbWidgets, and KbManager::stop().

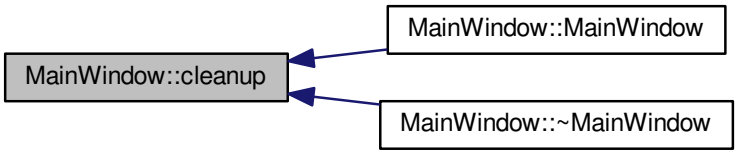
Referenced by MainWindow(), and ~MainWindow().

```
333                         {
334     foreach(KbWidget* w, kbWidgets)
335         delete w;
336     kbWidgets.clear();
337     KbManager::stop();
338     CkbSettings::cleanUp();
339 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



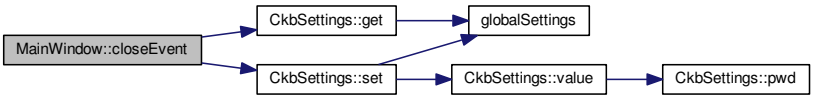
12.52.3.4 void MainWindow::closeEvent (QCloseEvent * *event*) [private]

Definition at line 247 of file mainwindow.cpp.

References CkbSettings::get(), and CkbSettings::set().

```
247                                          {
248     // If the window is hidden already or the event is non-spontaneous (can happen on OSX when using the
       Quit menu), accept it and close
249     if(!event->spontaneous() || isHidden()){
250         event->accept();
251         return;
252     }
253     if(!CkbSettings::get("Popups/BGWarning").toBool()){
254         QMessageBox::information(this, "ckb", "ckb will still run in the background.\nTo close it, choose
       Exit from the tray menu\nor click \"Quit ckb\" on the Settings screen.");
255         CkbSettings::set("Popups/BGWarning", true);
256     }
257     hide();
258     event->ignore();
259 }
```

Here is the call graph for this function:



12.52.3.5 void MainWindow::iconClicked (QSystemTrayIcon::ActivationReason *reason*) [private], [slot]

Definition at line 295 of file mainwindow.cpp.

References showWindow().

Referenced by MainWindow().

```
295                                                                {
296     // On Linux, hide/show the app when the tray icon is clicked
297     // On OSX this just shows the menu
298 #ifndef Q_OS_MACX
299     if(reason == QSystemTrayIcon::DoubleClick || reason == QSystemTrayIcon::Trigger){
300         if(isVisible())
301             hide();
302         else
```

```
303             showWindow();
304     }
305 #endif
306 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

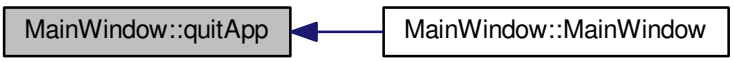
12.52.3.6 void MainWindow::quitApp ( ) [slot]

Definition at line 329 of file mainwindow.cpp.

Referenced by MainWindow().

```
329                     {
330     qApp->quit();
331 }
```

Here is the caller graph for this function:



12.52.3.7 void MainWindow::removeDevice (Kb * *device*) [private], [slot]

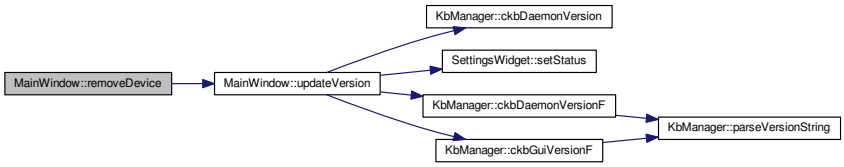
Definition at line 177 of file mainwindow.cpp.

References KbWidget::device, kbWidgets, Ui_MainWindow::tabWidget, ui, and updateVersion().

Referenced by MainWindow().

```
177                                         {
178     foreach(KbWidget* w, kbWidgets){
179         // Remove this device from the UI
180         if(w->device == device){
181             int i = kbWidgets.indexOf(w);
182             ui->tabWidget->removeTab(i);
183             kbWidgets.removeAt(i);
184             w->deleteLater();
185         }
186     }
187     // Update connected device count
188     updateVersion();
189 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

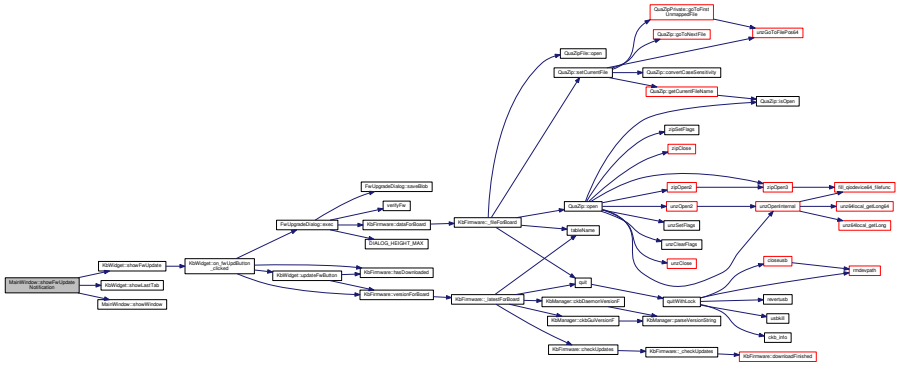
12.52.3.8 void MainWindow::showFwUpdateNotification (QWidget * *widget*, float *version*) [private], [slot]

Definition at line 230 of file mainwindow.cpp.

References KbWidget::device, kbWidgets, KbWidget::showFwUpdate(), KbWidget::showLastTab(), showWindow(), Ui_MainWindow::tabWidget, ui, and Kb::usbModel.

```
230                                                                                 {
231     static bool isShowing = false;
232     if(isShowing)
233         return;
234     isShowing = true;
235     showWindow();
236     KbWidget* w = (KbWidget*)widget;
237     // Ask for update
238     if(QMessageBox::information(this, "Firmware update", tr("A new firmware is available for your %1 (v%2)
      \nWould you like to install it now?").arg(w->device->usbModel, QString::number(version, 'f', 2)
      ), QMessageBox::StandardButtons(QMessageBox::Yes | QMessageBox::No), QMessageBox::Yes) == QMessageBox::Yes){
239         // If accepted, switch to the firmware tab and bring up the update window
240         w->showLastTab();
241         ui->tabWidget->setCurrentIndex(kbWidgets.indexOf(w));
242         w->showFwUpdate();
243     }
244     isShowing = false;
245 }
```

Here is the call graph for this function:



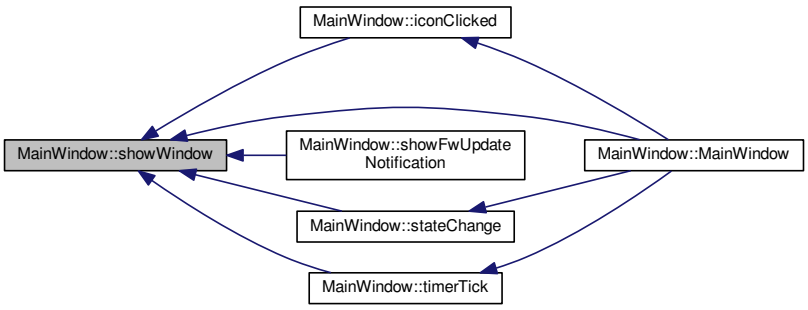
12.52.3.9 void MainWindow::showWindow ( ) [slot]

Definition at line 308 of file mainwindow.cpp.

Referenced by iconClicked(), MainWindow(), showFwUpdateNotification(), stateChange(), and timerTick().

```
308                              {
309     showNormal();
310     raise();
311     activateWindow();
312 }
```

Here is the caller graph for this function:



12.52.3.10 void MainWindow::stateChange (Qt::ApplicationState *state*) [slot]

Definition at line 314 of file mainwindow.cpp.

References showWindow().

Referenced by MainWindow().

```
314                                                 {
315     // On OSX it's possible for the app to be brought to the foreground without the window actually
      reappearing.
316     // We want to make sure it's shown when this happens.
317 #ifdef Q_OS_MAC
```

```
318     static quint64 lastStateChange = 0;
319     quint64 now = QDateTime::currentMSecsSinceEpoch();
320     if(state == Qt::ApplicationActive){
321         // This happens once at startup so ignore it. Also don't allow it to be called more than once every
        2s.
322         if(lastStateChange != 0 && now >= lastStateChange + 2 * 1000)
323             showWindow();
324         lastStateChange = now;
325     }
326 #endif
327 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.52.3.11 void MainWindow::timerTick ( ) [private], [slot]

Definition at line 261 of file mainwindow.cpp.

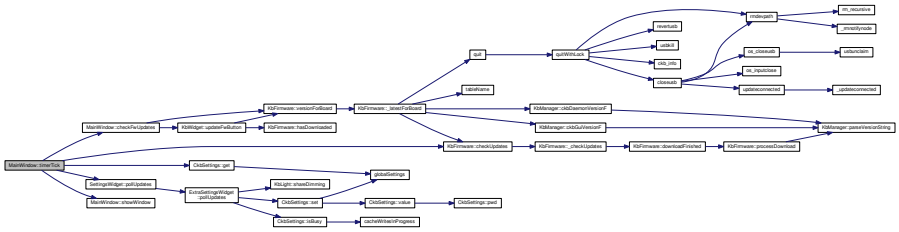
References appShare, checkFwUpdates(), KbFirmware::checkUpdates(), CkbSettings::get(), SettingsWidget::pollUpdates(), settingsWidget, and showWindow().

Referenced by MainWindow().

```
261                         {
262     // Check shared memory for changes
263     if(appShare.lock()){
264         void* data = appShare.data();
265         QStringList commands = QString((const char*)data).split("\n");
266         // Restore PID, remove all other data
267         snprintf((char*)appShare.data(), appShare.size(), "PID %ld", (long)getpid());
268         appShare.unlock();
269         // Parse commands
270         foreach(const QString& line, commands){
271             // Old ckb option line - bring application to foreground
272             if(line == "Open")
273                 showWindow();
274             if(line.startsWith("Option ")){
275                 // New ckb option line
276                 QString option = line.split(" ")[1];
277                 if(option == "Open")
278                     // Bring to foreground
279                     showWindow();
280                 else if(option == "Close")
281                     // Quit application
282                     qApp->quit();
283             }
284         }
```

```
285     }
286     // Check for firmware updates (when appropriate)
287     if(!CkbSettings::get("Program/DisableAutoFWCheck").toBool()){
288         KbFirmware::checkUpdates();
289         checkFwUpdates();
290     }
291     // Poll for setting updates
292     settingsWidget->pollUpdates();
293 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.52.3.12 void MainWindow::toggleTrayIcon (bool *visible*)

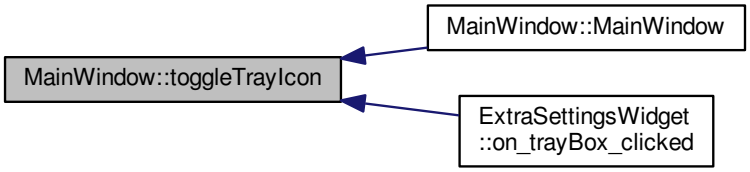
Definition at line 150 of file mainwindow.cpp.

References trayIcon.

Referenced by MainWindow(), and ExtraSettingsWidget::on_trayBox_clicked().

```
150                                         {
151 #ifdef USE_LIBAPPINDICATOR
152     if(unityDesktop)
153         app_indicator_set_status(indicator, visible ? APP_INDICATOR_STATUS_ACTIVE :
      APP_INDICATOR_STATUS_PASSIVE);
154     else
155 #endif // USE_LIBAPPINDICATOR
156         trayIcon->setVisible(visible);
157 }
```

Here is the caller graph for this function:



12.52.3.13 void MainWindow::updateVersion ( ) [private], [slot]

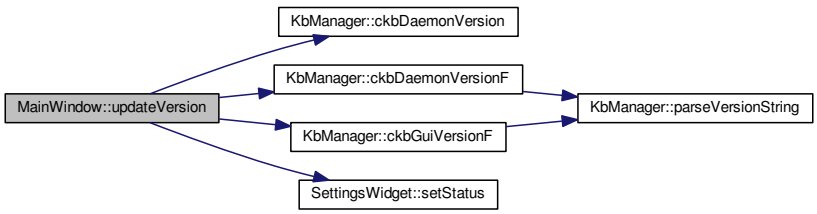
Definition at line 191 of file mainwindow.cpp.

References KbManager::ckbDaemonVersion(), KbManager::ckbDaemonVersionF(), KbManager::ckbGuiVersionF(), DAEMON_UNAVAILABLE_STR, kbWidgets, SettingsWidget::setStatus(), and settingsWidget.

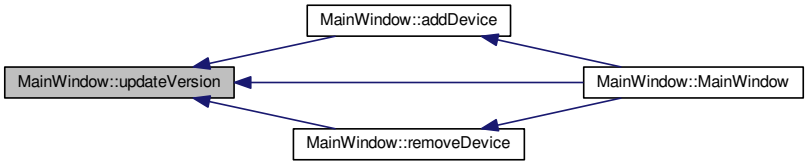
Referenced by addDevice(), MainWindow(), and removeDevice().

```
191                             {
192     QString daemonVersion = KbManager::ckbDaemonVersion();
193     if(daemonVersion == DAEMON_UNAVAILABLE_STR){
194         settingsWidget->setStatus("Driver inactive");
195         return;
196     }
197     int count = kbWidgets.count();
198     // Warn if the daemon version doesn't match the GUI
199     QString daemonWarning;
200     if(daemonVersion != CKB_VERSION_STR)
201         daemonWarning = "<br /><br /><b>Warning:</b> Driver version mismatch (" + daemonVersion + ").
       Please upgrade ckb" + QString(KbManager::ckbDaemonVersionF() >
      KbManager::ckbGuiVersionF() ? "" : "-daemon") + ". If the problem persists, try
       rebooting.";
202     if(count == 0)
203         settingsWidget->setStatus("No devices connected" + daemonWarning);
204     else if(count == 1)
205         settingsWidget->setStatus("1 device connected" + daemonWarning);
206     else
207         settingsWidget->setStatus(QString("%1 devices connected").arg(count) +
      daemonWarning);
208 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.52.4 Field Documentation

12.52.4.1 QAction* MainWindow::closeAction [private]

Definition at line 44 of file mainwindow.h.

Referenced by MainWindow().

12.52.4.2 QList<KbWidget*> MainWindow::kbWidgets [private]

Definition at line 41 of file mainwindow.h.

Referenced by addDevice(), checkFwUpdates(), cleanup(), removeDevice(), showFwUpdateNotification(), and updateVersion().

12.52.4.3 MainWindow * MainWindow::mainWindow = 0 [static]

Definition at line 35 of file mainwindow.h.

Referenced by checkFwUpdates(), MainWindow(), and ExtraSettingsWidget::on_trayBox_clicked().

12.52.4.4 QAction* MainWindow::restoreAction [private]

Definition at line 43 of file mainwindow.h.

Referenced by MainWindow().

12.52.4.5 SettingsWidget* MainWindow::settingsWidget [private]

Definition at line 40 of file mainwindow.h.

Referenced by MainWindow(), timerTick(), and updateVersion().

12.52.4.6 QSystemTrayIcon* MainWindow::trayIcon [private]

Definition at line 54 of file mainwindow.h.

Referenced by MainWindow(), and toggleTrayIcon().

12.52.4.7 QMenu* MainWindow::trayIconMenu [private]

Definition at line 53 of file mainwindow.h.

Referenced by MainWindow().

12.52.4.8 Ui::MainWindow* MainWindow::ui [private]

Definition at line 75 of file mainwindow.h.

Referenced by addDevice(), MainWindow(), removeDevice(), showFwUpdateNotification(), and ~MainWindow().

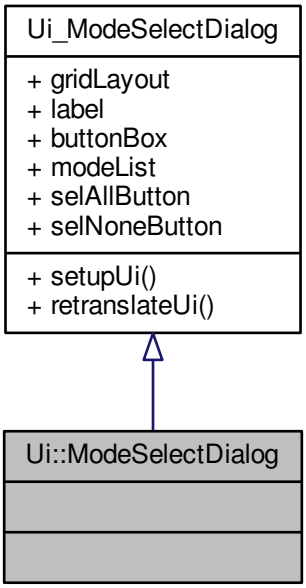
The documentation for this class was generated from the following files:

- src/ckb/mainwindow.h
- src/ckb/mainwindow.cpp

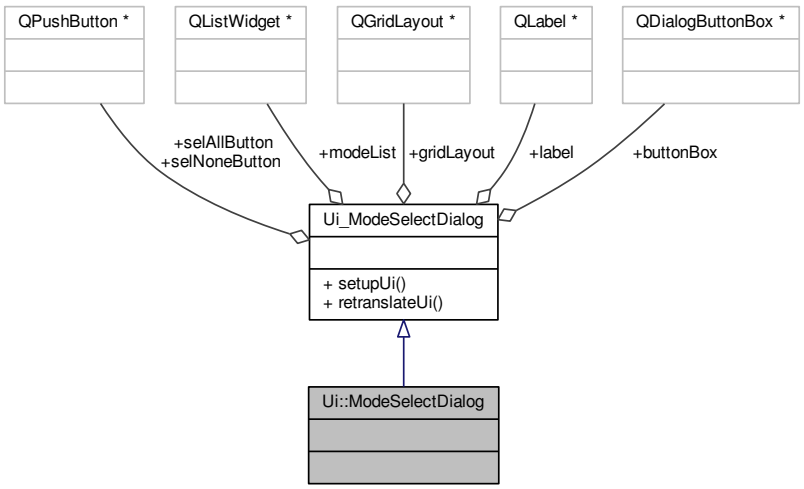
12.53 Ui::ModeSelectDialog Class Reference

`#include <src/ckb/ui_modeselectdialog.h>`

Inheritance diagram for Ui::ModeSelectDialog:



Collaboration diagram for Ui::ModeSelectDialog:



Public Member Functions

- void setupUi (QDialog *ModeSelectDialog)
- void retranslateUi (QDialog *ModeSelectDialog)

Data Fields

- QGridLayout * gridLayout
- QLabel * label
- QDialogButtonBox * buttonBox
- QListWidget * modeList
- QPushButton * selAllButton
- QPushButton * selNoneButton

12.53.1 Detailed Description

Definition at line 95 of file ui_modeselectdialog.h.

12.53.2 Member Function Documentation

12.53.2.1 void Ui_ModeSelectDialog::retranslateUi (QDialog * *ModeSelectDialog*) [inline], [inherited]

Definition at line 84 of file ui_modeselectdialog.h.

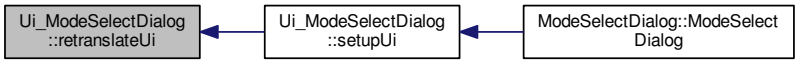
References Ui_ModeSelectDialog::label, Ui_ModeSelectDialog::selAllButton, and Ui_ModeSelectDialog::selNoneButton.

Referenced by Ui_ModeSelectDialog::setupUi().

```
85     {
86         ModeSelectDialog->setWindowTitle(QApplication::translate("ModeSelectDialog", "Select Modes",
    Q_NULLPTR));
87         label->setText(QApplication::translate("ModeSelectDialog", "TextLabel", Q_NULLPTR));
```

```
88          selAllButton->setText(QApplication::translate("ModeSelectDialog", "Select All",
    Q_NULLPTR));
89          selNoneButton->setText(QApplication::translate("ModeSelectDialog", "Select None",
    Q_NULLPTR));
90      } // retranslateUi
```

Here is the caller graph for this function:



12.53.2.2 void Ui_ModeSelectDialog::setupUi (QDialog * *ModeSelectDialog*) [inline],[inherited]

Definition at line 36 of file ui_modeselectdialog.h.

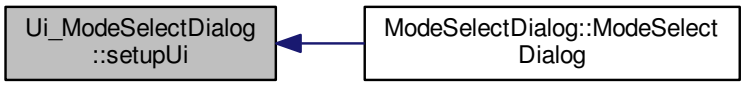
References Ui_ModeSelectDialog::buttonBox, Ui_ModeSelectDialog::gridLayout, Ui_ModeSelectDialog::label, Ui_-ModeSelectDialog::modeList, Ui_ModeSelectDialog::retranslateUi(), Ui_ModeSelectDialog::selAllButton, and Ui_-ModeSelectDialog::selNoneButton.

Referenced by ModeSelectDialog::ModeSelectDialog().

```
37      {
38          if (ModeSelectDialog->objectName().isEmpty())
39              ModeSelectDialog->setObjectName(QStringLiteral("ModeSelectDialog"));
40          ModeSelectDialog->resize(320, 320);
41          gridLayout = new QGridLayout(ModeSelectDialog);
42          gridLayout->setObjectName(QStringLiteral("gridLayout"));
43          label = new QLabel(ModeSelectDialog);
44          label->setObjectName(QStringLiteral("label"));
45          label->setAlignment(Qt::AlignLeading|Qt::AlignLeft|Qt::AlignTop);
46
47          gridLayout->addWidget(label, 0, 0, 1, 2);
48
49          buttonBox = new QDialogButtonBox(ModeSelectDialog);
50          buttonBox->setObjectName(QStringLiteral("buttonBox"));
51          buttonBox->setOrientation(Qt::Vertical);
52          buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
53          buttonBox->setCenterButtons(false);
54
55          gridLayout->addWidget(buttonBox, 0, 2, 3, 1);
56
57          modeList = new QListWidget(ModeSelectDialog);
58          modeList->setObjectName(QStringLiteral("modeList"));
59          modeList->setFrameShadow(QFrame::Plain);
60          modeList->setSelectionMode(QAbstractItemView::ExtendedSelection);
61
62          gridLayout->addWidget(modeList, 1, 0, 1, 2);
63
64          selAllButton = new QPushButton(ModeSelectDialog);
65          selAllButton->setObjectName(QStringLiteral("selAllButton"));
66
67          gridLayout->addWidget(selAllButton, 2, 0, 1, 1);
68
69          selNoneButton = new QPushButton(ModeSelectDialog);
70          selNoneButton->setObjectName(QStringLiteral("selNoneButton"));
71
72          gridLayout->addWidget(selNoneButton, 2, 1, 1, 1);
73
74          QWidget::setTabOrder(modeList, selAllButton);
75          QWidget::setTabOrder(selAllButton, selNoneButton);
76
77          retranslateUi(ModeSelectDialog);
78          QObject::connect(buttonBox, SIGNAL(accepted()), ModeSelectDialog, SLOT(accept()));
79          QObject::connect(buttonBox, SIGNAL(rejected()), ModeSelectDialog, SLOT(reject()));
80
81          QMetaObject::connectSlotsByName(ModeSelectDialog);
82      } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.53.3 Field Documentation

12.53.3.1 QDialogButtonBox* Ui_ModeSelectDialog::buttonBox [inherited]

Definition at line 31 of file ui_modeselectdialog.h.

Referenced by Ui_ModeSelectDialog::setupUi().

12.53.3.2 QGridLayout* Ui_ModeSelectDialog::gridLayout [inherited]

Definition at line 29 of file ui_modeselectdialog.h.

Referenced by Ui_ModeSelectDialog::setupUi().

12.53.3.3 QLabel* Ui_ModeSelectDialog::label [inherited]

Definition at line 30 of file ui_modeselectdialog.h.

Referenced by ModeSelectDialog::ModeSelectDialog(), Ui_ModeSelectDialog::retranslateUi(), and Ui_ModeSelectDialog::setupUi().

12.53.3.4 QListWidget* Ui_ModeSelectDialog::modeList [inherited]

Definition at line 32 of file ui_modeselectdialog.h.

Referenced by ModeSelectDialog::ModeSelectDialog(), ModeSelectDialog::on_selAllButton_clicked(), ModeSelectDialog::on_selNoneButton_clicked(), ModeSelectDialog::selection(), and Ui_ModeSelectDialog::setupUi().

12.53.3.5 QPushButton* Ui_ModeSelectDialog::selAllButton [inherited]

Definition at line 33 of file ui_modeselectdialog.h.

Referenced by Ui_ModeSelectDialog::retranslateUi(), and Ui_ModeSelectDialog::setupUi().

12.53.3.6 QPushButton* Ui_ModeSelectDialog::selNoneButton [inherited]

Definition at line 34 of file ui_modeselectdialog.h.

Referenced by Ui_ModeSelectDialog::retranslateUi(), and Ui_ModeSelectDialog::setupUi().

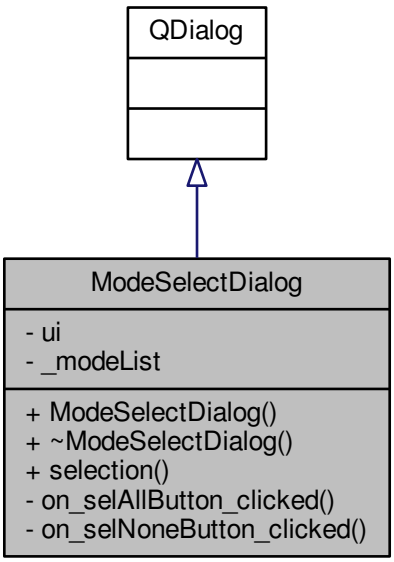
The documentation for this class was generated from the following file:

- src/ckb/ui_modeselectdialog.h

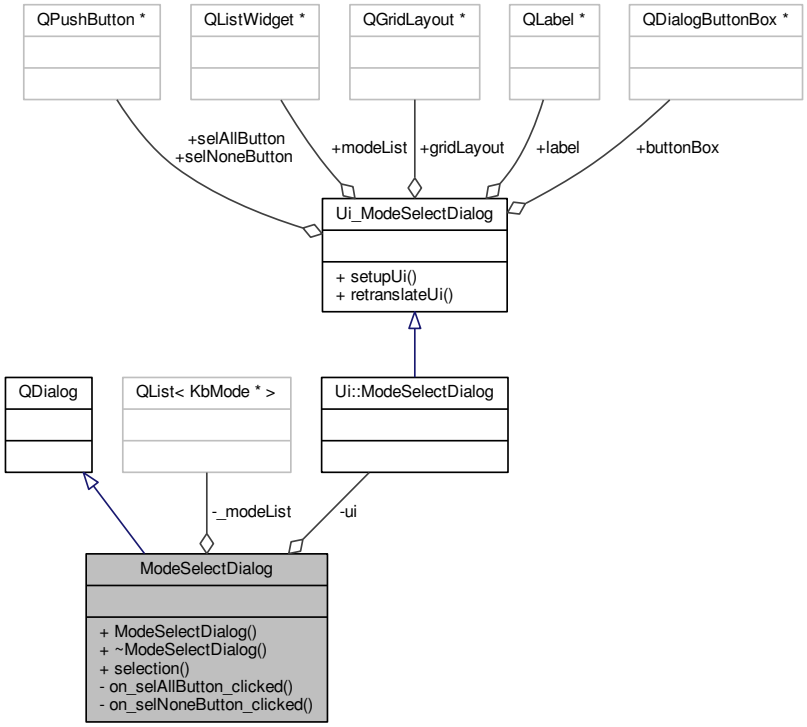
12.54 ModeSelectDialog Class Reference

`#include <src/ckb/modeselectdialog.h>`

Inheritance diagram for ModeSelectDialog:



Collaboration diagram for ModeSelectDialog:



Public Member Functions

- ModeSelectDialog (QWidget *parent, KbMode *currentMode, QList< KbMode * > modeList, const QString &textLabel)
- ~ModeSelectDialog ()
- QList< KbMode * > selection ()

Private Slots

- void on_selAllButton_clicked ()
- void on_selNoneButton_clicked ()

Private Attributes

- Ui::ModeSelectDialog * ui
- QList< KbMode * > _modeList

12.54.1 Detailed Description

Definition at line 13 of file modeselectdialog.h.

12.54.2 Constructor & Destructor Documentation

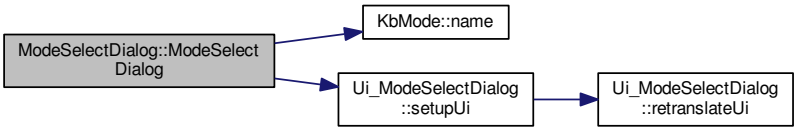
12.54.2.1 ModeSelectDialog::ModeSelectDialog (QWidget * *parent*, KbMode * *currentMode*, QList< KbMode * > *modeList*, const QString & *textLabel*)

Definition at line 4 of file modeselectdialog.cpp.

References _modeList, Ui_ModeSelectDialog::label, Ui_ModeSelectDialog::modeList, KbMode::name(), Ui_ModeSelectDialog::setupUi(), and ui.

```
4
                    :
5     QDialog(parent), ui(new Ui::ModeSelectDialog),
6     _modeList(modeList)
7 {
8     ui->setupUi(this);
9     _modeList.removeAll(currentMode);
10     ui->label->setText(textLabel);
11     // Populate list
12     foreach(KbMode* mode, _modeList)
13         ui->modeList->addItem(mode->name());
14 }
```

Here is the call graph for this function:



12.54.2.2 ModeSelectDialog::~ModeSelectDialog ( )

Definition at line 16 of file modeselectdialog.cpp.

References ui.

```
16                                  {
17     delete ui;
18 }
```

12.54.3 Member Function Documentation

12.54.3.1 void ModeSelectDialog::on_selAllButton_clicked ( ) [private], [slot]

Definition at line 30 of file modeselectdialog.cpp.

References Ui_ModeSelectDialog::modeList, and ui.

```
30                                          {
31     int count = ui->modeList->count();
32     for(int i = 0; i < count; i++)
33         ui->modeList->item(i)->setSelected(true);
34 }
```

12.54.3.2 void ModeSelectDialog::on_selNoneButton_clicked ( ) [private], [slot]

Definition at line 36 of file modeselectdialog.cpp.

References Ui_ModeSelectDialog::modeList, and ui.

```
36                                                {
37     int count = ui->modeList->count();
38     for(int i = 0; i < count; i++)
39         ui->modeList->item(i)->setSelected(false);
40 }
```

12.54.3.3 QList< KbMode * > ModeSelectDialog::selection ( )

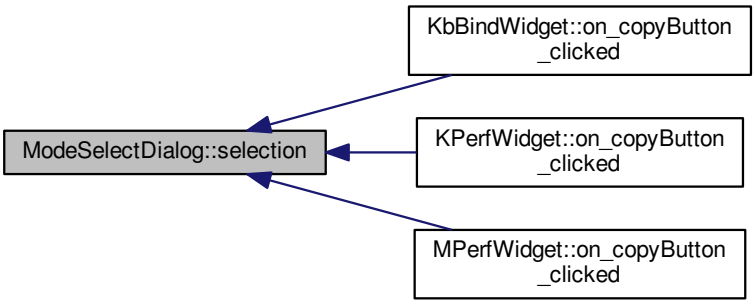
Definition at line 20 of file modeselectdialog.cpp.

References _modeList, Ui_ModeSelectDialog::modeList, and ui.

Referenced by KbBindWidget::on_copyButton_clicked(), KPerfWidget::on_copyButton_clicked(), and MPerfWidget-::on_copyButton_clicked().

```
20                                              {
21     QList<KbMode*> res;
22     int count = ui->modeList->count();
23     for(int i = 0; i < count; i++){
24         if(ui->modeList->item(i)->isSelected())
25             res.append(_modeList[i]);
26     }
27     return res;
28 }
```

Here is the caller graph for this function:



12.54.4 Field Documentation

12.54.4.1 QList<KbMode*> ModeSelectDialog::_modeList [private]

Definition at line 31 of file modeselectdialog.h.

Referenced by ModeSelectDialog(), and selection().

12.54.4.2 Ui::ModeSelectDialog* ModeSelectDialog::ui [private]

Definition at line 30 of file modeselectdialog.h.

Referenced by ModeSelectDialog(), on_selAllButton_clicked(), on_selNoneButton_clicked(), selection(), and ~-ModeSelectDialog().

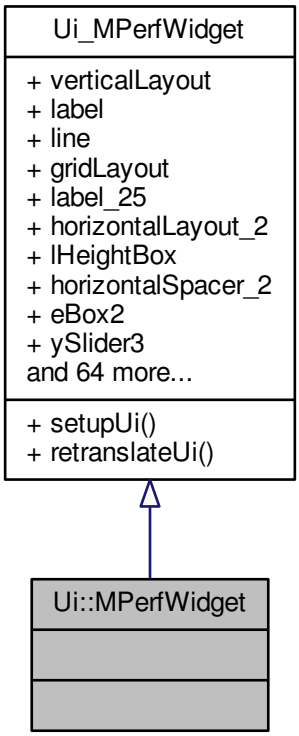
The documentation for this class was generated from the following files:

- src/ckb/modeselectdialog.h
- src/ckb/modeselectdialog.cpp

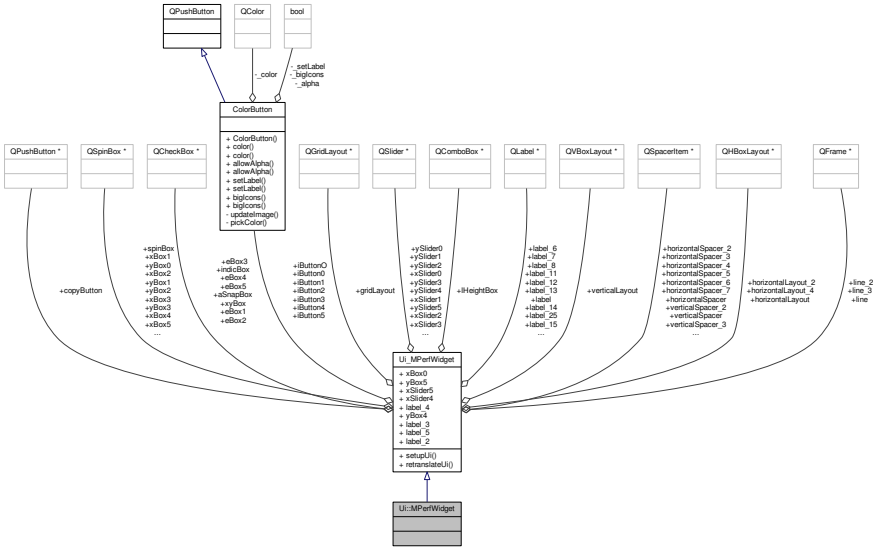
12.55 Ui::MPerfWidget Class Reference

`#include <src/ckb/ui_mperfwidget.h>`

Inheritance diagram for Ui::MPerfWidget:



Collaboration diagram for Ui::MPerfWidget:



Public Member Functions

- void setupUi (QWidget *MPerfWidget)
- void retranslateUi (QWidget *MPerfWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QLabel * label
- QFrame * line
- QGridLayout * gridLayout
- QLabel * label_25
- QHBoxLayout * horizontalLayout_2
- QComboBox * lHeightBox
- QSpacerItem * horizontalSpacer_2
- QCheckBox * eBox2
- QSlider * ySlider3
- QSlider * xSlider2
- QSpinBox * xBox3
- QSpinBox * yBox0
- QSlider * ySlider0
- ColorButton * iButton4
- ColorButton * iButton5
- QSpacerItem * horizontalSpacer_4
- QSpacerItem * verticalSpacer_3
- QSpacerItem * horizontalSpacer_5
- ColorButton * iButton3
- QLabel * label_12
- QSpinBox * xBox4
- QSpinBox * xBox0
- QSpinBox * yBox5
- QCheckBox * eBox5

- QSlider * xSlider5
- QSpinBox * xBox1
- QSpinBox * yBox2
- QSpinBox * yBox3
- QSlider * xSlider4
- QLabel * label_4
- QSlider * xSlider0
- ColorButton * iButton2
- QFrame * line_3
- ColorButton * iButtonO
- QSpinBox * yBox1
- QSlider * ySlider2
- QSpinBox * xBox2
- QSlider * xSlider1
- QSpacerItem * horizontalSpacer
- QSpinBox * yBox4
- ColorButton * iButton1
- QSlider * ySlider4
- QSlider * ySlider1
- ColorButton * iButton0
- QSlider * xSlider3
- QSpinBox * xBox5
- QLabel * label_3
- QLabel * label_6
- QSlider * ySlider5
- QLabel * label_7
- QLabel * label_8
- QLabel * label_5
- QLabel * label_15
- QLabel * label_13
- QLabel * label_14
- QCheckBox * eBox4
- QLabel * label_11
- QCheckBox * eBox3
- QCheckBox * eBox1
- QLabel * label_2
- QSpacerItem * verticalSpacer
- QCheckBox * aSnapBox
- QFrame * line_2
- QHBoxLayout * horizontalLayout_4
- QCheckBox * indicBox
- QSpacerItem * horizontalSpacer_3
- QSpinBox * spinBox
- QSpacerItem * horizontalSpacer_7
- QCheckBox * xyBox
- QSpacerItem * verticalSpacer_2
- QHBoxLayout * horizontalLayout
- QPushButton * copyButton
- QSpacerItem * horizontalSpacer_6

12.55.1 Detailed Description

Definition at line 768 of file ui_mperfwidget.h.

12.55.2 Member Function Documentation

12.55.2.1 void Ui_MPerfWidget::retranslateUi (QWidget * *MPerfWidget*) [inline],[inherited]

Definition at line 682 of file ui_mperfwidget.h.

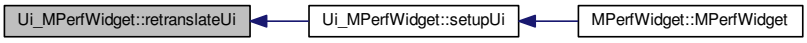
References Ui_MPerfWidget::aSnapBox, Ui_MPerfWidget::copyButton, Ui_MPerfWidget::eBox1, Ui_MPerfWidget::eBox2, Ui_MPerfWidget::eBox3, Ui_MPerfWidget::eBox4, Ui_MPerfWidget::eBox5, Ui_MPerfWidget::iButton0, Ui_MPerfWidget::iButton1, Ui_MPerfWidget::iButton2, Ui_MPerfWidget::iButton3, Ui_MPerfWidget::iButton4, Ui_MPerfWidget::iButton5, Ui_MPerfWidget::iButtonO, Ui_MPerfWidget::indicBox, Ui_MPerfWidget::label, Ui_MPerfWidget::label_11, Ui_MPerfWidget::label_12, Ui_MPerfWidget::label_13, Ui_MPerfWidget::label_14, Ui_MPerfWidget::label_15, Ui_MPerfWidget::label_2, Ui_MPerfWidget::label_25, Ui_MPerfWidget::label_3, Ui_MPerfWidget::label_4, Ui_MPerfWidget::label_5, Ui_MPerfWidget::label_6, Ui_MPerfWidget::label_7, Ui_MPerfWidget::label_8, Ui_MPerfWidget::lHeightBox, Ui_MPerfWidget::spinBox, and Ui_MPerfWidget::xyBox.

Referenced by Ui_MPerfWidget::setupUi().

```
683     {
684         MPerfWidget->setWindowTitle(QApplication::translate("MPerfWidget", "Form", Q_NULLPTR));
685         label->setText(QApplication::translate("MPerfWidget", "DPI", Q_NULLPTR));
686         label_25->setText(QApplication::translate("MPerfWidget", "Lift height:", Q_NULLPTR));
687         lHeightBox->clear();
688         lHeightBox->insertItems(0, QStringList()
689          << QApplication::translate("MPerfWidget", "Low", Q_NULLPTR)
690          << QApplication::translate("MPerfWidget", "Medium-low", Q_NULLPTR)
691          << QApplication::translate("MPerfWidget", "Medium", Q_NULLPTR)
692          << QApplication::translate("MPerfWidget", "Medium-high", Q_NULLPTR)
693          << QApplication::translate("MPerfWidget", "High", Q_NULLPTR)
694         );
695 #ifndef QT_NO_TOOLTIP
696         eBox2->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
    while moving up or down.", Q_NULLPTR));
697 #endif // QT_NO_TOOLTIP
698         eBox2->setText(QString());
699 #ifndef QT_NO_TOOLTIP
700         iButton4->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
701 #endif // QT_NO_TOOLTIP
702         iButton4->setText(QString());
703 #ifndef QT_NO_TOOLTIP
704         iButton5->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
705 #endif // QT_NO_TOOLTIP
706         iButton5->setText(QString());
707 #ifndef QT_NO_TOOLTIP
708         iButton3->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
709 #endif // QT_NO_TOOLTIP
710         iButton3->setText(QString());
711         label_12->setText(QApplication::translate("MPerfWidget", "X Value:", Q_NULLPTR));
712 #ifndef QT_NO_TOOLTIP
713         eBox5->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
    while moving up or down.", Q_NULLPTR));
714 #endif // QT_NO_TOOLTIP
715         eBox5->setText(QString());
716         label_4->setText(QApplication::translate("MPerfWidget", "Y Value:", Q_NULLPTR));
717 #ifndef QT_NO_TOOLTIP
718         iButton2->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
719 #endif // QT_NO_TOOLTIP
720         iButton2->setText(QString());
721 #ifndef QT_NO_TOOLTIP
722         iButtonO->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
723 #endif // QT_NO_TOOLTIP
724         iButtonO->setText(QString());
725 #ifndef QT_NO_TOOLTIP
726         iButton1->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
727 #endif // QT_NO_TOOLTIP
728         iButton1->setText(QString());
729 #ifndef QT_NO_TOOLTIP
730         iButton0->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
    colors at once", Q_NULLPTR));
731 #endif // QT_NO_TOOLTIP
732         iButton0->setText(QString());
733         label_3->setText(QApplication::translate("MPerfWidget", "You can assign other DPIs in the
    Binding tab.", Q_NULLPTR));
734         label_6->setText(QApplication::translate("MPerfWidget", "2:", Q_NULLPTR));
735         label_7->setText(QApplication::translate("MPerfWidget", "3:", Q_NULLPTR));
736         label_8->setText(QApplication::translate("MPerfWidget", "4:", Q_NULLPTR));
```

```
737         label_5->setText(QApplication::translate("MPerfWidget", "1:", Q_NULLPTR));
738         label_15->setText(QApplication::translate("MPerfWidget", "Other:", Q_NULLPTR));
739         label_13->setText(QApplication::translate("MPerfWidget", "5:", Q_NULLPTR));
740         label_14->setText(QApplication::translate("MPerfWidget", "Sniper:", Q_NULLPTR));
741 #ifndef QT_NO_TOOLTIP
742         eBox4->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
    while moving up or down.", Q_NULLPTR));
743 #endif // QT_NO_TOOLTIP
744         eBox4->setText(QString());
745         label_11->setText(QApplication::translate("MPerfWidget", "DPI Stage:", Q_NULLPTR));
746 #ifndef QT_NO_TOOLTIP
747         eBox3->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
    while moving up or down.", Q_NULLPTR));
748 #endif // QT_NO_TOOLTIP
749         eBox3->setText(QString());
750 #ifndef QT_NO_TOOLTIP
751         eBox1->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
    while moving up or down.", Q_NULLPTR));
752 #endif // QT_NO_TOOLTIP
753         eBox1->setText(QString());
754         label_2->setText(QApplication::translate("MPerfWidget", "Movement", Q_NULLPTR));
755         aSnapBox->setText(QApplication::translate("MPerfWidget", "Angle snap", Q_NULLPTR));
756         indicBox->setText(QApplication::translate("MPerfWidget", "Use DPI indicator:", Q_NULLPTR));
757 #ifndef QT_NO_TOOLTIP
758         spinBox->setToolTip(QApplication::translate("MPerfWidget", "Indicator intensity", Q_NULLPTR)
  );
759 #endif // QT_NO_TOOLTIP
760         spinBox->setSuffix(QApplication::translate("MPerfWidget", "%", Q_NULLPTR));
761         xyBox->setText(QApplication::translate("MPerfWidget", "Independent X/Y states", Q_NULLPTR));
762         copyButton->setText(QApplication::translate("MPerfWidget", "Copy to mode...", Q_NULLPTR))
  ;
763     } // retranslateUi
```

Here is the caller graph for this function:



12.55.2.2 void Ui_MPerfWidget::setupUi (QWidget * *MPerfWidget*) [inline],[inherited]

Definition at line 111 of file ui_mperfwidget.h.

References Ui_MPerfWidget::aSnapBox, Ui_MPerfWidget::copyButton, Ui_MPerfWidget::eBox1, Ui_MPerfWidget::eBox2, Ui_MPerfWidget::eBox3, Ui_MPerfWidget::eBox4, Ui_MPerfWidget::eBox5, Ui_MPerfWidget::gridLayout, Ui_MPerfWidget::horizontalLayout, Ui_MPerfWidget::horizontalLayout_2, Ui_MPerfWidget::horizontalLayout_4, Ui_MPerfWidget::horizontalSpacer, Ui_MPerfWidget::horizontalSpacer_2, Ui_MPerfWidget::horizontalSpacer_3, Ui_MPerfWidget::horizontalSpacer_4, Ui_MPerfWidget::horizontalSpacer_5, Ui_MPerfWidget::horizontalSpacer_6, Ui_MPerfWidget::horizontalSpacer_7, Ui_MPerfWidget::iButton0, Ui_MPerfWidget::iButton1, Ui_MPerfWidget::iButton2, Ui_MPerfWidget::iButton3, Ui_MPerfWidget::iButton4, Ui_MPerfWidget::iButton5, Ui_MPerfWidget::iButtonO, Ui_MPerfWidget::indicBox, Ui_MPerfWidget::label, Ui_MPerfWidget::label_11, Ui_MPerfWidget::label_12, Ui_MPerfWidget::label_13, Ui_MPerfWidget::label_14, Ui_MPerfWidget::label_15, Ui_MPerfWidget::label_2, Ui_MPerfWidget::label_25, Ui_MPerfWidget::label_3, Ui_MPerfWidget::label_4, Ui_MPerfWidget::label_5, Ui_MPerfWidget::label_6, Ui_MPerfWidget::label_7, Ui_MPerfWidget::label_8, Ui_MPerfWidget::lHeightBox, Ui_MPerfWidget::line, Ui_MPerfWidget::line_2, Ui_MPerfWidget::line_3, Ui_MPerfWidget::retranslateUi(), Ui_MPerfWidget::spinBox, Ui_MPerfWidget::verticalLayout, Ui_MPerfWidget::verticalSpacer, Ui_MPerfWidget::verticalSpacer_2, Ui_MPerfWidget::verticalSpacer_3, Ui_MPerfWidget::xBox0, Ui_MPerfWidget::xBox1, Ui_MPerfWidget::xBox2, Ui_MPerfWidget::xBox3, Ui_MPerfWidget::xBox4, Ui_MPerfWidget::xBox5, Ui_MPerfWidget::xSlider0, Ui_MPerfWidget::xSlider1, Ui_MPerfWidget::xSlider2, Ui_MPerfWidget::xSlider3, Ui_MPerfWidget::xSlider4, Ui_MPerfWidget::xSlider5, Ui_MPerfWidget::xyBox, Ui_MPerfWidget::yBox0, Ui_MPerfWidget::yBox1, Ui_MPerfWidget::yBox2, Ui_MPerfWidget::yBox3, Ui_MPerfWidget::yBox4, Ui_MPerfWidget::yBox5, Ui_MPerfWidget::ySlider0, Ui_MPerfWidget::ySlider1, Ui_MPerfWidget::ySlider2, Ui_MPerfWidget::ySlider3, Ui_MPerfWidget::ySlider4, and Ui_MPerfWidget::ySlider5.

Referenced by MPerfWidget::MPerfWidget().

```
112     {
113         if (MPerfWidget->objectName().isEmpty())
114             MPerfWidget->setObjectName(QStringLiteral("MPerfWidget"));
115         MPerfWidget->resize(932, 637);
116         verticalLayout = new QVBoxLayout(MPerfWidget);
117         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
118         label = new QLabel(MPerfWidget);
119         label->setObjectName(QStringLiteral("label"));
120         QFont font;
121         font.setBold(true);
122         font.setWeight(75);
123         label->setFont(font);
124 
125         verticalLayout->addWidget(label);
126 
127         line = new QFrame(MPerfWidget);
128         line->setObjectName(QStringLiteral("line"));
129         line->setFrameShape(QFrame::HLine);
130         line->setFrameShadow(QFrame::Sunken);
131 
132         verticalLayout->addWidget(line);
133 
134         gridLayout = new QGridLayout();
135         gridLayout->setObjectName(QStringLiteral("gridLayout"));
136         label_25 = new QLabel(MPerfWidget);
137         label_25->setObjectName(QStringLiteral("label_25"));
138         label_25->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
139 
140         gridLayout->addWidget(label_25, 16, 5, 1, 2);
141 
142         horizontalLayout_2 = new QHBoxLayout();
143         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
144         lHeightBox = new QComboBox(MPerfWidget);
145         lHeightBox->setObjectName(QStringLiteral("lHeightBox"));
146 
147         horizontalLayout_2->addWidget(lHeightBox);
148 
149         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
150 
151         horizontalLayout_2->addItem(horizontalSpacer_2);
152 
153 
154         gridLayout->addLayout(horizontalLayout_2, 16, 8, 1, 3);
155 
156         eBox2 = new QCheckBox(MPerfWidget);
157         eBox2->setObjectName(QStringLiteral("eBox2"));
158 
159         gridLayout->addWidget(eBox2, 7, 1, 1, 1);
160 
161         ySlider3 = new QSlider(MPerfWidget);
162         ySlider3->setObjectName(QStringLiteral("ySlider3"));
163         ySlider3->setMinimum(0);
164         ySlider3->setMaximum(1000);
165         ySlider3->setSingleStep(10);
166         ySlider3->setPageStep(50);
167         ySlider3->setValue(0);
168         ySlider3->setOrientation(Qt::Horizontal);
169 
170         gridLayout->addWidget(ySlider3, 8, 8, 1, 1);
171 
172         xSlider2 = new QSlider(MPerfWidget);
173         xSlider2->setObjectName(QStringLiteral("xSlider2"));
174         xSlider2->setMinimum(0);
175         xSlider2->setMaximum(1000);
176         xSlider2->setSingleStep(10);
177         xSlider2->setPageStep(50);
178         xSlider2->setValue(0);
179         xSlider2->setOrientation(Qt::Horizontal);
180 
181         gridLayout->addWidget(xSlider2, 7, 5, 1, 1);
182 
183         xBox3 = new QSpinBox(MPerfWidget);
184         xBox3->setObjectName(QStringLiteral("xBox3"));
185         xBox3->setMinimum(100);
186         xBox3->setMaximum(12000);
187         xBox3->setSingleStep(50);
188 
189         gridLayout->addWidget(xBox3, 8, 6, 1, 1);
190 
191         yBox0 = new QSpinBox(MPerfWidget);
192         yBox0->setObjectName(QStringLiteral("yBox0"));
193         yBox0->setMinimum(100);
194         yBox0->setMaximum(12000);
195         yBox0->setSingleStep(50);
196 
197         gridLayout->addWidget(yBox0, 11, 9, 1, 1);
```

```
198
199         ySlider0 = new QSlider(MPerfWidget);
200         ySlider0->setObjectName(QStringLiteral("ySlider0"));
201         ySlider0->setMinimum(0);
202         ySlider0->setMaximum(1000);
203         ySlider0->setSingleStep(10);
204         ySlider0->setPageStep(50);
205         ySlider0->setValue(0);
206         ySlider0->setOrientation(Qt::Horizontal);
207
208         gridLayout->addWidget(ySlider0, 11, 8, 1, 1);
209
210         iButton4 = new ColorButton(MPerfWidget);
211         iButton4->setObjectName(QStringLiteral("iButton4"));
212         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
213         sizePolicy.setHorizontalStretch(0);
214         sizePolicy.setVerticalStretch(0);
215         sizePolicy.setHeightForWidth(iButton4->sizePolicy().hasHeightForWidth());
216         iButton4->setSizePolicy(sizePolicy);
217         iButton4->setMinimumSize(QSize(0, 40));
218
219         gridLayout->addWidget(iButton4, 9, 2, 1, 1);
220
221         iButton5 = new ColorButton(MPerfWidget);
222         iButton5->setObjectName(QStringLiteral("iButton5"));
223         sizePolicy.setHeightForWidth(iButton5->sizePolicy().hasHeightForWidth());
224         iButton5->setSizePolicy(sizePolicy);
225         iButton5->setMinimumSize(QSize(0, 40));
226
227         gridLayout->addWidget(iButton5, 10, 2, 1, 1);
228
229         horizontalSpacer_4 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
230
231         gridLayout->addItem(horizontalSpacer_4, 4, 4, 1, 1);
232
233         verticalSpacer_3 = new QSpacerItem(20, 15, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
234
235         gridLayout->addItem(verticalSpacer_3, 3, 2, 1, 1);
236
237         horizontalSpacer_5 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
238
239         gridLayout->addItem(horizontalSpacer_5, 4, 7, 1, 1);
240
241         iButton3 = new ColorButton(MPerfWidget);
242         iButton3->setObjectName(QStringLiteral("iButton3"));
243         sizePolicy.setHeightForWidth(iButton3->sizePolicy().hasHeightForWidth());
244         iButton3->setSizePolicy(sizePolicy);
245         iButton3->setMinimumSize(QSize(0, 40));
246
247         gridLayout->addWidget(iButton3, 8, 2, 1, 1);
248
249         label_12 = new QLabel(MPerfWidget);
250         label_12->setObjectName(QStringLiteral("label_12"));
251         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Preferred);
252         sizePolicy1.setHorizontalStretch(0);
253         sizePolicy1.setVerticalStretch(0);
254         sizePolicy1.setHeightForWidth(label_12->sizePolicy().hasHeightForWidth());
255         label_12->setSizePolicy(sizePolicy1);
256
257         gridLayout->addWidget(label_12, 4, 5, 1, 1);
258
259         xBox4 = new QSpinBox(MPerfWidget);
260         xBox4->setObjectName(QStringLiteral("xBox4"));
261         xBox4->setMinimum(100);
262         xBox4->setMaximum(12000);
263         xBox4->setSingleStep(50);
264
265         gridLayout->addWidget(xBox4, 9, 6, 1, 1);
266
267         xBox0 = new QSpinBox(MPerfWidget);
268         xBox0->setObjectName(QStringLiteral("xBox0"));
269         xBox0->setMinimum(100);
270         xBox0->setMaximum(12000);
271         xBox0->setSingleStep(50);
272
273         gridLayout->addWidget(xBox0, 11, 6, 1, 1);
274
275         yBox5 = new QSpinBox(MPerfWidget);
276         yBox5->setObjectName(QStringLiteral("yBox5"));
277         yBox5->setMinimum(100);
278         yBox5->setMaximum(12000);
279         yBox5->setSingleStep(50);
280
281         gridLayout->addWidget(yBox5, 10, 9, 1, 1);
```

```
282
283         eBox5 = new QCheckBox(MPerfWidget);
284         eBox5->setObjectName(QStringLiteral("eBox5"));
285
286         gridLayout->addWidget(eBox5, 10, 1, 1, 1);
287
288         xSlider5 = new QSlider(MPerfWidget);
289         xSlider5->setObjectName(QStringLiteral("xSlider5"));
290         xSlider5->setMinimum(0);
291         xSlider5->setMaximum(1000);
292         xSlider5->setSingleStep(10);
293         xSlider5->setPageStep(50);
294         xSlider5->setValue(0);
295         xSlider5->setOrientation(Qt::Horizontal);
296
297         gridLayout->addWidget(xSlider5, 10, 5, 1, 1);
298
299         xBox1 = new QSpinBox(MPerfWidget);
300         xBox1->setObjectName(QStringLiteral("xBox1"));
301         sizePolicy.setHeightForWidth(xBox1->sizePolicy().hasHeightForWidth());
302         xBox1->setSizePolicy(sizePolicy);
303         xBox1->setMinimum(100);
304         xBox1->setMaximum(12000);
305         xBox1->setSingleStep(50);
306
307         gridLayout->addWidget(xBox1, 6, 6, 1, 1);
308
309         yBox2 = new QSpinBox(MPerfWidget);
310         yBox2->setObjectName(QStringLiteral("yBox2"));
311         yBox2->setMinimum(100);
312         yBox2->setMaximum(12000);
313         yBox2->setSingleStep(50);
314
315         gridLayout->addWidget(yBox2, 7, 9, 1, 1);
316
317         yBox3 = new QSpinBox(MPerfWidget);
318         yBox3->setObjectName(QStringLiteral("yBox3"));
319         yBox3->setMinimum(100);
320         yBox3->setMaximum(12000);
321         yBox3->setSingleStep(50);
322
323         gridLayout->addWidget(yBox3, 8, 9, 1, 1);
324
325         xSlider4 = new QSlider(MPerfWidget);
326         xSlider4->setObjectName(QStringLiteral("xSlider4"));
327         xSlider4->setMinimum(0);
328         xSlider4->setMaximum(1000);
329         xSlider4->setSingleStep(10);
330         xSlider4->setPageStep(50);
331         xSlider4->setValue(0);
332         xSlider4->setOrientation(Qt::Horizontal);
333
334         gridLayout->addWidget(xSlider4, 9, 5, 1, 1);
335
336         label_4 = new QLabel(MPerfWidget);
337         label_4->setObjectName(QStringLiteral("label_4"));
338         sizePolicy1.setHeightForWidth(label_4->sizePolicy().hasHeightForWidth());
339         label_4->setSizePolicy(sizePolicy1);
340
341         gridLayout->addWidget(label_4, 4, 8, 1, 1);
342
343         xSlider0 = new QSlider(MPerfWidget);
344         xSlider0->setObjectName(QStringLiteral("xSlider0"));
345         xSlider0->setMinimum(0);
346         xSlider0->setMaximum(1000);
347         xSlider0->setSingleStep(10);
348         xSlider0->setPageStep(50);
349         xSlider0->setValue(0);
350         xSlider0->setOrientation(Qt::Horizontal);
351
352         gridLayout->addWidget(xSlider0, 11, 5, 1, 1);
353
354         iButton2 = new ColorButton(MPerfWidget);
355         iButton2->setObjectName(QStringLiteral("iButton2"));
356         sizePolicy.setHeightForWidth(iButton2->sizePolicy().hasHeightForWidth());
357         iButton2->setSizePolicy(sizePolicy);
358         iButton2->setMinimumSize(QSize(0, 40));
359
360         gridLayout->addWidget(iButton2, 7, 2, 1, 1);
361
362         line_3 = new QFrame(MPerfWidget);
363         line_3->setObjectName(QStringLiteral("line_3"));
364         line_3->setFrameShape(QFrame::HLine);
365         line_3->setFrameShadow(QFrame::Sunken);
366
367         gridLayout->addWidget(line_3, 5, 0, 1, 11);
368
```

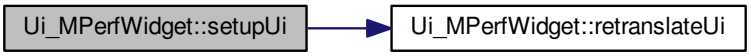
```
369         iButton0 = new ColorButton(MPerfWidget);
370         iButton0->setObjectName(QStringLiteral("iButton0"));
371         sizePolicy.setHeightForWidth(iButton0->sizePolicy().hasHeightForWidth());
372         iButton0->setSizePolicy(sizePolicy);
373         iButton0->setMinimumSize(QSize(0, 40));
374 
375         gridLayout->addWidget(iButton0, 12, 2, 1, 1);
376 
377         yBox1 = new QSpinBox(MPerfWidget);
378         yBox1->setObjectName(QStringLiteral("yBox1"));
379         sizePolicy.setHeightForWidth(yBox1->sizePolicy().hasHeightForWidth());
380         yBox1->setSizePolicy(sizePolicy);
381         yBox1->setMinimum(100);
382         yBox1->setMaximum(12000);
383         yBox1->setSingleStep(50);
384 
385         gridLayout->addWidget(yBox1, 6, 9, 1, 1);
386 
387         ySlider2 = new QSlider(MPerfWidget);
388         ySlider2->setObjectName(QStringLiteral("ySlider2"));
389         ySlider2->setMinimum(0);
390         ySlider2->setMaximum(1000);
391         ySlider2->setSingleStep(10);
392         ySlider2->setPageStep(50);
393         ySlider2->setValue(0);
394         ySlider2->setOrientation(Qt::Horizontal);
395 
396         gridLayout->addWidget(ySlider2, 7, 8, 1, 1);
397 
398         xBox2 = new QSpinBox(MPerfWidget);
399         xBox2->setObjectName(QStringLiteral("xBox2"));
400         xBox2->setMinimum(100);
401         xBox2->setMaximum(12000);
402         xBox2->setSingleStep(50);
403 
404         gridLayout->addWidget(xBox2, 7, 6, 1, 1);
405 
406         xSlider1 = new QSlider(MPerfWidget);
407         xSlider1->setObjectName(QStringLiteral("xSlider1"));
408         QSizePolicy sizePolicy2(QSizePolicy::Expanding, QSizePolicy::Fixed);
409         sizePolicy2.setHorizontalStretch(0);
410         sizePolicy2.setVerticalStretch(0);
411         sizePolicy2.setHeightForWidth(xSlider1->sizePolicy().hasHeightForWidth());
412         xSlider1->setSizePolicy(sizePolicy2);
413         xSlider1->setMinimum(0);
414         xSlider1->setMaximum(1000);
415         xSlider1->setSingleStep(10);
416         xSlider1->setPageStep(50);
417         xSlider1->setValue(0);
418         xSlider1->setOrientation(Qt::Horizontal);
419 
420         gridLayout->addWidget(xSlider1, 6, 5, 1, 1);
421 
422         horizontalSpacer = new QSpacerItem(5, 20, QSizePolicy::Fixed, QSizePolicy::Minimum)
    ;
423 
424         gridLayout->addItem(horizontalSpacer, 0, 0, 1, 1);
425 
426         yBox4 = new QSpinBox(MPerfWidget);
427         yBox4->setObjectName(QStringLiteral("yBox4"));
428         yBox4->setMinimum(100);
429         yBox4->setMaximum(12000);
430         yBox4->setSingleStep(50);
431 
432         gridLayout->addWidget(yBox4, 9, 9, 1, 1);
433 
434         iButton1 = new ColorButton(MPerfWidget);
435         iButton1->setObjectName(QStringLiteral("iButton1"));
436         sizePolicy.setHeightForWidth(iButton1->sizePolicy().hasHeightForWidth());
437         iButton1->setSizePolicy(sizePolicy);
438         iButton1->setMinimumSize(QSize(0, 40));
439 
440         gridLayout->addWidget(iButton1, 6, 2, 1, 1);
441 
442         ySlider4 = new QSlider(MPerfWidget);
443         ySlider4->setObjectName(QStringLiteral("ySlider4"));
444         ySlider4->setMinimum(0);
445         ySlider4->setMaximum(1000);
446         ySlider4->setSingleStep(10);
447         ySlider4->setPageStep(50);
448         ySlider4->setValue(0);
449         ySlider4->setOrientation(Qt::Horizontal);
450 
451         gridLayout->addWidget(ySlider4, 9, 8, 1, 1);
452 
453         ySlider1 = new QSlider(MPerfWidget);
454         ySlider1->setObjectName(QStringLiteral("ySlider1"));
```

```
455         ySlider1->setMinimum(0);
456         ySlider1->setMaximum(1000);
457         ySlider1->setSingleStep(10);
458         ySlider1->setPageStep(50);
459         ySlider1->setValue(0);
460         ySlider1->setOrientation(Qt::Horizontal);
461 
462         gridLayout->addWidget(ySlider1, 6, 8, 1, 1);
463 
464         iButton0 = new ColorButton(MPerfWidget);
465         iButton0->setObjectName(QStringLiteral("iButton0"));
466         sizePolicy.setHeightForWidth(iButton0->sizePolicy().hasHeightForWidth());
467         iButton0->setSizePolicy(sizePolicy);
468         iButton0->setMinimumSize(QSize(0, 40));
469 
470         gridLayout->addWidget(iButton0, 11, 2, 1, 1);
471 
472         xSlider3 = new QSlider(MPerfWidget);
473         xSlider3->setObjectName(QStringLiteral("xSlider3"));
474         xSlider3->setMinimum(0);
475         xSlider3->setMaximum(1000);
476         xSlider3->setSingleStep(10);
477         xSlider3->setPageStep(50);
478         xSlider3->setValue(0);
479         xSlider3->setOrientation(Qt::Horizontal);
480 
481         gridLayout->addWidget(xSlider3, 8, 5, 1, 1);
482 
483         xBox5 = new QSpinBox(MPerfWidget);
484         xBox5->setObjectName(QStringLiteral("xBox5"));
485         xBox5->setMinimum(100);
486         xBox5->setMaximum(12000);
487         xBox5->setSingleStep(50);
488 
489         gridLayout->addWidget(xBox5, 10, 6, 1, 1);
490 
491         label_3 = new QLabel(MPerfWidget);
492         label_3->setObjectName(QStringLiteral("label_3"));
493 
494         gridLayout->addWidget(label_3, 12, 5, 1, 5);
495 
496         label_6 = new QLabel(MPerfWidget);
497         label_6->setObjectName(QStringLiteral("label_6"));
498 
499         gridLayout->addWidget(label_6, 7, 3, 1, 1);
500 
501         ySlider5 = new QSlider(MPerfWidget);
502         ySlider5->setObjectName(QStringLiteral("ySlider5"));
503         ySlider5->setMinimum(0);
504         ySlider5->setMaximum(1000);
505         ySlider5->setSingleStep(10);
506         ySlider5->setPageStep(50);
507         ySlider5->setValue(0);
508         ySlider5->setOrientation(Qt::Horizontal);
509 
510         gridLayout->addWidget(ySlider5, 10, 8, 1, 1);
511 
512         label_7 = new QLabel(MPerfWidget);
513         label_7->setObjectName(QStringLiteral("label_7"));
514 
515         gridLayout->addWidget(label_7, 8, 3, 1, 1);
516 
517         label_8 = new QLabel(MPerfWidget);
518         label_8->setObjectName(QStringLiteral("label_8"));
519 
520         gridLayout->addWidget(label_8, 9, 3, 1, 1);
521 
522         label_5 = new QLabel(MPerfWidget);
523         label_5->setObjectName(QStringLiteral("label_5"));
524 
525         gridLayout->addWidget(label_5, 6, 3, 1, 1);
526 
527         label_15 = new QLabel(MPerfWidget);
528         label_15->setObjectName(QStringLiteral("label_15"));
529 
530         gridLayout->addWidget(label_15, 12, 3, 1, 1);
531 
532         label_13 = new QLabel(MPerfWidget);
533         label_13->setObjectName(QStringLiteral("label_13"));
534 
535         gridLayout->addWidget(label_13, 10, 3, 1, 1);
536 
537         label_14 = new QLabel(MPerfWidget);
538         label_14->setObjectName(QStringLiteral("label_14"));
539 
540         gridLayout->addWidget(label_14, 11, 3, 1, 1);
541 
```

```
542         eBox4 = new QCheckBox(MPerfWidget);
543         eBox4->setObjectName(QStringLiteral("eBox4"));
544 
545         gridLayout->addWidget(eBox4, 9, 1, 1, 1);
546 
547         label_11 = new QLabel(MPerfWidget);
548         label_11->setObjectName(QStringLiteral("label_11"));
549 
550         gridLayout->addWidget(label_11, 4, 2, 1, 2);
551 
552         eBox3 = new QCheckBox(MPerfWidget);
553         eBox3->setObjectName(QStringLiteral("eBox3"));
554 
555         gridLayout->addWidget(eBox3, 8, 1, 1, 1);
556 
557         eBox1 = new QCheckBox(MPerfWidget);
558         eBox1->setObjectName(QStringLiteral("eBox1"));
559 
560         gridLayout->addWidget(eBox1, 6, 1, 1, 1);
561 
562         label_2 = new QLabel(MPerfWidget);
563         label_2->setObjectName(QStringLiteral("label_2"));
564         label_2->setFont(font);
565 
566         gridLayout->addWidget(label_2, 14, 0, 1, 10);
567 
568         verticalSpacer = new QSpacerItem(20, 30, QSizePolicy::Minimum, QSizePolicy::Fixed);
569 
570         gridLayout->addItem(verticalSpacer, 13, 2, 1, 1);
571 
572         aSnapBox = new QCheckBox(MPerfWidget);
573         aSnapBox->setObjectName(QStringLiteral("aSnapBox"));
574 
575         gridLayout->addWidget(aSnapBox, 16, 1, 1, 4);
576 
577         line_2 = new QFrame(MPerfWidget);
578         line_2->setObjectName(QStringLiteral("line_2"));
579         line_2->setFrameShape(QFrame::HLine);
580         line_2->setFrameShadow(QFrame::Sunken);
581 
582         gridLayout->addWidget(line_2, 15, 0, 1, 10);
583 
584         horizontalLayout_4 = new QHBoxLayout();
585         horizontalLayout_4->setObjectName(QStringLiteral("horizontalLayout_4"));
586         indicBox = new QCheckBox(MPerfWidget);
587         indicBox->setObjectName(QStringLiteral("indicBox"));
588 
589         horizontalLayout_4->addWidget(indicBox);
590 
591         horizontalSpacer_3 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
592 
593         horizontalLayout_4->addItem(horizontalSpacer_3);
594 
595         spinBox = new QSpinBox(MPerfWidget);
596         spinBox->setObjectName(QStringLiteral("spinBox"));
597         spinBox->setMaximum(100);
598         spinBox->setValue(100);
599 
600         horizontalLayout_4->addWidget(spinBox);
601 
602         horizontalSpacer_7 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
603 
604         horizontalLayout_4->addItem(horizontalSpacer_7);
605 
606 
607         gridLayout->addLayout(horizontalLayout_4, 0, 1, 1, 9);
608 
609         xyBox = new QCheckBox(MPerfWidget);
610         xyBox->setObjectName(QStringLiteral("xyBox"));
611 
612         gridLayout->addWidget(xyBox, 2, 1, 1, 9);
613 
614 
615         verticalLayout->addLayout(gridLayout);
616 
617         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
618 
619         verticalLayout->addItem(verticalSpacer_2);
620 
621         horizontalLayout = new QHBoxLayout();
622         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
623         copyButton = new QPushButton(MPerfWidget);
624         copyButton->setObjectName(QStringLiteral("copyButton"));
625 
```

```
626         horizontalLayout->addWidget(copyButton);
627 
628         horizontalSpacer_6 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
629 
630         horizontalLayout->addItem(horizontalSpacer_6);
631 
632 
633         verticalLayout->addLayout(horizontalLayout);
634 
635         QWidget::setTabOrder(indicBox, spinBox);
636         QWidget::setTabOrder(spinBox, xyBox);
637         QWidget::setTabOrder(xyBox, eBox1);
638         QWidget::setTabOrder(eBox1, eBox2);
639         QWidget::setTabOrder(eBox2, eBox3);
640         QWidget::setTabOrder(eBox3, eBox4);
641         QWidget::setTabOrder(eBox4, eBox5);
642         QWidget::setTabOrder(eBox5, iButton1);
643         QWidget::setTabOrder(iButton1, iButton2);
644         QWidget::setTabOrder(iButton2, iButton3);
645         QWidget::setTabOrder(iButton3, iButton4);
646         QWidget::setTabOrder(iButton4, iButton5);
647         QWidget::setTabOrder(iButton5, iButton0);
648         QWidget::setTabOrder(iButton0, iButtonO);
649         QWidget::setTabOrder(iButtonO, xSlider1);
650         QWidget::setTabOrder(xSlider1, xSlider2);
651         QWidget::setTabOrder(xSlider2, xSlider3);
652         QWidget::setTabOrder(xSlider3, xSlider4);
653         QWidget::setTabOrder(xSlider4, xSlider5);
654         QWidget::setTabOrder(xSlider5, xSlider0);
655         QWidget::setTabOrder(xSlider0, xBox1);
656         QWidget::setTabOrder(xBox1, xBox2);
657         QWidget::setTabOrder(xBox2, xBox3);
658         QWidget::setTabOrder(xBox3, xBox4);
659         QWidget::setTabOrder(xBox4, xBox5);
660         QWidget::setTabOrder(xBox5, xBox0);
661         QWidget::setTabOrder(xBox0, ySlider1);
662         QWidget::setTabOrder(ySlider1, ySlider2);
663         QWidget::setTabOrder(ySlider2, ySlider3);
664         QWidget::setTabOrder(ySlider3, ySlider4);
665         QWidget::setTabOrder(ySlider4, ySlider5);
666         QWidget::setTabOrder(ySlider5, ySlider0);
667         QWidget::setTabOrder(ySlider0, yBox1);
668         QWidget::setTabOrder(yBox1, yBox2);
669         QWidget::setTabOrder(yBox2, yBox3);
670         QWidget::setTabOrder(yBox3, yBox4);
671         QWidget::setTabOrder(yBox4, yBox5);
672         QWidget::setTabOrder(yBox5, yBox0);
673         QWidget::setTabOrder(yBox0, aSnapBox);
674         QWidget::setTabOrder(aSnapBox, lHeightBox);
675         QWidget::setTabOrder(lHeightBox, copyButton);
676 
677         retranslateUi(MPerfWidget);
678 
679         QMetaObject::connectSlotsByName(MPerfWidget);
680     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.55.3 Field Documentation

12.55.3.1 QCheckBox* Ui_MPerfWidget::aSnapBox [inherited]

Definition at line 98 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), MPerfWidget::setPerf(), and Ui_MPerfWidget::setupUi().

12.55.3.2 QPushButton* Ui_MPerfWidget::copyButton [inherited]

Definition at line 108 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.3 QCheckBox* Ui_MPerfWidget::eBox1 [inherited]

Definition at line 95 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.4 QCheckBox* Ui_MPerfWidget::eBox2 [inherited]

Definition at line 44 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.5 QCheckBox* Ui_MPerfWidget::eBox3 [inherited]

Definition at line 94 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.6 QCheckBox* Ui_MPerfWidget::eBox4 [inherited]

Definition at line 92 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.7 QCheckBox* Ui_MPerfWidget::eBox5 [inherited]

Definition at line 60 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.8 QGridLayout* Ui_MPerfWidget::gridLayout [inherited]

Definition at line 39 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.9 QHBoxLayout* Ui_MPerfWidget::horizontalLayout [inherited]

Definition at line 107 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.10 QHBoxLayout* Ui_MPerfWidget::horizontalLayout_2 [inherited]

Definition at line 41 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.11 QHBoxLayout* Ui_MPerfWidget::horizontalLayout_4 [inherited]

Definition at line 100 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.12 QSpacerItem* Ui_MPerfWidget::horizontalSpacer [inherited]

Definition at line 75 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.13 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_2 [inherited]

Definition at line 43 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.14 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_3 [inherited]

Definition at line 102 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.15 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_4 [inherited]

Definition at line 52 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.16 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_5 [inherited]

Definition at line 54 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.17 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_6 [inherited]

Definition at line 109 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.18 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_7 [inherited]

Definition at line 104 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.19 ColorButton* Ui_MPerfWidget::iButton0 [inherited]

Definition at line 80 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.20 ColorButton* Ui_MPerfWidget::iButton1 [inherited]

Definition at line 77 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.21 ColorButton* Ui_MPerfWidget::iButton2 [inherited]

Definition at line 68 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.22 ColorButton* Ui_MPerfWidget::iButton3 [inherited]

Definition at line 55 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.23 ColorButton* Ui_MPerfWidget::iButton4 [inherited]

Definition at line 50 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.24 ColorButton* Ui_MPerfWidget::iButton5 [inherited]

Definition at line 51 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.25 ColorButton* Ui_MPerfWidget::iButtonO [inherited]

Definition at line 70 of file ui_mperfwidget.h.

Referenced by MPerfWidget::colorChanged(), MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), MPerfWidget::setPerf(), and Ui_MPerfWidget::setupUi().

12.55.3.26 QCheckBox* Ui_MPerfWidget::indicBox [inherited]

Definition at line 101 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), MPerfWidget::setPerf(), and Ui_MPerfWidget::setupUi().

12.55.3.27 QLabel* Ui_MPerfWidget::label [inherited]

Definition at line 37 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.28 QLabel* Ui_MPerfWidget::label_11 [inherited]

Definition at line 93 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.29 QLabel* Ui_MPerfWidget::label_12 [inherited]

Definition at line 56 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.30 QLabel* Ui_MPerfWidget::label_13 [inherited]

Definition at line 90 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.31 QLabel* Ui_MPerfWidget::label_14 [inherited]

Definition at line 91 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.32 QLabel* Ui_MPerfWidget::label_15 [inherited]

Definition at line 89 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.33 QLabel* Ui_MPerfWidget::label_2 [inherited]

Definition at line 96 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.34 QLabel* Ui_MPerfWidget::label_25 [inherited]

Definition at line 40 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.35 QLabel* Ui_MPerfWidget::label_3 [inherited]

Definition at line 83 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.36 QLabel* Ui_MPerfWidget::label_4 [inherited]

Definition at line 66 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.37 QLabel* Ui_MPerfWidget::label_5 [inherited]

Definition at line 88 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.38 QLabel* Ui_MPerfWidget::label_6 [inherited]

Definition at line 84 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.39 QLabel* Ui_MPerfWidget::label_7 [inherited]

Definition at line 86 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.40 QLabel* Ui_MPerfWidget::label_8 [inherited]

Definition at line 87 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), and Ui_MPerfWidget::setupUi().

12.55.3.41 QComboBox* Ui_MPerfWidget::lHeightBox [inherited]

Definition at line 42 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), MPerfWidget::setPerf(), and Ui_MPerfWidget::setupUi().

12.55.3.42 QFrame* Ui_MPerfWidget::line [inherited]

Definition at line 38 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.43 QFrame* Ui_MPerfWidget::line_2 [inherited]

Definition at line 99 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.44 QFrame* Ui_MPerfWidget::line_3 [inherited]

Definition at line 69 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.45 QSpinBox* Ui_MPerfWidget::spinBox [inherited]

Definition at line 103 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::retranslateUi(), MPerfWidget::setPerf(), and Ui_MPerfWidget::setupUi().

12.55.3.46 QVBoxLayout* Ui_MPerfWidget::verticalLayout [inherited]

Definition at line 36 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.47 QSpacerItem* Ui_MPerfWidget::verticalSpacer [inherited]

Definition at line 97 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.48 QSpacerItem* Ui_MPerfWidget::verticalSpacer_2 [inherited]

Definition at line 106 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.49 QSpacerItem* Ui_MPerfWidget::verticalSpacer_3 [inherited]

Definition at line 53 of file ui_mperfwidget.h.

Referenced by Ui_MPerfWidget::setupUi().

12.55.3.50 QSpinBox* Ui_MPerfWidget::xBox0 [inherited]

Definition at line 58 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.51 QSpinBox* Ui_MPerfWidget::xBox1 [inherited]

Definition at line 62 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.52 QSpinBox* Ui_MPerfWidget::xBox2 [inherited]

Definition at line 73 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.53 QSpinBox* Ui_MPerfWidget::xBox3 [inherited]

Definition at line 47 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.54 QSpinBox* Ui_MPerfWidget::xBox4 [inherited]

Definition at line 57 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.55 QSpinBox* Ui_MPerfWidget::xBox5 [inherited]

Definition at line 82 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.56 QSlider* Ui_MPerfWidget::xSlider0 [inherited]

Definition at line 67 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.57 QSlider* Ui_MPerfWidget::xSlider1 [inherited]

Definition at line 74 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.58 QSlider* Ui_MPerfWidget::xSlider2 [inherited]

Definition at line 46 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.59 QSlider* Ui_MPerfWidget::xSlider3 [inherited]

Definition at line 81 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.60 QSlider* Ui_MPerfWidget::xSlider4 [inherited]

Definition at line 65 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.61 QSlider* Ui_MPerfWidget::xSlider5 [inherited]

Definition at line 61 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.62 QCheckBox* Ui_MPerfWidget::xyBox [inherited]

Definition at line 105 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), Ui_MPerfWidget::retranslateUi(), Ui_MPerfWidget::setupUi(), and MPerfWidget::sliderYMoved().

12.55.3.63 QSpinBox* Ui_MPerfWidget::yBox0 [inherited]

Definition at line 48 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.64 QSpinBox* Ui_MPerfWidget::yBox1 [inherited]

Definition at line 71 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.65 QSpinBox* Ui_MPerfWidget::yBox2 [inherited]

Definition at line 63 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.66 QSpinBox* Ui_MPerfWidget::yBox3 [inherited]

Definition at line 64 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.67 QSpinBox* Ui_MPerfWidget::yBox4 [inherited]

Definition at line 76 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.68 QSpinBox* Ui_MPerfWidget::yBox5 [inherited]

Definition at line 59 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.69 QSlider* Ui_MPerfWidget::ySlider0 [inherited]

Definition at line 49 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.70 QSlider* Ui_MPerfWidget::ySlider1 [inherited]

Definition at line 79 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.71 QSlider* Ui_MPerfWidget::ySlider2 [inherited]

Definition at line 72 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.72 QSlider* Ui_MPerfWidget::ySlider3 [inherited]

Definition at line 45 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.73 QSlider* Ui_MPerfWidget::ySlider4 [inherited]

Definition at line 78 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

12.55.3.74 QSlider* Ui_MPerfWidget::ySlider5 [inherited]

Definition at line 85 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and Ui_MPerfWidget::setupUi().

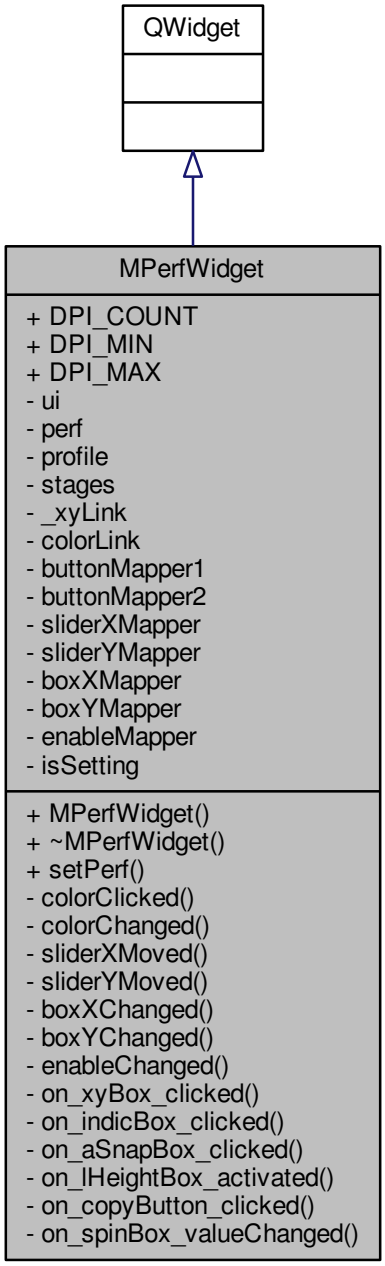
The documentation for this class was generated from the following file:

- src/ckb/ui_mperfwidget.h

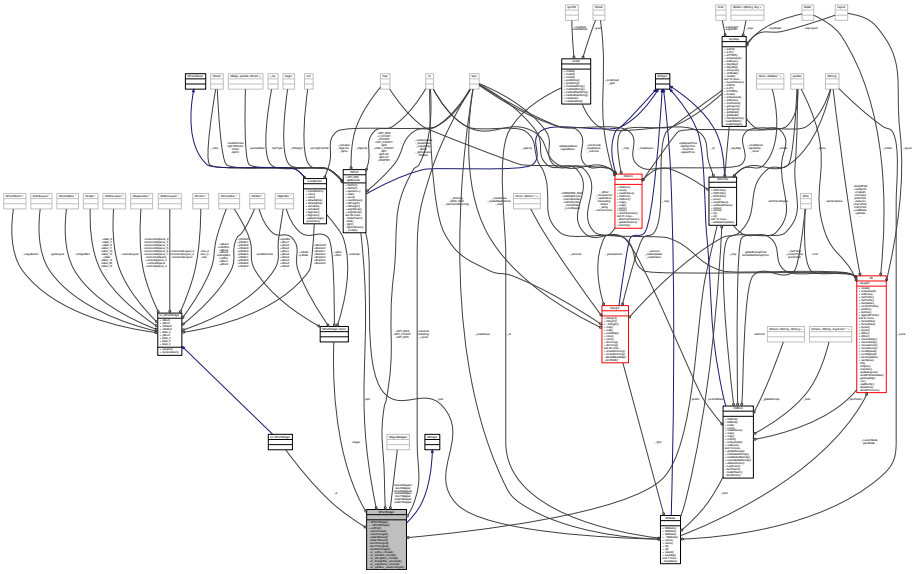
12.56 MPerfWidget Class Reference

`#include <src/ckb/mperfwidget.h>`

Inheritance diagram for MPerfWidget:



Collaboration diagram for MPerfWidget:



Data Structures

- struct DpiUi

Public Member Functions

- MPerfWidget (QWidget *parent=0)
- ~MPerfWidget ()
- void setPerf (KbPerf *newPerf, KbProfile *newProfile)

Static Public Attributes

- static const int DPI_COUNT = KbPerf::DPI_COUNT
- static const int DPI_MIN = KbPerf::DPI_MIN
- static const int DPI_MAX = KbPerf::DPI_MAX

Private Slots

- void colorClicked (int index)
- void colorChanged (int index)
- void sliderXMoved (int index)
- void sliderYMoved (int index)
- void boxXChanged (int index)
- void boxYChanged (int index)
- void enableChanged (int index)
- void on_xyBox_clicked (bool checked)
- void on_indicBox_clicked (bool checked)
- void on_aSnapBox_clicked (bool checked)
- void on_lHeightBox_activated (int index)
- void on_copyButton_clicked ()
- void on_spinBox_valueChanged (int arg1)

Private Attributes

- Ui::MPerfWidget * ui
- KbPerf * perf
- KbProfile * profile
- DpiUi stages [DPI_COUNT]
- bool _xyLink
- bool colorLink
- QSignalMapper buttonMapper1
- QSignalMapper buttonMapper2
- QSignalMapper sliderXMapper
- QSignalMapper sliderYMapper
- QSignalMapper boxXMapper
- QSignalMapper boxYMapper
- QSignalMapper enableMapper
- bool isSetting

12.56.1 Detailed Description

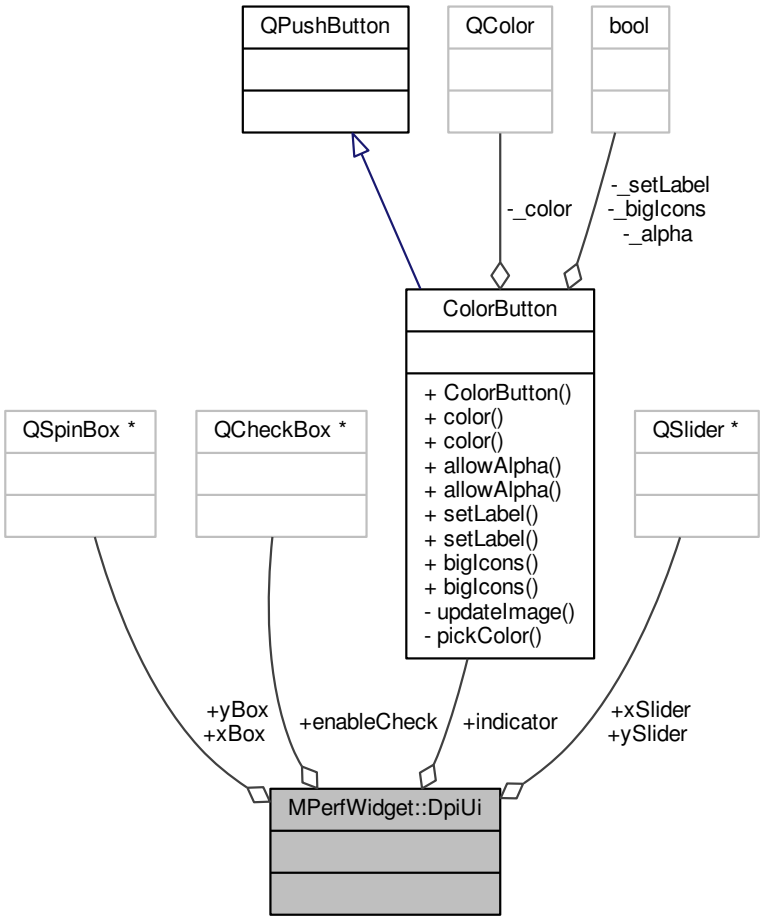
Definition at line 18 of file mperfwidget.h.

12.56.2 Data Structure Documentation

12.56.2.1 struct MPerfWidget::DpiUi

Definition at line 37 of file mperfwidget.h.

Collaboration diagram for MPerfWidget::DpiUi:



Data Fields

QCheckBox *	enableCheck	
ColorButton *	indicator	
QSpinBox *	xBox	
QSlider *	xSlider	
QSpinBox *	yBox	
QSlider *	ySlider	

12.56.3 Constructor & Destructor Documentation

12.56.3.1 MPerfWidget::MPerfWidget (QWidget * *parent* = 0) [explicit]

Definition at line 8 of file mperfwidget.cpp.

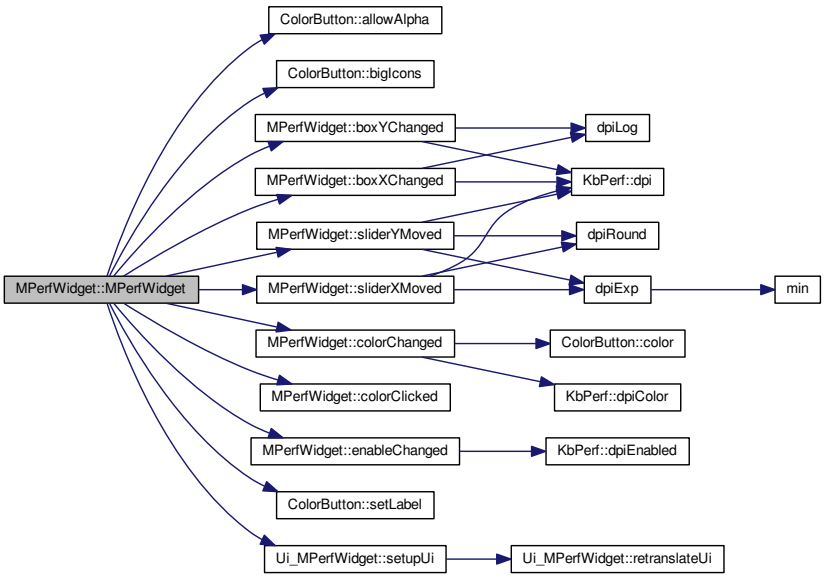
References _xyLink, ColorButton::allowAlpha(), ColorButton::bigIcons(), boxXChanged(), boxXMapper, boxYChanged(), boxYMapper, buttonMapper1, buttonMapper2, colorChanged(), colorClicked(), DPI_COUNT, Ui_MPerfWidget::eBox1, Ui_MPerfWidget::eBox2, Ui_MPerfWidget::eBox3, Ui_MPerfWidget::eBox4, Ui_MPerfWidget::eBox5, enableChanged(), MPerfWidget::DpiUi::enableCheck, enableMapper, Ui_MPerfWidget::iButton0,

Ui_MPerfWidget::iButton1, Ui_MPerfWidget::iButton2, Ui_MPerfWidget::iButton3, Ui_MPerfWidget::iButton4, Ui_MPerfWidget::iButton5, Ui_MPerfWidget::iButtonO, MPerfWidget::DpiUi::indicator, ColorButton::setLabel(), Ui_MPerfWidget::setupUi(), sliderXMapper, sliderXMoved(), sliderYMapper, sliderYMoved(), stages, ui, MPerfWidget::DpiUi::xBox, Ui_MPerfWidget::xBox0, Ui_MPerfWidget::xBox1, Ui_MPerfWidget::xBox2, Ui_MPerfWidget::xBox3, Ui_MPerfWidget::xBox4, Ui_MPerfWidget::xBox5, MPerfWidget::DpiUi::xSlider, Ui_MPerfWidget::xSlider0, Ui_MPerfWidget::xSlider1, Ui_MPerfWidget::xSlider2, Ui_MPerfWidget::xSlider3, Ui_MPerfWidget::xSlider4, Ui_MPerfWidget::xSlider5, Ui_MPerfWidget::xyBox, MPerfWidget::DpiUi::yBox, Ui_MPerfWidget::yBox0, Ui_MPerfWidget::yBox1, Ui_MPerfWidget::yBox2, Ui_MPerfWidget::yBox3, Ui_MPerfWidget::yBox4, Ui_MPerfWidget::yBox5, MPerfWidget::DpiUi::ySlider, Ui_MPerfWidget::ySlider0, Ui_MPerfWidget::ySlider1, Ui_MPerfWidget::ySlider2, Ui_MPerfWidget::ySlider3, Ui_MPerfWidget::ySlider4, and Ui_MPerfWidget::ySlider5.

```
8                                       :
9     QWidget(parent), ui(new Ui::MPerfWidget), perf(0),
      profile(0), _xyLink(!CkbSettings::get(xyLinkPath).toBool()),
      colorLink(false), isSetting(false) {
10     ui->setupUi(this);
11     ui->xyBox->setChecked(!_xyLink);
12     // Set up DPI stages
13     stages[0].indicator = ui->iButton0; stages[0].
      xSlider = ui->xSlider0; stages[0].ySlider = ui->
      ySlider0; stages[0].xBox = ui->xBox0; stages[0].
      yBox = ui->yBox0; stages[0].enableCheck = 0;
14     stages[1].indicator = ui->iButton1; stages[1].
      xSlider = ui->xSlider1; stages[1].ySlider = ui->
      ySlider1; stages[1].xBox = ui->xBox1; stages[1].
      yBox = ui->yBox1; stages[1].enableCheck = ui->
      eBox1;
15     stages[2].indicator = ui->iButton2; stages[2].
      xSlider = ui->xSlider2; stages[2].ySlider = ui->
      ySlider2; stages[2].xBox = ui->xBox2; stages[2].
      yBox = ui->yBox2; stages[2].enableCheck = ui->
      eBox2;
16     stages[3].indicator = ui->iButton3; stages[3].
      xSlider = ui->xSlider3; stages[3].ySlider = ui->
      ySlider3; stages[3].xBox = ui->xBox3; stages[3].
      yBox = ui->yBox3; stages[3].enableCheck = ui->
      eBox3;
17     stages[4].indicator = ui->iButton4; stages[4].
      xSlider = ui->xSlider4; stages[4].ySlider = ui->
      ySlider4; stages[4].xBox = ui->xBox4; stages[4].
      yBox = ui->yBox4; stages[4].enableCheck = ui->
      eBox4;
18     stages[5].indicator = ui->iButton5; stages[5].
      xSlider = ui->xSlider5; stages[5].ySlider = ui->
      ySlider5; stages[5].xBox = ui->xBox5; stages[5].
      yBox = ui->yBox5; stages[5].enableCheck = ui->
      eBox5;
19     ui->iButtonO->setLabel(false);
20     ui->iButtonO->bigIcons(true);
21     ui->iButtonO->allowAlpha(true);
22     for(int i = 0; i < DPI_COUNT; i++){
23         stages[i].indicator->setLabel(false);
24         stages[i].indicator->bigIcons(true);
25         stages[i].indicator->allowAlpha(true);
26         // Map signals
27         connect(stages[i].indicator, SIGNAL(clicked(bool)), &buttonMapper1, SLOT(map()))
      ;
28         connect(stages[i].indicator, SIGNAL(colorChanged(QColor)), &
      buttonMapper2, SLOT(map()));
29         connect(stages[i].xSlider, SIGNAL(valueChanged(int)), &
      sliderXMapper, SLOT(map()));
30         connect(stages[i].ySlider, SIGNAL(valueChanged(int)), &
      sliderYMapper, SLOT(map()));
31         connect(stages[i].xBox, SIGNAL(valueChanged(int)), &boxXMapper, SLOT(map()));
32         connect(stages[i].yBox, SIGNAL(valueChanged(int)), &boxYMapper, SLOT(map()));
33         if(stages[i].enableCheck)
34             // Sniper has no enable
35             connect(stages[i].enableCheck, SIGNAL(stateChanged(int)), &
      enableMapper, SLOT(map()));
36         // Set names
37         buttonMapper1.setMapping(stages[i].indicator, i);
38         buttonMapper2.setMapping(stages[i].indicator, i);
39         sliderXMapper.setMapping(stages[i].xSlider, i);
40         sliderYMapper.setMapping(stages[i].ySlider, i);
41         boxXMapper.setMapping(stages[i].xBox, i);
42         boxYMapper.setMapping(stages[i].yBox, i);
43         if(stages[i].enableCheck)
44             enableMapper.setMapping(stages[i].enableCheck, i);
45     }
46     // Connect to slots
47     connect(&buttonMapper1, SIGNAL(mapped(int)), this, SLOT(
      colorClicked(int)));
```

```
48     connect(&buttonMapper2, SIGNAL(mapped(int)), this, SLOT(
       colorChanged(int)));
49     connect(&sliderXMapper, SIGNAL(mapped(int)), this, SLOT(
       sliderXMoved(int)));
50     connect(&sliderYMapper, SIGNAL(mapped(int)), this, SLOT(
       sliderYMoved(int)));
51     connect(&boxXMapper, SIGNAL(mapped(int)), this, SLOT(boxXChanged(int)));
52     connect(&boxYMapper, SIGNAL(mapped(int)), this, SLOT(boxYChanged(int)));
53     connect(&enableMapper, SIGNAL(mapped(int)), this, SLOT(
       enableChanged(int)));
54 }
```

Here is the call graph for this function:



12.56.3.2 MPerfWidget::~MPerfWidget ( )

Definition at line 56 of file mperfwidget.cpp.

References ui.

```
56                          {
57     delete ui;
58 }
```

12.56.4 Member Function Documentation

12.56.4.1 void MPerfWidget::boxXChanged (int *index*) [private], [slot]

Definition at line 157 of file mperfwidget.cpp.

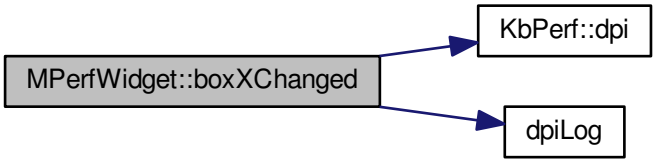
References _xyLink, KbPerf::dpi(), dpiLog(), perf, SET_END, SET_START, stages, MPerfWidget::DpiUi::xBox, MPerfWidget::DpiUi::xSlider, and MPerfWidget::DpiUi::yBox.

Referenced by MPerfWidget().

```
157                                     {
158     SET_START;
159     QSpinBox* box = stages[index].xBox;
160     QSlider* slider = stages[index].xSlider;
```

```
161     int value = box->value();
162     slider->setValue(dpiLog(value, slider->minimum(), slider->maximum()));
163     if(_xyLink)
164         perf->dpi(index, value);
165     else
166         perf->dpi(index, QPoint(value, perf->dpi(index).y()));
167     SET_END;
168     if(_xyLink)
169         stages[index].yBox->setValue(value);
170 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.56.4.2 void MPerfWidget::boxYChanged (int *index*) [private],[slot]

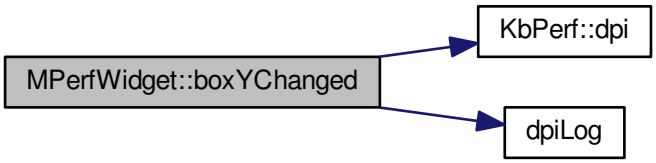
Definition at line 172 of file mperfwidget.cpp.

References _xyLink, KbPerf::dpi(), dpiLog(), perf, SET_END, SET_START, stages, MPerfWidget::DpiUi::xBox, MPerfWidget::DpiUi::yBox, and MPerfWidget::DpiUi::ySlider.

Referenced by MPerfWidget().

```
172                                     {
173     SET_START;
174     QSpinBox* box = stages[index].yBox;
175     QSlider* slider = stages[index].ySlider;
176     int value = box->value();
177     slider->setValue(dpiLog(value, slider->minimum(), slider->maximum()));
178     if(_xyLink)
179         perf->dpi(index, value);
180     else
181         perf->dpi(index, QPoint(perf->dpi(index).x(), value));
182     SET_END;
183     if(_xyLink)
184         stages[index].xBox->setValue(value);
185 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.56.4.3 void MPerfWidget::colorChanged (int *index*) [private],[slot]

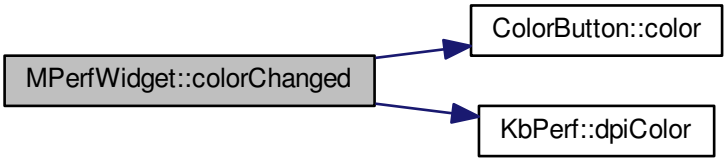
Definition at line 86 of file mperfwidget.cpp.

References ColorButton::color(), colorLink, DPI_COUNT, KbPerf::dpiColor(), Ui_MPerfWidget::iButtonO, MPerfWidget::DpiUi::indicator, KbPerf::OTHER, perf, stages, and ui.

Referenced by MPerfWidget().

```
86                                          {
87     QColor color = stages[index].indicator->color();
88     if(colorLink){
89         // Alt was held down - set all
90         for(int i = 0; i < DPI_COUNT; i++){
91             stages[i].indicator->color(color);
92             perf->dpiColor(i, color);
93         }
94         ui->iButtonO->color(color);
95         perf->dpiColor(KbPerf::OTHER, color);
96     } else {
97         // Set one
98         perf->dpiColor(index, color);
99     }
100     colorLink = false;
101 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.56.4.4 void MPerfWidget::colorClicked (int *index*) [private],[slot]

Definition at line 81 of file mperfwidget.cpp.

References colorLink.

Referenced by MPerfWidget().

```
81                                      {
82     // Set all colors at once if Alt is held down
83     colorLink = !!(qApp->keyboardModifiers() & Qt::AltModifier);
84 }
```

Here is the caller graph for this function:

12.56.4.5 void MPerfWidget::enableChanged (int *index*) [private],[slot]

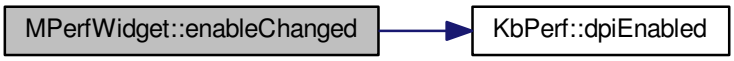
Definition at line 187 of file mperfwidget.cpp.

References KbPerf::dpiEnabled(), perf, and stages.

Referenced by MPerfWidget().

```
187                                         {
188     perf->dpiEnabled(index, stages[index].enableCheck->isChecked());
189 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

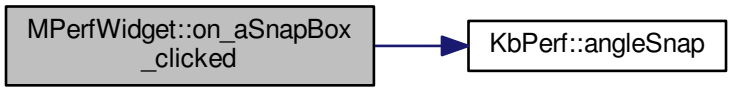
12.56.4.6 void MPerfWidget::on_aSnapBox_clicked (bool *checked*) [private], [slot]

Definition at line 201 of file mperfwidget.cpp.

References KbPerf::angleSnap(), and perf.

```
201                                                 {
202     perf->angleSnap(checked);
203 }
```

Here is the call graph for this function:



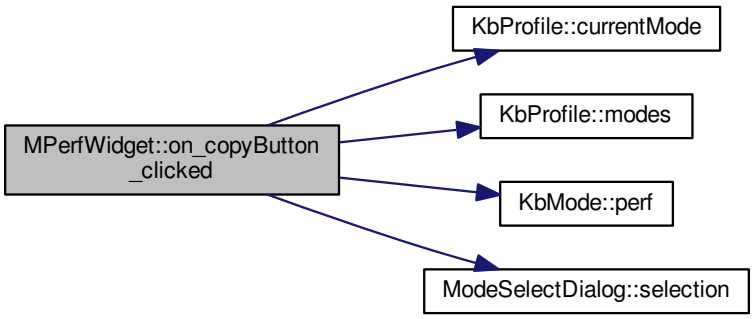
12.56.4.7 void MPerfWidget::on_copyButton_clicked ( ) [private], [slot]

Definition at line 209 of file mperfwidget.cpp.

References KbProfile::currentMode(), KbProfile::modes(), perf, KbMode::perf(), profile, and ModeSelectDialog::selection().

```
209                                     {
210     ModeSelectDialog dialog(this, profile->currentMode(),
      profile->modes(), "Copy performance settings to:");
211     if(dialog.exec() != QDialog::Accepted)
212         return;
213     QList<KbMode*> selectedModes = dialog.selection();
214     foreach(KbMode* mode, selectedModes){
215         *mode->perf() = *perf;
216     }
217 }
```

Here is the call graph for this function:



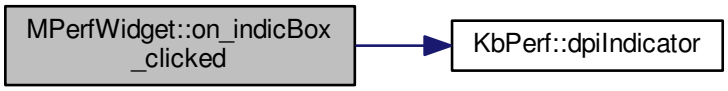
12.56.4.8 void MPerfWidget::on_indicBox_clicked (bool *checked*) [private],[slot]

Definition at line 197 of file mperfwidget.cpp.

References KbPerf::dpiIndicator(), and perf.

```
197                                                 {
198     perf->dpiIndicator(checked);
199 }
```

Here is the call graph for this function:



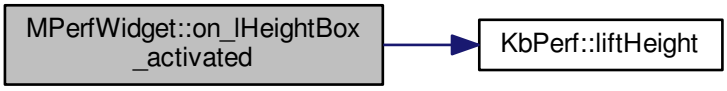
12.56.4.9 void MPerfWidget::on_lHeightBox_activated (int *index*) [private],[slot]

Definition at line 205 of file mperfwidget.cpp.

References KbPerf::liftHeight(), and perf.

```
205                                                 {
206     perf->liftHeight((KbPerf::height)(index + 1));
207 }
```

Here is the call graph for this function:



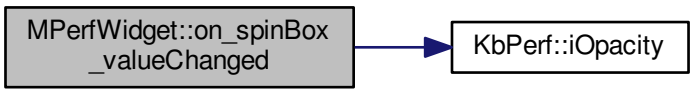
12.56.4.10 void MPerfWidget::on_spinBox_valueChanged (int *arg1*) [private], [slot]

Definition at line 219 of file mperfwidget.cpp.

References KbPerf::iOpacity(), and perf.

```
219                                                 {
220     if(!perf)
221         return;
222     perf->iOpacity(arg1 / 100.f);
223 }
```

Here is the call graph for this function:



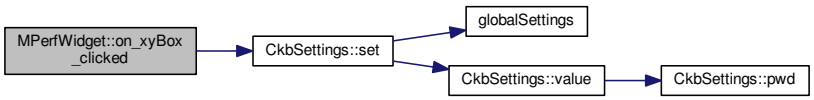
12.56.4.11 void MPerfWidget::on_xyBox_clicked (bool *checked*) [private], [slot]

Definition at line 191 of file mperfwidget.cpp.

References _xyLink, CkbSettings::set(), and xyLinkPath.

```
191                                                 {
192     _xyLink = !checked;
193     CkbSettings::set(xyLinkPath, checked);
194     // FIXME: update other MPerfWidgets, since this is a global setting
195 }
```

Here is the call graph for this function:



12.56.4.12 void MPerfWidget::setPerf (KbPerf * *newPerf,* KbProfile * *newProfile*)

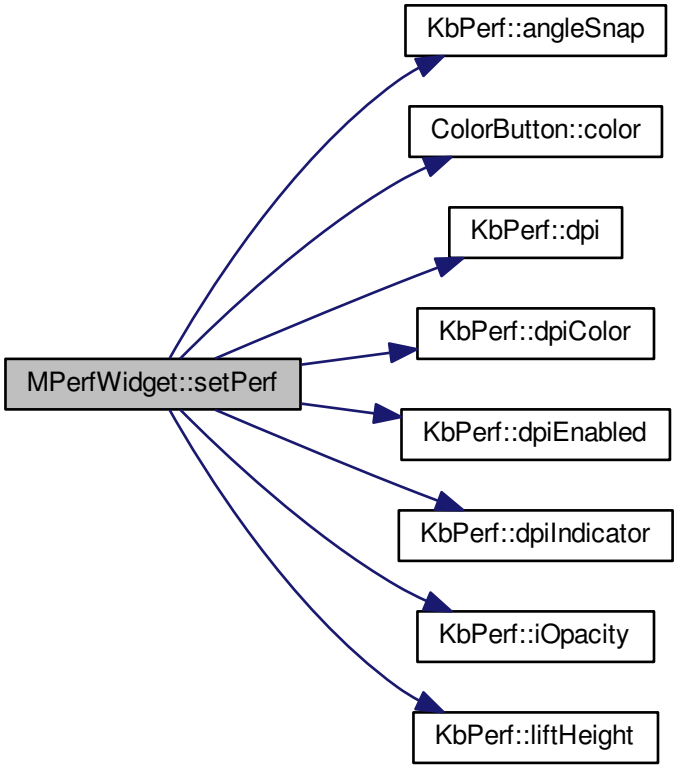
Definition at line 60 of file mperfwidget.cpp.

References _xyLink, KbPerf::angleSnap(), Ui_MPerfWidget::aSnapBox, ColorButton::color(), KbPerf::dpi(), DPI_COUNT, KbPerf::dpiColor(), KbPerf::dpiEnabled(), KbPerf::dpiIndicator(), MPerfWidget::DpiUi::enableCheck, Ui_MPerfWidget::iButtonO, MPerfWidget::DpiUi::indicator, Ui_MPerfWidget::indicBox, KbPerf::iOpacity(), Ui_MPerfWidget::lHeightBox, KbPerf::liftHeight(), KbPerf::OTHER, perf, profile, Ui_MPerfWidget::spinBox, stages, ui, MPerfWidget::DpiUi::xBox, and MPerfWidget::DpiUi::yBox.

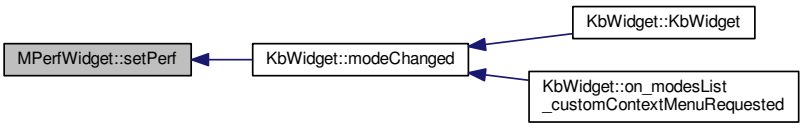
Referenced by KbWidget::modeChanged().

```
60                                                          {
61     perf = newPerf;
62     profile = newProfile;
63     ui->spinBox->setValue(round(perf->iOpacity() * 100.f));
64     for(int i = 0; i < DPI_COUNT; i++){
65         stages[i].indicator->color(perf->dpiColor(i));
66         bool oldLink = _xyLink;
67         // Don't force X/Y to the same value if they were set differently in the past
68         _xyLink = false;
69         stages[i].xBox->setValue(perf->dpi(i).x());
70         stages[i].yBox->setValue(perf->dpi(i).y());
71         _xyLink = oldLink;
72         if(stages[i].enableCheck)
73             stages[i].enableCheck->setChecked(perf->
      dpiEnabled(i));
74     }
75     ui->iButtonO->color(perf->dpiColor(KbPerf::OTHER));
76     ui->aSnapBox->setChecked(perf->angleSnap());
77     ui->lHeightBox->setCurrentIndex(perf->liftHeight() - 1);
78     ui->indicBox->setChecked(perf->dpiIndicator());
79 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.56.4.13 void MPerfWidget::sliderXMoved (int *index*) [private], [slot]

Definition at line 128 of file mperfwidget.cpp.

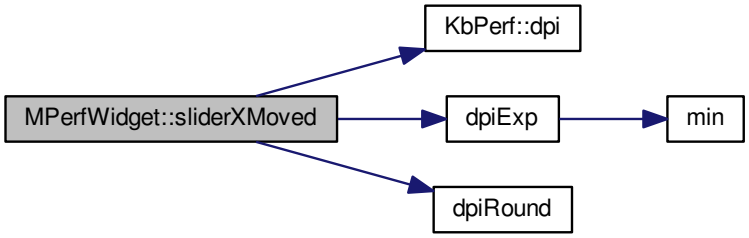
References _xyLink, KbPerf::dpi(), dpiExp(), dpiRound(), perf, SET_END, SET_START, stages, MPerfWidget::DpiUi::xBox, MPerfWidget::DpiUi::xSlider, and MPerfWidget::DpiUi::ySlider.

Referenced by MPerfWidget().

```
128                                    {
129     SET_START;
```

```
130     QSlider* slider = stages[index].xSlider;
131     int value = dpiRound(dpiExp(slider->value(), slider->minimum(), slider->maximum()));
132     stages[index].xBox->setValue(value);
133     if(_xyLink)
134         perf->dpi(index, value);
135     else
136         perf->dpi(index, QPoint(value, perf->dpi(index).y()));
137     SET_END;
138     if(_xyLink)
139         stages[index].ySlider->setValue(slider->value());
140 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.56.4.14 void MPerfWidget::sliderYMoved (int *index*) [private], [slot]

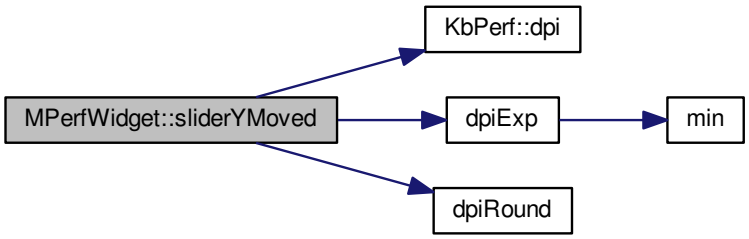
Definition at line 142 of file mperfwidget.cpp.

References _xyLink, KbPerf::dpi(), dpiExp(), dpiRound(), perf, SET_END, SET_START, stages, ui, Ui_MPerfWidget::xyBox, MPerfWidget::DpiUi::yBox, and MPerfWidget::DpiUi::ySlider.

Referenced by MPerfWidget().

```
142                                         {
143     SET_START;
144     QSlider* slider = stages[index].ySlider;
145     int value = dpiRound(dpiExp(slider->value(), slider->minimum(), slider->maximum()));
146     stages[index].yBox->setValue(value);
147     if(_xyLink)
148         perf->dpi(index, value);
149     else
150         perf->dpi(index, QPoint(perf->dpi(index).x(), value));
151     SET_END;
152     if(!ui->xyBox->isChecked())
153         // X/Y linked?
154         stages[index].xSlider->setValue(slider->value());
155 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.56.5 Field Documentation

12.56.5.1 bool MPerfWidget::_xyLink [private]

Definition at line 44 of file mperfwidget.h.

Referenced by boxXChanged(), boxYChanged(), MPerfWidget(), on_xyBox_clicked(), setPerf(), sliderXMoved(), and sliderYMoved().

12.56.5.2 QSignalMapper MPerfWidget::boxXMapper [private]

Definition at line 49 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.3 QSignalMapper MPerfWidget::boxYMapper [private]

Definition at line 49 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.4 QSignalMapper MPerfWidget::buttonMapper1 [private]

Definition at line 47 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.5 QSignalMapper MPerfWidget::buttonMapper2 [private]

Definition at line 47 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.6 bool MPerfWidget::colorLink [private]

Definition at line 45 of file mperfwidget.h.

Referenced by colorChanged(), and colorClicked().

12.56.5.7 const int MPerfWidget::DPI_COUNT = KbPerf::DPI_COUNT [static]

Definition at line 28 of file mperfwidget.h.

Referenced by colorChanged(), MPerfWidget(), and setPerf().

12.56.5.8 const int MPerfWidget::DPI_MAX = KbPerf::DPI_MAX [static]

Definition at line 29 of file mperfwidget.h.

12.56.5.9 const int MPerfWidget::DPI_MIN = KbPerf::DPI_MIN [static]

Definition at line 29 of file mperfwidget.h.

12.56.5.10 QSignalMapper MPerfWidget::enableMapper [private]

Definition at line 50 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.11 bool MPerfWidget::isSetting [private]

Definition at line 53 of file mperfwidget.h.

12.56.5.12 KbPerf* MPerfWidget::perf [private]

Definition at line 34 of file mperfwidget.h.

Referenced by boxXChanged(), boxYChanged(), colorChanged(), enableChanged(), on_aSnapBox_clicked(), on_copyButton_clicked(), on_indicBox_clicked(), on_lHeightBox_activated(), on_spinBox_valueChanged(), setPerf(), sliderXMoved(), and sliderYMoved().

12.56.5.13 KbProfile* MPerfWidget::profile [private]

Definition at line 35 of file mperfwidget.h.

Referenced by on_copyButton_clicked(), and setPerf().

12.56.5.14 QSignalMapper MPerfWidget::sliderXMapper [private]

Definition at line 48 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.15 QSignalMapper MPerfWidget::sliderYMapper [private]

Definition at line 48 of file mperfwidget.h.

Referenced by MPerfWidget().

12.56.5.16 DpiUi MPerfWidget::stages[DPI_COUNT] [private]

Definition at line 43 of file mperfwidget.h.

Referenced by boxXChanged(), boxYChanged(), colorChanged(), enableChanged(), MPerfWidget(), setPerf(), sliderXMoved(), and sliderYMoved().

12.56.5.17 Ui::MPerfWidget* MPerfWidget::ui [private]

Definition at line 32 of file mperfwidget.h.

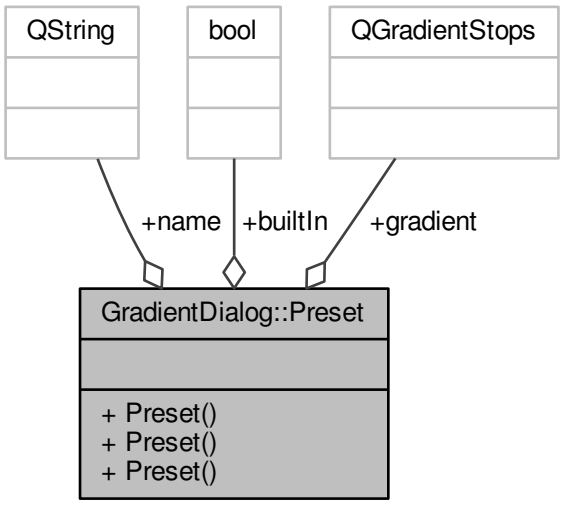
Referenced by colorChanged(), MPerfWidget(), setPerf(), sliderYMoved(), and ~MPerfWidget().

The documentation for this class was generated from the following files:

- src/ckb/mperfwidget.h
- src/ckb/mperfwidget.cpp

12.57 GradientDialog::Preset Struct Reference

Collaboration diagram for GradientDialog::Preset:



Public Member Functions

- Preset (QString _name, bool _builtIn=false)
- Preset (QString _name, const QGradientStops &_gradient)

- Preset ()

Data Fields

- QString name
- QGradientStops gradient
- bool builtIn

12.57.1 Detailed Description

Definition at line 23 of file gradientdialog.h.

12.57.2 Constructor & Destructor Documentation

12.57.2.1 GradientDialog::Preset::Preset (QString *_name,* bool *_builtIn* = `false`) [inline]

Definition at line 27 of file gradientdialog.h.

```
27 : name(_name), builtIn(_builtIn) {}
```

12.57.2.2 GradientDialog::Preset::Preset (QString *_name,* const QGradientStops & *_gradient*) [inline]

Definition at line 28 of file gradientdialog.h.

```
28 : name(_name), gradient(_gradient), builtIn(false) {}
```

12.57.2.3 GradientDialog::Preset::Preset ( ) [inline]

Definition at line 29 of file gradientdialog.h.

```
29 : builtIn(false) {}
```

12.57.3 Field Documentation

12.57.3.1 bool GradientDialog::Preset::builtIn

Definition at line 26 of file gradientdialog.h.

Referenced by GradientDialog::on_presetDelete_clicked(), GradientDialog::on_presetSave_clicked(), GradientDialog::setPreset(), and GradientDialog::~GradientDialog().

12.57.3.2 QGradientStops GradientDialog::Preset::gradient

Definition at line 25 of file gradientdialog.h.

Referenced by GradientDialog::GradientDialog(), GradientDialog::makeIcon(), GradientDialog::setPreset(), and GradientDialog::~GradientDialog().

12.57.3.3 QString GradientDialog::Preset::name

Definition at line 24 of file gradientdialog.h.

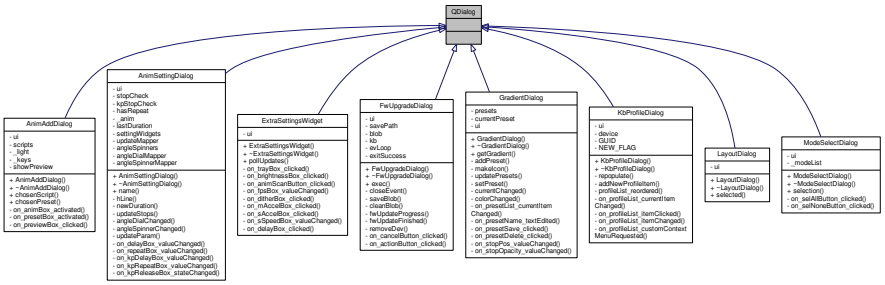
Referenced by GradientDialog::addPreset(), GradientDialog::on_presetDelete_clicked(), GradientDialog::on_presetSave_clicked(), GradientDialog::setPreset(), GradientDialog::updatePresets(), and GradientDialog::~GradientDialog().

The documentation for this struct was generated from the following file:

- src/ckb/gradientdialog.h

12.58 QDialog Class Reference

Inheritance diagram for QDialog:



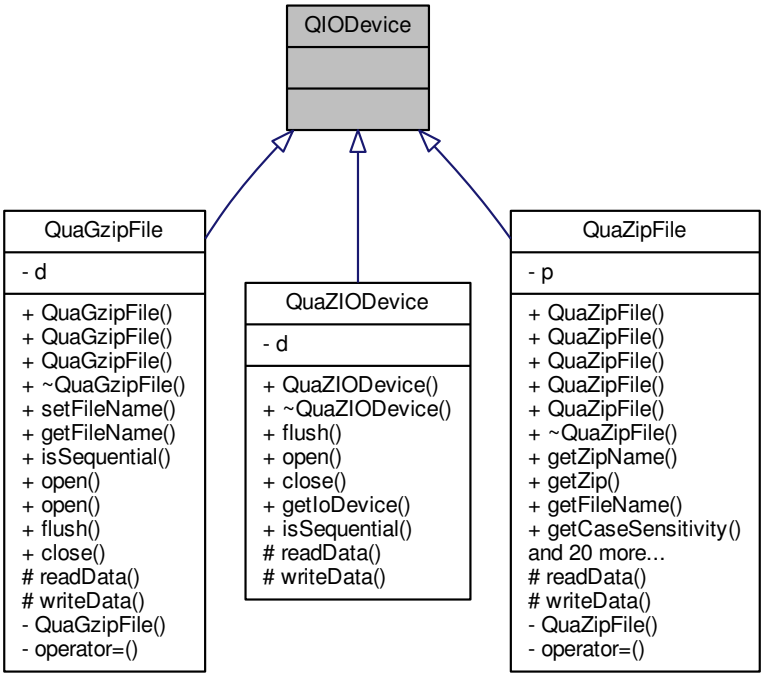
Collaboration diagram for QDialog:

The documentation for this class was generated from the following file:

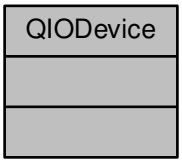
- src/ckb/extrasettingswidget.h

12.59 QIODevice Class Reference

Inheritance diagram for QIODevice:



Collaboration diagram for QIODevice:

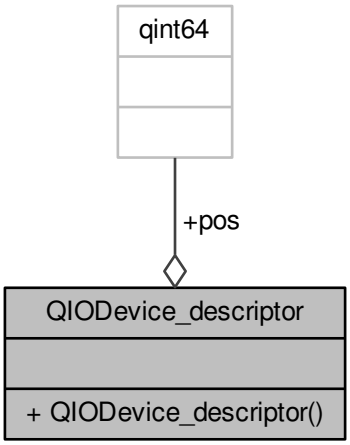


The documentation for this class was generated from the following file:

- src/ckb/quazip/quazipfile.h

12.60 QIODevice_descriptor Struct Reference

Collaboration diagram for QIODevice_descriptor:



Public Member Functions

- QIODevice_descriptor ()

Data Fields

- qint64 pos

12.60.1 Detailed Description

Definition at line 78 of file qioapi.cpp.

12.60.2 Constructor & Destructor Documentation

12.60.2.1 QIODevice_descriptor::QIODevice_descriptor ( ) [inline]

Definition at line 81 of file qioapi.cpp.

```
81                                  :
82          pos(0)
83      {}
```

12.60.3 Field Documentation

12.60.3.1 qint64 QIODevice_descriptor::pos

Definition at line 80 of file qioapi.cpp.

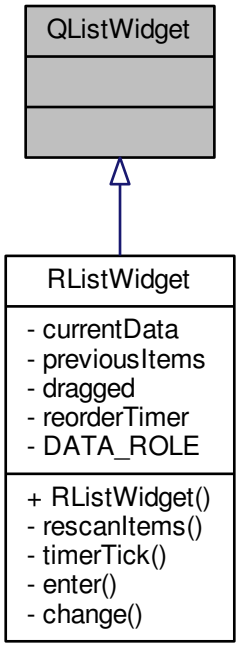
Referenced by qiodevice64_tell_file_func(), qiodevice_open_file_func(), qiodevice_read_file_func(), qiodevice_tell_file_func(), and qiodevice_write_file_func().

The documentation for this struct was generated from the following file:

- src/ckb/quazip/qioapi.cpp

12.61 QListWidget Class Reference

Inheritance diagram for QListWidget:



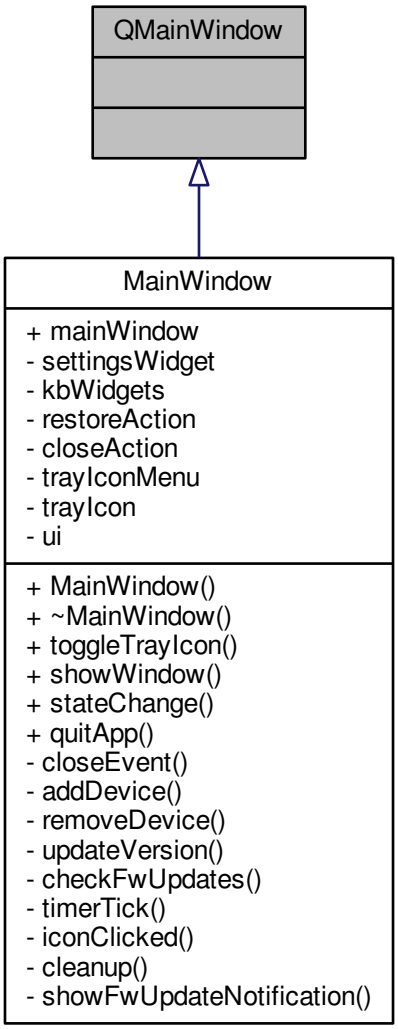
Collaboration diagram for QListWidget:

The documentation for this class was generated from the following file:

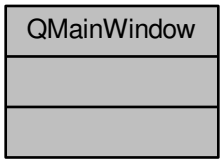
- src/ckb/rlistwidget.h

12.62 QMainWindow Class Reference

Inheritance diagram for QMainWindow:



Collaboration diagram for QMainWindow:

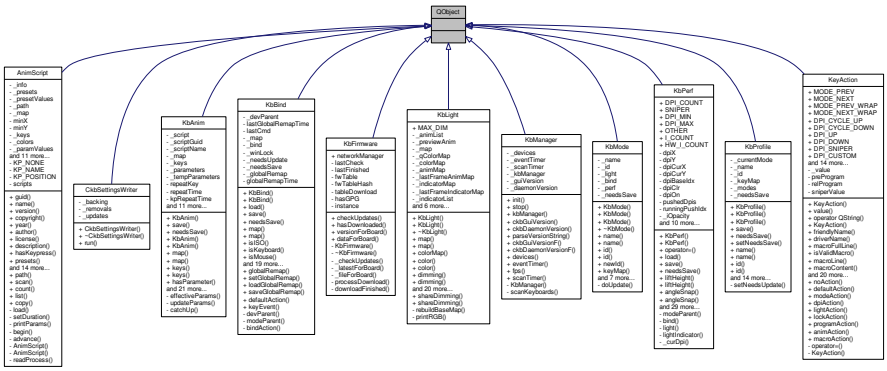


The documentation for this class was generated from the following file:

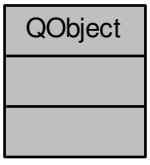
- src/ckb/mainwindow.h

12.63 QObject Class Reference

Inheritance diagram for QObject:



Collaboration diagram for QObject:

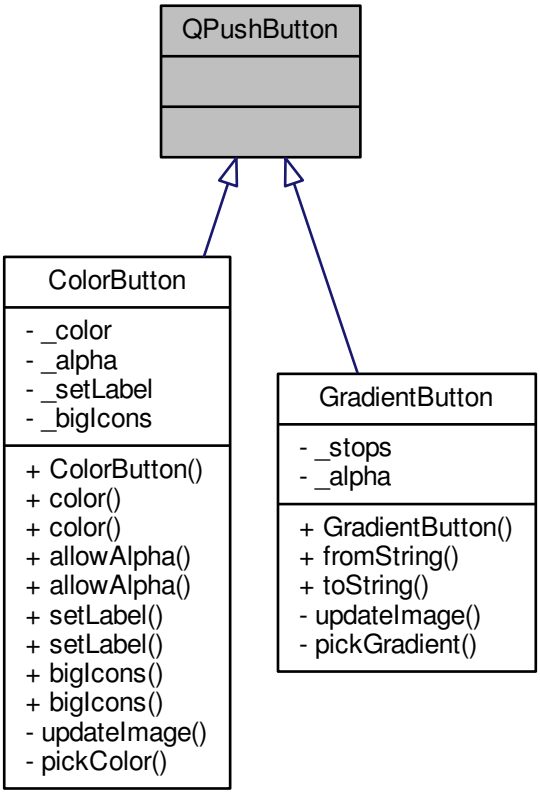


The documentation for this class was generated from the following file:

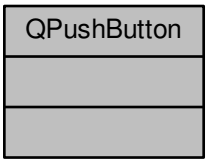
- src/ckb/kbprofile.h

12.64 QPushButton Class Reference

Inheritance diagram for QPushButton:



Collaboration diagram for QPushButton:

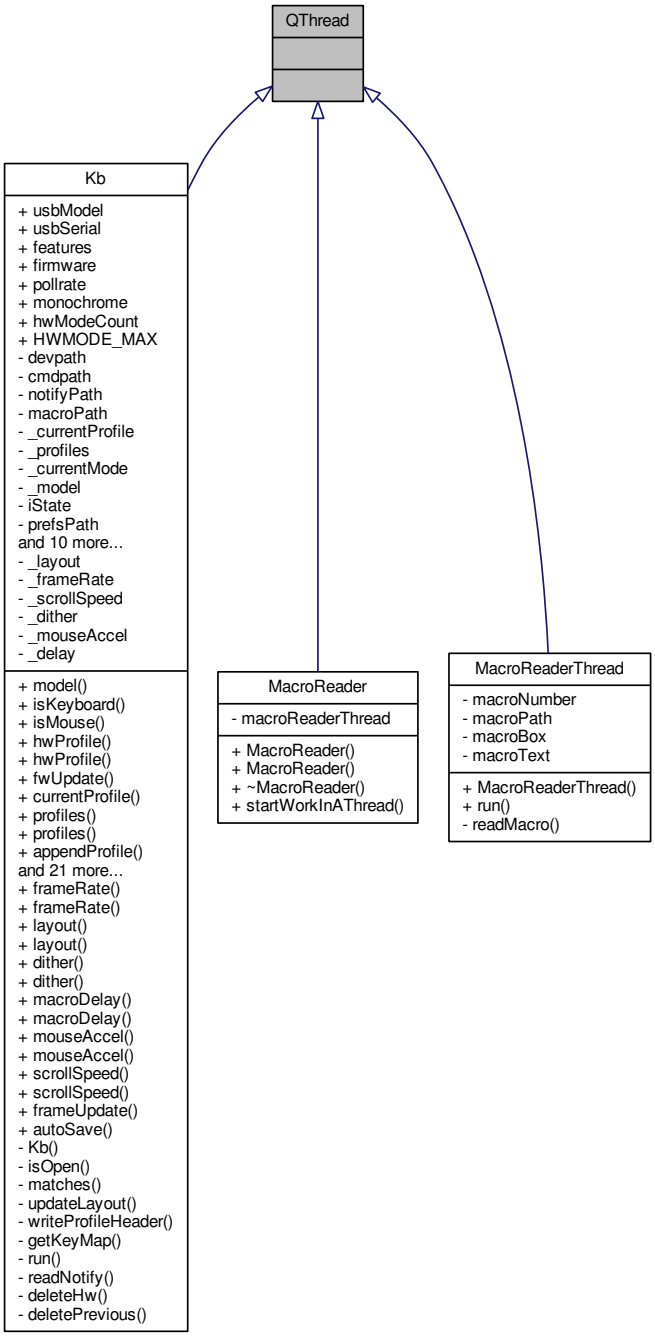


The documentation for this class was generated from the following file:

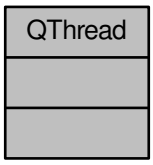
- src/ckb/colorbutton.h

12.65 QThread Class Reference

Inheritance diagram for QThread:



Collaboration diagram for QThread:



The documentation for this class was generated from the following file:

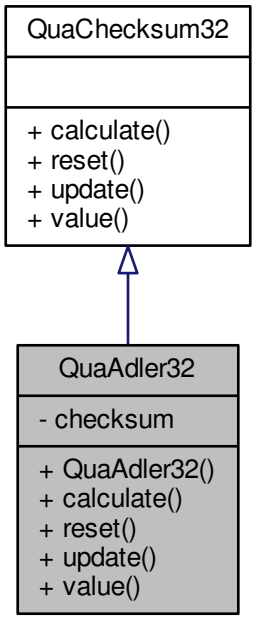
- src/ckb/macroreader.h

12.66 QuaAdler32 Class Reference

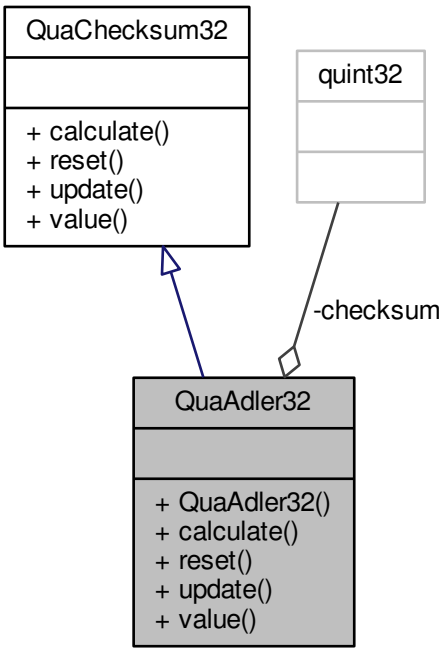
Adler32 checksum.

`#include <quazip/quaadler32.h>`

Inheritance diagram for QuaAdler32:



Collaboration diagram for QuaAdler32:



Public Member Functions

- QuaAdler32 ()
- quint32 calculate (const QByteArray &data)

  Calculates the checksum for data.
- void reset ()

  Resets the calculation on a checksun for a stream.
- void update (const QByteArray &buf)

  Updates the calculated checksum for the stream.
- quint32 value ()

  Value of the checksum calculated for the stream passed throw update().

Private Attributes

- quint32 checksum

12.66.1 Detailed Description

This class wrappers the adler32 function with the QuaChecksum32 interface. See QuaChecksum32 for more info.

Definition at line 38 of file quaadler32.h.

12.66.2 Constructor & Destructor Documentation

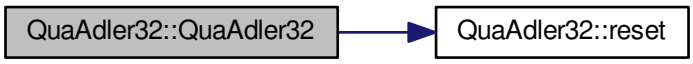
12.66.2.1 QuaAdler32::QuaAdler32 ( )

Definition at line 30 of file quaadler32.cpp.

References reset().

```
31 {
32     reset();
33 }
```

Here is the call graph for this function:



12.66.3 Member Function Documentation

12.66.3.1 quint32 QuaAdler32::calculate (const QByteArray & *data*) [virtual]

data source data

Returns

data checksum

This function has no efect on the value returned by value().

Implements QuaChecksum32.

Definition at line 35 of file quaadler32.cpp.

```
36 {
37     return adler32( adler32(0L, Z_NULL, 0), (const Bytef*)data.data(), data.size() );
38 }
```

12.66.3.2 void QuaAdler32::reset ( ) [virtual]

Implements QuaChecksum32.

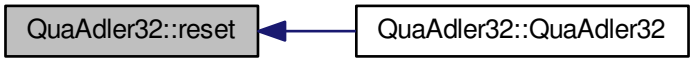
Definition at line 40 of file quaadler32.cpp.

References checksum.

Referenced by QuaAdler32().

```
41 {
42     checksum = adler32(0L, Z_NULL, 0);
43 }
```

Here is the caller graph for this function:



12.66.3.3 void QuaAdler32::update (const QByteArray & *buf*) [virtual]

buf next portion of data from the stream

Implements QuaChecksum32.

Definition at line 45 of file quaadler32.cpp.

References checksum.

```
46 {
47     checksum = adler32( checksum, (const Bytef*)buf.data(), buf.size() );
48 }
```

12.66.3.4 quint32 QuaAdler32::value ( ) [virtual]

Returns

checksum

Implements QuaChecksum32.

Definition at line 50 of file quaadler32.cpp.

References checksum.

```
51 {
52     return checksum;
53 }
```

12.66.4 Field Documentation

12.66.4.1 quint32 QuaAdler32::checksum [private]

Definition at line 51 of file quaadler32.h.

Referenced by reset(), update(), and value().

The documentation for this class was generated from the following files:

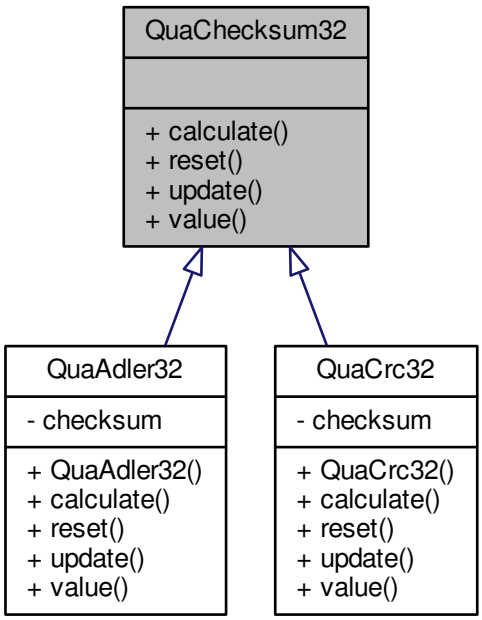
- src/ckb/quazip/quaadler32.h
- src/ckb/quazip/quaadler32.cpp

12.67 QuaChecksum32 Class Reference

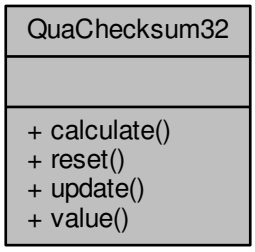
Checksum interface.

`#include <quazip/quachecksum32.h>`

Inheritance diagram for QuaChecksum32:



Collaboration diagram for QuaChecksum32:



Public Member Functions

- virtual quint32 calculate (const QByteArray &data)=0

  Calculates the checksum for data.
- virtual void reset ()=0

  Resets the calculation on a checksun for a stream.
- virtual void update (const QByteArray &buf)=0

Updates the calculated checksum for the stream.

- virtual quint32 value ()=0

  Value of the checksum calculated for the stream passed throw update().

12.67.1 Detailed Description

This is an interface for 32 bit checksums. Classes implementing this interface can calcunate a certin checksum in a single step:

```
QChecksum32 *crc32 = new QuaCrc32();
rasoult = crc32->calculate(data);
```

or by streaming the data:

```
QChecksum32 *crc32 = new QuaCrc32();
while(!fileA.atEnd())
    crc32->update(fileA.read(bufSize));
resoultA = crc32->value();
crc32->reset();
while(!fileB.atEnd())
    crc32->update(fileB.read(bufSize));
resoultB = crc32->value();
```

Definition at line 52 of file quachecksum32.h.

12.67.2 Member Function Documentation

12.67.2.1 virtual quint32 QuaChecksum32::calculate (const QByteArray & *data*) [pure virtual]

data source data

Returns

data checksum

This function has no efect on the value returned by value().

Implemented in QuaAdler32, and QuaCrc32.

12.67.2.2 virtual void QuaChecksum32::reset ( ) [pure virtual]

Implemented in QuaAdler32, and QuaCrc32.

12.67.2.3 virtual void QuaChecksum32::update (const QByteArray & *buf*) [pure virtual]

buf next portion of data from the stream

Implemented in QuaAdler32, and QuaCrc32.

12.67.2.4 virtual quint32 QuaChecksum32::value ( ) [pure virtual]

Returns

checksum

Implemented in QuaAdler32, and QuaCrc32.

The documentation for this class was generated from the following file:

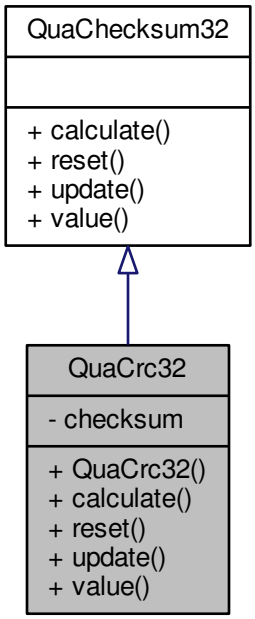
- src/ckb/quazip/quachecksum32.h

12.68 QuaCrc32 Class Reference

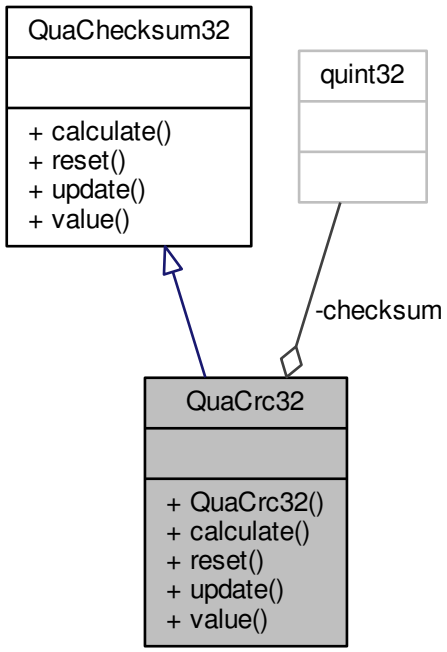
CRC32 checksum.

`#include <quazip/quacrc32.h>`

Inheritance diagram for QuaCrc32:



Collaboration diagram for QuaCrc32:



Public Member Functions

- QuaCrc32 ()
- quint32 calculate (const QByteArray &data)

  Calculates the checksum for data.
- void reset ()

  Resets the calculation on a checksun for a stream.
- void update (const QByteArray &buf)

  Updates the calculated checksum for the stream.
- quint32 value ()

  Value of the checksum calculated for the stream passed throw update().

Private Attributes

- quint32 checksum

12.68.1 Detailed Description

This class wrappers the crc32 function with the QuaChecksum32 interface. See QuaChecksum32 for more info.

Definition at line 35 of file quacrc32.h.

12.68.2 Constructor & Destructor Documentation

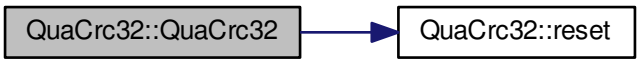
12.68.2.1 QuaCrc32::QuaCrc32 ( )

Definition at line 29 of file quacrc32.cpp.

References reset().

```
30 {
31     reset();
32 }
```

Here is the call graph for this function:



12.68.3 Member Function Documentation

12.68.3.1 quint32 QuaCrc32::calculate (const QByteArray & *data*) [virtual]

data source data

Returns

data checksum

This function has no efect on the value returned by value().

Implements QuaChecksum32.

Definition at line 34 of file quacrc32.cpp.

```
35 {
36     return crc32( crc32(0L, Z_NULL, 0), (const Bytef*)data.data(), data.size() );
37 }
```

12.68.3.2 void QuaCrc32::reset ( ) [virtual]

Implements QuaChecksum32.

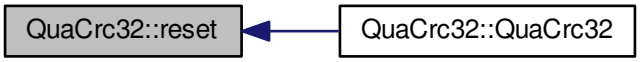
Definition at line 39 of file quacrc32.cpp.

References checksum.

Referenced by QuaCrc32().

```
40 {
41     checksum = crc32(0L, Z_NULL, 0);
42 }
```

Here is the caller graph for this function:



12.68.3.3 void QuaCrc32::update (const QByteArray & *buf*) [virtual]

buf next portion of data from the stream

Implements QuaChecksum32.

Definition at line 44 of file quacrc32.cpp.

References checksum.

```
45 {
46     checksum = crc32( checksum, (const Bytef*)buf.data(), buf.size() );
47 }
```

12.68.3.4 quint32 QuaCrc32::value ( ) [virtual]

Returns

checksum

Implements QuaChecksum32.

Definition at line 49 of file quacrc32.cpp.

References checksum.

```
50 {
51     return checksum;
52 }
```

12.68.4 Field Documentation

12.68.4.1 quint32 QuaCrc32::checksum [private]

Definition at line 47 of file quacrc32.h.

Referenced by reset(), update(), and value().

The documentation for this class was generated from the following files:

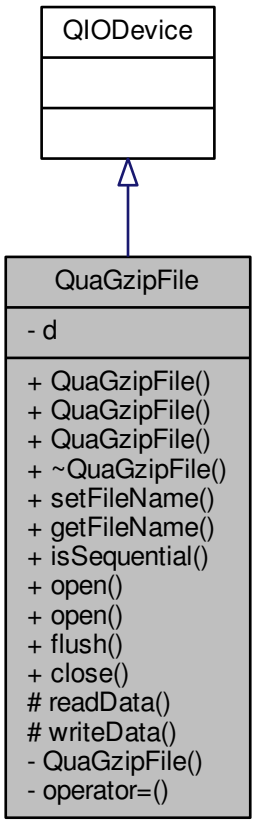
- src/ckb/quazip/quacrc32.h
- src/ckb/quazip/quacrc32.cpp

12.69 QuaGzipFile Class Reference

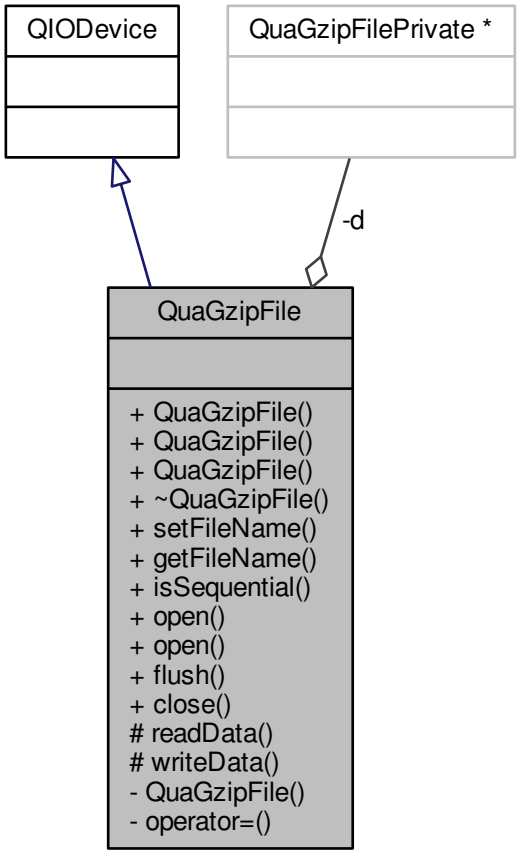
GZIP file.

`#include <src/ckb/quazip/quagzipfile.h>`

Inheritance diagram for QuaGzipFile:



Collaboration diagram for QuaGzipFile:



Public Member Functions

- QuaGzipFile ()

  Empty constructor.
- QuaGzipFile (QObject *parent)

  Empty constructor with a parent.
- QuaGzipFile (const QString &fileName, QObject *parent=NULL)

  Constructor.
- virtual ~QuaGzipFile ()

  Destructor.
- void setFileName (const QString &fileName)

  Sets the name of the GZIP file to be opened.
- QString getFileName () const

  Returns the name of the GZIP file.
- virtual bool isSequential () const

  Returns true.
- virtual bool open (QIODevice::OpenMode mode)

  Opens the file.

- virtual bool open (int fd, QIODevice::OpenMode mode)

  Opens the file.
- virtual bool flush ()

  Flushes data to file.
- virtual void close ()

  Closes the file.

Protected Member Functions

- virtual qint64 readData (char *data, qint64 maxSize)

  Implementation of QIODevice::readData().
- virtual qint64 writeData (const char *data, qint64 maxSize)

  Implementation of QIODevice::writeData().

Private Member Functions

- QuaGzipFile (const QuaGzipFile &that)
- QuaGzipFile & operator= (const QuaGzipFile &that)

Private Attributes

- QuaGzipFilePrivate * d

12.69.1 Detailed Description

This class is a wrapper around GZIP file access functions in zlib. Unlike QuaZip classes, it doesn't allow reading from a GZIP file opened as QIODevice, for example, if your GZIP file is in QBuffer. It only provides QIODevice access to a GZIP file contents, but the GZIP file itself must be identified by its name on disk or by descriptor id.

Definition at line 39 of file quagzipfile.h.

12.69.2 Constructor & Destructor Documentation

12.69.2.1 QuaGzipFile::QuaGzipFile ( )

Must call setFileName() before trying to open.

Definition at line 87 of file quagzipfile.cpp.

```
87                          :
88 d(new QuaGzipFilePrivate())
89 {
90 }
```

12.69.2.2 QuaGzipFile::QuaGzipFile (QObject * *parent*)

Must call setFileName() before trying to open.

Parameters

parent	The parent object, as per QObject logic.

Definition at line 92 of file quagzipfile.cpp.

```
92                                       :
93 QIODevice(parent),
94 d(new QuaGzipFilePrivate())
95 {
96 }
```

12.69.2.3 QuaGzipFile::QuaGzipFile (const QString & *fileName*, QObject * *parent* = NULL)

Parameters

fileName	The name of the GZIP file.
parent	The parent object, as per QObject logic.

Definition at line 98 of file quagzipfile.cpp.

```
98                                                                  :
99   QIODevice(parent),
100 d(new QuaGzipFilePrivate(fileName))
101 {
102 }
```

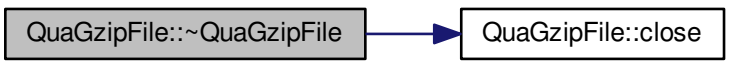
12.69.2.4 QuaGzipFile::~QuaGzipFile ( ) [virtual]

Definition at line 104 of file quagzipfile.cpp.

References close(), and d.

```
105 {
106   if (isOpen()) {
107     close();
108   }
109   delete d;
110 }
```

Here is the call graph for this function:



12.69.2.5 QuaGzipFile::QuaGzipFile (const QuaGzipFile & *that*) [private]

12.69.3 Member Function Documentation

12.69.3.1 void QuaGzipFile::close ( ) [virtual]

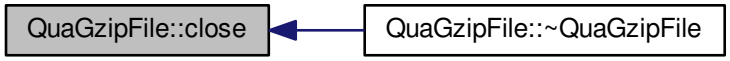
Definition at line 152 of file quagzipfile.cpp.

References d.

Referenced by ~QuaGzipFile().

```
153 {
154   QIODevice::close();
155   gzclose(d->gzd);
156 }
```

Here is the caller graph for this function:



12.69.3.2 bool QuaGzipFile::flush ( ) [virtual]

The data is written using Z_SYNC_FLUSH mode. Doesn't make any sense when reading.

Definition at line 147 of file quagzipfile.cpp.

References d.

```
148 {
149     return gzflush(d->gzd, Z_SYNC_FLUSH) == Z_OK;
150 }
```

12.69.3.3 QString QuaGzipFile::getFileName ( ) const

Definition at line 117 of file quagzipfile.cpp.

References d.

```
118 {
119     return d->fileName;
120 }
```

12.69.3.4 bool QuaGzipFile::isSequential ( ) const [virtual]

Strictly speaking, zlib supports seeking for GZIP files, but it is poorly implemented, because there is no way to implement it properly. For reading, seeking backwards is very slow, and for writing, it is downright impossible. Therefore, QuaGzipFile does not support seeking at all.

Definition at line 122 of file quagzipfile.cpp.

```
123 {
124   return true;
125 }
```

12.69.3.5 bool QuaGzipFile::open (QIODevice::OpenMode *mode*) [virtual]

Parameters

mode	Can be either QIODevice::Write or QIODevice::Read. ReadWrite and Append aren't supported.

Definition at line 127 of file quagzipfile.cpp.

References d.

```
128 {
129     QString error;
130     if (!d->open(d->fileName, mode, error)) {
131         setErrorString(error);
132         return false;
133     }
134     return QIODevice::open(mode);
135 }
```

12.69.3.6 bool QuaGzipFile::open (int *fd*, QIODevice::OpenMode *mode*) [virtual]

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Parameters

fd	The file descriptor to read/write the GZIP file from/to.
mode	Can be either QIODevice::Write or QIODevice::Read. ReadWrite and Append aren't supported.

Definition at line 137 of file quagzipfile.cpp.

References d.

```
138 {
139     QString error;
140     if (!d->open(fd, mode, error)) {
141         setErrorString(error);
142         return false;
143     }
144     return QIODevice::open(mode);
145 }
```

12.69.3.7 QuaGzipFile& QuaGzipFile::operator= (const QuaGzipFile & *that*) [private]

12.69.3.8 qint64 QuaGzipFile::readData (char * *data*, qint64 *maxSize*) [protected], [virtual]

Definition at line 158 of file quagzipfile.cpp.

References d.

```
159 {
160     return gzread(d->gzd, (voidp)data, (unsigned)maxSize);
161 }
```

12.69.3.9 void QuaGzipFile::setFileName (const QString & *fileName*)

Definition at line 112 of file quagzipfile.cpp.

References d.

```
113 {
114     d->fileName = fileName;
115 }
```

12.69.3.10 qint64 QuaGzipFile::writeData (const char * *data*, qint64 *maxSize*) [protected], [virtual]

Definition at line 163 of file quagzipfile.cpp.

References d.

```
164 {
165     if (maxSize == 0)
166         return 0;
167     int written = gzwrite(d->gzd, (voidp)data, (unsigned)maxSize);
168     if (written == 0)
169         return -1;
170     else
171         return written;
172 }
```

12.69.4 Field Documentation

12.69.4.1 QuaGzipFilePrivate* QuaGzipFile::d [private]

Definition at line 105 of file quagzipfile.h.

Referenced by close(), flush(), getFileName(), open(), readData(), setFileName(), writeData(), and ~QuaGzipFile().

The documentation for this class was generated from the following files:

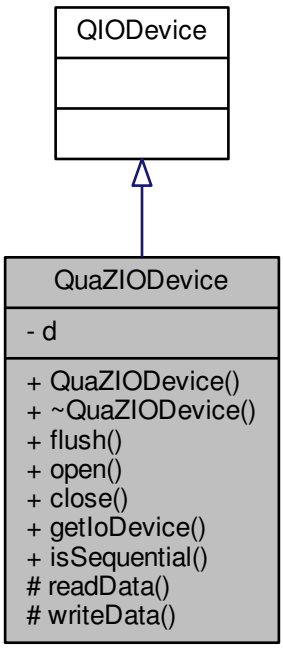
- src/ckb/quazip/quagzipfile.h
- src/ckb/quazip/quagzipfile.cpp

12.70 QuaZIODevice Class Reference

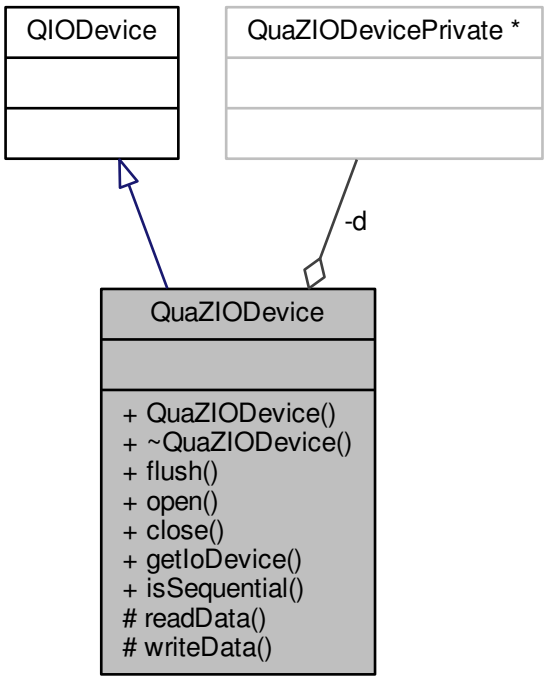
A class to compress/decompress QIODevice.

`#include <src/ckb/quazip/quaziodevice.h>`

Inheritance diagram for QuaZIODevice:



Collaboration diagram for QuaZIODevice:



Public Member Functions

- QuaZIODevice (QIODevice ∗io, QObject ∗parent=NULL)

  Constructor.
- ∼QuaZIODevice ()

  Destructor.
- virtual bool flush ()

  Flushes data waiting to be written.
- virtual bool open (QIODevice::OpenMode mode)

  Opens the device.
- virtual void close ()

  Closes this device, but not the underlying one.
- QIODevice ∗ getIoDevice () const

  Returns the underlying device.
- virtual bool isSequential () const

  Returns true.

Protected Member Functions

- virtual qint64 readData (char ∗data, qint64 maxSize)

  Implementation of QIODevice::readData().
- virtual qint64 writeData (const char ∗data, qint64 maxSize)

  Implementation of QIODevice::writeData().

Private Attributes

- QuaZIODevicePrivate * d

12.70.1 Detailed Description

This class can be used to compress any data written to QIODevice or decompress it back. Compressing data sent over a QTcpSocket is a good example.

Definition at line 41 of file quaziodevice.h.

12.70.2 Constructor & Destructor Documentation

12.70.2.1 QuaZIODevice::QuaZIODevice (QIODevice * *io*, QObject * *parent* = NULL)

Parameters

io	The QIODevice to read/write.
parent	The parent object, as per QObject logic.

Definition at line 123 of file quaziodevice.cpp.

```
123                                                     :
124     QIODevice(parent),
125     d(new QuaZIODevicePrivate(io))
126 {
127   connect(io, SIGNAL(readyRead()), SIGNAL(readyRead()));
128 }
```

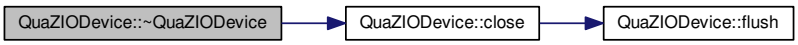
12.70.2.2 QuaZIODevice::~QuaZIODevice ( )

Definition at line 130 of file quaziodevice.cpp.

References close(), and d.

```
131 {
132     if (isOpen())
133         close();
134     delete d;
135 }
```

Here is the call graph for this function:



12.70.3 Member Function Documentation

12.70.3.1 void QuaZIODevice::close ( ) [virtual]

The underlying QIODevice is not closed in case you want to write something else to it.

Definition at line 169 of file quaziodevice.cpp.

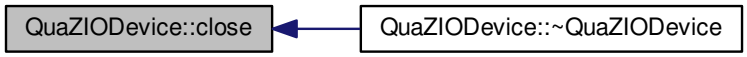
References d, and flush().

Referenced by ~QuaZIODevice().

```
170 {
171     if ((openMode() & QIODevice::ReadOnly) != 0) {
172         if (inflateEnd(&d->zins) != Z_OK) {
173             setErrorString(d->zins.msg);
174         }
175     }
176     if ((openMode() & QIODevice::WriteOnly) != 0) {
177         flush();
178         if (deflateEnd(&d->zouts) != Z_OK) {
179             setErrorString(d->zouts.msg);
180         }
181     }
182     QIODevice::close();
183 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.70.3.2 bool QuaZIODevice::flush ( ) [virtual]

Unfortunately, as QIODevice doesn't support flush() by itself, the only thing this method does is write the compressed data into the device using Z_SYNC_FLUSH mode. If you need the compressed data to actually be flushed from the buffer of the underlying QIODevice, you need to call its flush() method as well, providing it supports it (like QTcpSocket does). Example:

```
QuaZIODevice dev(&sock);
dev.open(QIODevice::Write);
dev.write(yourDataGoesHere);
dev.flush();
sock->flush(); // this actually sends data to network
```

This may change in the future versions of QuaZIP by implementing an ugly hack: trying to cast the QIODevice using qobject_cast to known flush()-supporting subclasses, and calling flush if the resulting pointer is not zero.

Definition at line 281 of file quaziodevice.cpp.

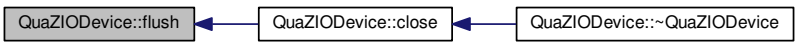
References d, and QUAZIO_OUTBUFSIZE.

Referenced by close().

```
282 {
283     QString error;
284     if (d->doFlush(error) < 0) {
```

```
285         setErrorString(error);
286         return false;
287     }
288     // can't flush buffer, some data is still waiting
289     if (d->outBufPos < d->outBufSize)
290         return true;
291     Bytef c = 0;
292     d->zouts.next_in = &c; // fake input buffer
293     d->zouts.avail_in = 0; // of zero size
294     do {
295         d->zouts.next_out = (Bytef *) d->outBuf;
296         d->zouts.avail_out = QUAZIO_OUTBUFSIZE;
297         switch (deflate(&d->zouts, Z_SYNC_FLUSH)) {
298         case Z_OK:
299           d->outBufSize = (char *) d->zouts.next_out - d->outBuf;
300           if (d->doFlush(error) < 0) {
301               setErrorString(error);
302               return false;
303           }
304           if (d->outBufPos < d->outBufSize)
305               return true;
306           break;
307         case Z_BUF_ERROR: // nothing to write?
308           return true;
309         default:
310           setErrorString(QString::fromLocal8Bit(d->zouts.msg));
311           return false;
312         }
313     } while (d->zouts.avail_out == 0);
314     return true;
315 }
```

Here is the caller graph for this function:



12.70.3.3 QIODevice * QuaZIODevice::getIoDevice ( ) const

Definition at line 137 of file quaziodevice.cpp.

References d.

```
138 {
139     return d->io;
140 }
```

12.70.3.4 bool QuaZIODevice::isSequential ( ) const [virtual]

Definition at line 317 of file quaziodevice.cpp.

```
318 {
319   return true;
320 }
```

12.70.3.5 bool QuaZIODevice::open (QIODevice::OpenMode *mode*) [virtual]

Parameters

mode	Neither QIODevice::ReadWrite nor QIODevice::Append are not supported.

Definition at line 142 of file quaziodevice.cpp.

References d.

```
143 {
144     if ((mode & QIODevice::Append) != 0) {
145         setErrorString(trUtf8("QIODevice::Append is not supported for"
146                     " QuaZIODevice"));
147         return false;
148     }
149     if ((mode & QIODevice::ReadWrite) == QIODevice::ReadWrite) {
150         setErrorString(trUtf8("QIODevice::ReadWrite is not supported for"
151                     " QuaZIODevice"));
152         return false;
153     }
154     if ((mode & QIODevice::ReadOnly) != 0) {
155         if (inflateInit(&d->zins) != Z_OK) {
156             setErrorString(d->zins.msg);
157             return false;
158         }
159     }
160     if ((mode & QIODevice::WriteOnly) != 0) {
161         if (deflateInit(&d->zouts, Z_DEFAULT_COMPRESSION) != Z_OK) {
162             setErrorString(d->zouts.msg);
163             return false;
164         }
165     }
166     return QIODevice::open(mode);
167 }
```

12.70.3.6 qint64 QuaZIODevice::readData (char * *data*, qint64 *maxSize*) [protected], [virtual]

Definition at line 185 of file quaziodevice.cpp.

References d, and QUAZIO_INBUFSIZE.

```
186 {
187   int read = 0;
188   while (read < maxSize) {
189     if (d->inBufPos == d->inBufSize) {
190       d->inBufPos = 0;
191       d->inBufSize = d->io->read(d->inBuf, QUAZIO_INBUFSIZE);
192       if (d->inBufSize == -1) {
193         d->inBufSize = 0;
194         setErrorString(d->io->errorString());
195         return -1;
196       }
197       if (d->inBufSize == 0)
198         break;
199     }
200     while (read < maxSize && d->inBufPos < d->inBufSize) {
201       d->zins.next_in = (Bytef *) (d->inBuf + d->inBufPos);
202       d->zins.avail_in = d->inBufSize - d->inBufPos;
203       d->zins.next_out = (Bytef *) (data + read);
204       d->zins.avail_out = (uInt) (maxSize - read); // hope it's less than 2GB
205       int more = 0;
206       switch (inflate(&d->zins, Z_SYNC_FLUSH)) {
207       case Z_OK:
208         read = (char *) d->zins.next_out - data;
209         d->inBufPos = (char *) d->zins.next_in - d->inBuf;
210         break;
211       case Z_STREAM_END:
212         read = (char *) d->zins.next_out - data;
213         d->inBufPos = (char *) d->zins.next_in - d->inBuf;
214         return read;
215       case Z_BUF_ERROR: // this should never happen, but just in case
216         if (!d->zBufError) {
217           qWarning("Z_BUF_ERROR detected with %d/%d in/out, weird",
218               d->zins.avail_in, d->zins.avail_out);
219           d->zBufError = true;
220         }
221         memmove(d->inBuf, d->inBuf + d->inBufPos, d->inBufSize - d->inBufPos);
222         d->inBufSize -= d->inBufPos;
223         d->inBufPos = 0;
224         more = d->io->read(d->inBuf + d->inBufSize, QUAZIO_INBUFSIZE -
      d->inBufSize);
```

```
225         if (more == -1) {
226           setErrorString(d->io->errorString());
227           return -1;
228         }
229         if (more == 0)
230           return read;
231         d->inBufSize += more;
232         break;
233       default:
234         setErrorString(QString::fromLocal8Bit(d->zins.msg));
235         return -1;
236       }
237     }
238   }
239 #ifdef QUAZIP_ZIODEVICE_DEBUG_INPUT
240   indebug.write(data, read);
241 #endif
242   return read;
243 }
```

12.70.3.7 qint64 QuaZIODevice::writeData (const char * *data*, qint64 *maxSize*) [protected], [virtual]

Definition at line 245 of file quaziodevice.cpp.

References d, and QUAZIO_OUTBUFSIZE.

```
246 {
247   int written = 0;
248   QString error;
249   if (d->doFlush(error) == -1) {
250     setErrorString(error);
251     return -1;
252   }
253   while (written < maxSize) {
254       // there is some data waiting in the output buffer
255     if (d->outBufPos < d->outBufSize)
256       return written;
257     d->zouts.next_in = (Bytef *) (data + written);
258     d->zouts.avail_in = (uInt) (maxSize - written); // hope it's less than 2GB
259     d->zouts.next_out = (Bytef *) d->outBuf;
260     d->zouts.avail_out = QUAZIO_OUTBUFSIZE;
261     switch (deflate(&d->zouts, Z_NO_FLUSH)) {
262     case Z_OK:
263       written = (char *) d->zouts.next_in - data;
264       d->outBufSize = (char *) d->zouts.next_out - d->outBuf;
265       break;
266     default:
267       setErrorString(QString::fromLocal8Bit(d->zouts.msg));
268       return -1;
269     }
270     if (d->doFlush(error) == -1) {
271       setErrorString(error);
272       return -1;
273     }
274   }
275 #ifdef QUAZIP_ZIODEVICE_DEBUG_OUTPUT
276   debug.write(data, written);
277 #endif
278   return written;
279 }
```

12.70.4 Field Documentation

12.70.4.1 QuaZIODevicePrivate* QuaZIODevice::d [private]

Definition at line 96 of file quaziodevice.h.

Referenced by close(), flush(), getIoDevice(), open(), readData(), writeData(), and ~QuaZIODevice().

The documentation for this class was generated from the following files:

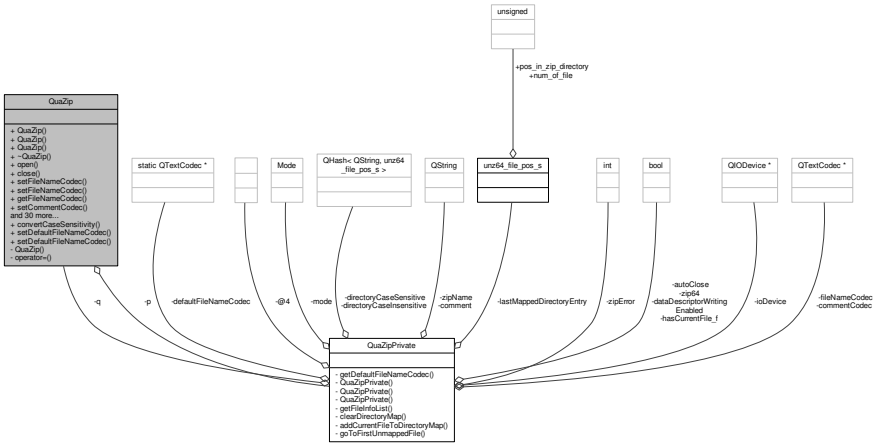
- src/ckb/quazip/quaziodevice.h
- src/ckb/quazip/quaziodevice.cpp

12.71 QuaZip Class Reference

ZIP archive.

`#include <quazip/quazip.h>`

Collaboration diagram for QuaZip:



Public Types

- enum Constants { MAX_FILE_NAME_LENGTH =256 }

  Useful constants.
- enum Mode {
  mdNotOpen, mdUnzip, mdCreate, mdAppend,
  mdAdd }

  Open mode of the ZIP file.
- enum CaseSensitivity { csDefault =0, csSensitive =1, csInsensitive =2 }

  Case sensitivity for the file names.

Public Member Functions

- QuaZip ()

  Constructs QuaZip object.
- QuaZip (const QString &zipName)

  Constructs QuaZip object associated with ZIP file zipName.
- QuaZip (QIODevice *ioDevice)

  Constructs QuaZip object associated with ZIP file represented by ioDevice.
- ~QuaZip ()

  Destroys QuaZip object.
- bool open (Mode mode, zlib_filefunc_def *ioApi=NULL)

  Opens ZIP file.
- void close ()

  Closes ZIP file.
- void setFileNameCodec (QTextCodec *fileNameCodec)

  Sets the codec used to encode/decode file names inside archive.
- void setFileNameCodec (const char *fileNameCodecName)

Sets the codec used to encode/decode file names inside archive.

- QTextCodec * getFileNameCodec () const

  Returns the codec used to encode/decode comments inside archive.
- void setCommentCodec (QTextCodec *commentCodec)

  Sets the codec used to encode/decode comments inside archive.
- void setCommentCodec (const char *commentCodecName)

  Sets the codec used to encode/decode comments inside archive.
- QTextCodec * getCommentCodec () const

  Returns the codec used to encode/decode comments inside archive.
- QString getZipName () const

  Returns the name of the ZIP file.
- void setZipName (const QString &zipName)

  Sets the name of the ZIP file.
- QIODevice * getIoDevice () const

  Returns the device representing this ZIP file.
- void setIoDevice (QIODevice *ioDevice)

  Sets the device representing the ZIP file.
- Mode getMode () const

  Returns the mode in which ZIP file was opened.
- bool isOpen () const

  Returns `true` if ZIP file is open, `false` otherwise.
- int getZipError () const

  Returns the error code of the last operation.
- int getEntriesCount () const

  Returns number of the entries in the ZIP central directory.
- QString getComment () const

  Returns global comment in the ZIP file.
- void setComment (const QString &comment)

  Sets the global comment in the ZIP file.
- bool goToFirstFile ()

  Sets the current file to the first file in the archive.
- bool goToNextFile ()

  Sets the current file to the next file in the archive.
- bool setCurrentFile (const QString &fileName, CaseSensitivity cs=csDefault)

  Sets current file by its name.
- bool hasCurrentFile () const

  Returns `true` if the current file has been set.
- bool getCurrentFileInfo (QuaZipFileInfo *info) const

  Retrieves information about the current file.
- bool getCurrentFileInfo (QuaZipFileInfo64 *info) const

  Retrieves information about the current file.
- QString getCurrentFileName () const

  Returns the current file name.
- unzFile getUnzFile ()

  Returns `unzFile` handle.
- zipFile getZipFile ()

  Returns `zipFile` handle.
- void setDataDescriptorWritingEnabled (bool enabled)

  Changes the data descriptor writing mode.
- bool isDataDescriptorWritingEnabled () const

  Returns the data descriptor default writing mode.

- QStringList getFileNameList () const

  Returns a list of files inside the archive.
- QList< QuaZipFileInfo > getFileInfoList () const

  Returns information list about all files inside the archive.
- QList< QuaZipFileInfo64 > getFileInfoList64 () const

  Returns information list about all files inside the archive.
- void setZip64Enabled (bool zip64)

  Enables the zip64 mode.
- bool isZip64Enabled () const

  Returns whether the zip64 mode is enabled.
- bool isAutoClose () const

  Returns the auto-close flag.
- void setAutoClose (bool autoClose) const

  Sets or unsets the auto-close flag.

Static Public Member Functions

- static Qt::CaseSensitivity convertCaseSensitivity (CaseSensitivity cs)

  Returns the actual case sensitivity for the specified QuaZIP one.
- static void setDefaultFileNameCodec (QTextCodec *codec)

  Sets the default file name codec to use.
- static void setDefaultFileNameCodec (const char *codecName)

  This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Equivalent to calling setDefltFileNameCodec(QTextCodec::codecForName(codecName)).

Private Member Functions

- QuaZip (const QuaZip &that)
- QuaZip & operator= (const QuaZip &that)

Private Attributes

- QuaZipPrivate * p

Friends

- class QuaZipPrivate

12.71.1 Detailed Description

This class implements basic interface to the ZIP archive. It can be used to read table contents of the ZIP archive and retreiving information about the files inside it.

You can also use this class to open files inside archive by passing pointer to the instance of this class to the constructor of the QuaZipFile class. But see QuaZipFile::QuaZipFile(QuaZip*, QObject*) for the possible pitfalls.

This class is indended to provide interface to the ZIP subpackage of the ZIP/UNZIP package as well as to the UNZIP subpackage. But currently it supports only UNZIP.

The use of this class is simple - just create instance using constructor, then set ZIP archive file name using setFile() function (if you did not passed the name to the constructor), then open() and then use different functions to work with it! Well, if you are paranoid, you may also wish to call close before destructing the instance, to check for errors on close.

You may also use getUnzFile() and getZipFile() functions to get the ZIP archive handle and use it with ZIP/UNZIP package API directly.

This class supports localized file names inside ZIP archive, but you have to set up proper codec with setCodec() function. By default, locale codec will be used, which is probably ok for UNIX systems, but will almost certainly fail with ZIP archives created in Windows. This is because Windows ZIP programs have strange habit of using DOS encoding for file names in ZIP archives. For example, ZIP archive with cyrillic names created in Windows will have file names in `IBM866` encoding instead of `WINDOWS-1251`. I think that calling one function is not much trouble, but for true platform independency it would be nice to have some mechanism for file name encoding auto detection using locale information. Does anyone know a good way to do it?

Definition at line 84 of file quazip.h.

12.71.2 Member Enumeration Documentation

12.71.2.1 enum QuaZip::CaseSensitivity

This is what you specify when accessing files in the archive. Works perfectly fine with any characters thanks to Qt's great unicode support. This is different from ZIP/UNZIP API, where only US-ASCII characters was supported.

Enumerator

csDefault Default for platform. Case sensitive for UNIX, not for Windows.

csSensitive Case sensitive.

csInsensitive Case insensitive.

Definition at line 114 of file quazip.h.

```
114                         {
115       csDefault=0,
116       csSensitive=1,
117       csInsensitive=2
118     };
```

12.71.2.2 enum QuaZip::Constants

Enumerator

MAX_FILE_NAME_LENGTH Maximum file name length. Taken from `UNZ_MAXFILENAMEINZIP` constant in unzip.c.

Definition at line 88 of file quazip.h.

```
88                  {
89       MAX_FILE_NAME_LENGTH=256
92     };
```

12.71.2.3 enum QuaZip::Mode

Enumerator

mdNotOpen ZIP file is not open. This is the initial mode.

mdUnzip ZIP file is open for reading files inside it.

mdCreate ZIP file was created with open() call.

mdAppend ZIP file was opened in append mode. This refers to `APPEND_STATUS_CREATEAFTER` mode in ZIP/UNZIP package and means that zip is appended to some existing file what is useful when that file contains self-extractor code. This is obviously *not* what you whant to use to add files to the existing ZIP archive.

mdAdd ZIP file was opened for adding files in the archive.

Definition at line 94 of file quazip.h.

```
94              {
95       mdNotOpen,
96       mdUnzip,
97       mdCreate,
98       mdAppend,
106       mdAdd
107     };
```

12.71.3 Constructor & Destructor Documentation

12.71.3.1 QuaZip::QuaZip (const QuaZip & *that*) [private]

12.71.3.2 QuaZip::QuaZip ( )

Call setName() before opening constructed object.

Definition at line 193 of file quazip.cpp.

```
193               :
194   p(new QuaZipPrivate(this))
195 {
196 }
```

12.71.3.3 QuaZip::QuaZip (const QString & *zipName*)

Definition at line 198 of file quazip.cpp.

```
198                                    :
199   p(new QuaZipPrivate(this, zipName))
200 {
201 }
```

12.71.3.4 QuaZip::QuaZip (QIODevice * *ioDevice*)

The IO device must be seekable, otherwise an error will occur when opening.

Definition at line 203 of file quazip.cpp.

```
203                                  :
204   p(new QuaZipPrivate(this, ioDevice))
205 {
206 }
```

12.71.3.5 QuaZip::~QuaZip ( )

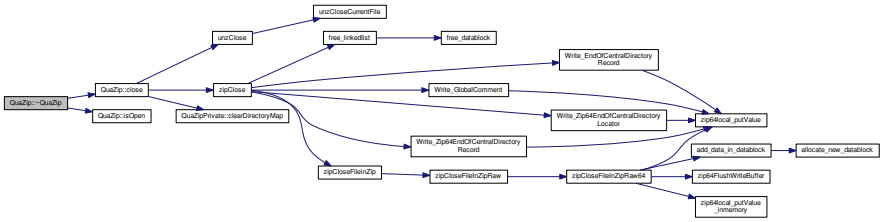
Calls close() if necessary.

Definition at line 208 of file quazip.cpp.

References close(), isOpen(), and p.

```
209 {
210   if(isOpen())
211     close();
212   delete p;
213 }
```

Here is the call graph for this function:



12.71.4 Member Function Documentation

12.71.4.1 void QuaZip::close ( )

Call getZipError() to determine if the close was successful.

If the file was opened by name, then the underlying QIODevice is closed and deleted.

If the underlying QIODevice was set explicitly using setIoDevice() or the appropriate constructor, then it is closed if the auto-close flag is set (which it is by default). Call setAutoClose() to clear the auto-close flag if this behavior is undesirable.

Since Qt 5.1, the QSaveFile was introduced. It breaks the QIODevice API by making close() private and crashing the application if it is called from the base class where it is public. It is an excellent example of poor design that illustrates why you should never ever break an is-a relationship between the base class and a subclass. QuaZIP works around this bug by checking if the QIODevice is an instance of QSaveFile, using qobject_cast<>, and if it is, calls QSaveFile::commit() instead of close(). It is a really ugly hack, but at least it makes your programs work instead of crashing. Note that if the auto-close flag is cleared, then this is a non-issue, and commit() isn't called.

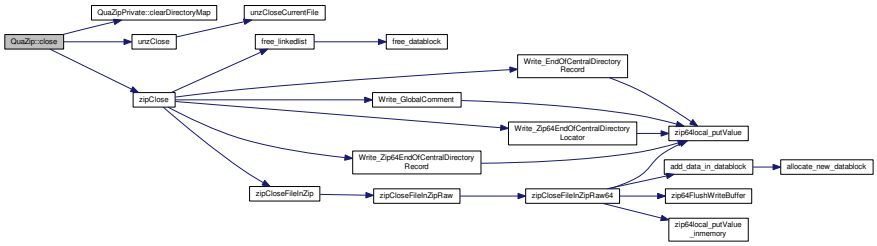
Definition at line 324 of file quazip.cpp.

References QuaZipPrivate::clearDirectoryMap(), QuaZipPrivate::comment, QuaZipPrivate::commentCodec, QuaZipPrivate::ioDevice, mdAdd, mdAppend, mdCreate, mdNotOpen, mdUnzip, QuaZipPrivate::mode, p, UNZ_OK, unzClose(), zipClose(), QuaZipPrivate::zipError, and QuaZipPrivate::zipName.

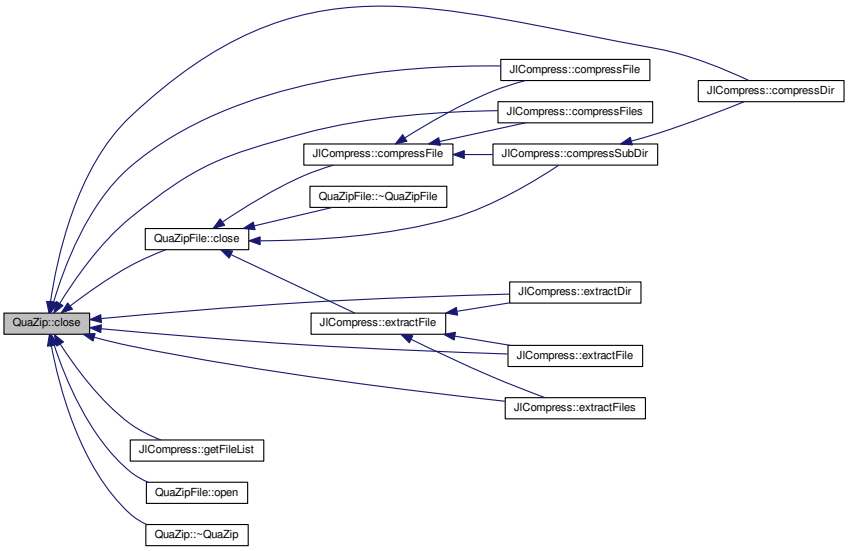
Referenced by QuaZipFile::close(), JlCompress::compressDir(), JlCompress::compressFile(), JlCompress::compressFiles(), JlCompress::extractDir(), JlCompress::extractFile(), JlCompress::extractFiles(), JlCompress::getFileList(), QuaZipFile::open(), and ~QuaZip().

```
325 {
326   p->zipError=UNZ_OK;
327   switch(p->mode) {
328     case mdNotOpen:
329       qWarning("QuaZip::close(): ZIP is not open");
330       return;
331     case mdUnzip:
332       p->zipError=unzClose(p->unzFile_f);
333       break;
334     case mdCreate:
335     case mdAppend:
336     case mdAdd:
337       p->zipError=zipClose(p->zipFile_f,
338           p->comment.isNull() ? NULL
339           : p->commentCodec->fromUnicode(p->comment).constData());
340       break;
341     default:
342       qWarning("QuaZip::close(): unknown mode: %d", (int)p->mode);
343       return;
344   }
345   // opened by name, need to delete the internal IO device
346   if (!p->zipName.isEmpty()) {
347       delete p->ioDevice;
348       p->ioDevice = NULL;
349   }
350   p->clearDirectoryMap();
351   if(p->zipError==UNZ_OK)
352     p->mode=mdNotOpen;
353 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.2 Qt::CaseSensitivity QuaZip::convertCaseSensitivity (QuaZip::CaseSensitivity *cs*) [static]

Parameters

cs	The value to convert.

Returns

If CaseSensitivity::csDefault, then returns the default file name case sensitivity for the platform. Otherwise, just returns the appropriate value from the Qt::CaseSensitivity enum.

Definition at line 747 of file quazip.cpp.

References csDefault, and csSensitive.

Referenced by QuaZipDir::exists(), and setCurrentFile().

```
748 {
749   if (cs == csDefault) {
750 #ifdef Q_WS_WIN
751       return Qt::CaseInsensitive;
752 #else
```

```
753       return Qt::CaseSensitive;
754 #endif
755   } else {
756       return cs == csSensitive ? Qt::CaseSensitive : Qt::CaseInsensitive;
757   }
758 }
```

Here is the caller graph for this function:

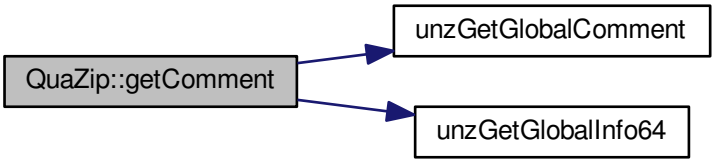
12.71.4.3 QString QuaZip::getComment ( ) const

Definition at line 389 of file quazip.cpp.

References QuaZipPrivate::commentCodec, mdUnzip, QuaZipPrivate::mode, p, unz_global_info64_s::size_comment, UNZ_OK, unzGetGlobalComment(), unzGetGlobalInfo64(), and QuaZipPrivate::zipError.

```
390 {
391   QuaZip *fakeThis=(QuaZip*)this; // non-const
392   fakeThis->p->zipError=UNZ_OK;
393   if(p->mode!=mdUnzip) {
394     qWarning("QuaZip::getComment(): ZIP is not open in mdUnzip mode");
395     return QString();
396   }
397   unz_global_info64 globalInfo;
398   QByteArray comment;
399   if((fakeThis->p->zipError=unzGetGlobalInfo64(p->unzFile_f, &globalInfo))!=
      UNZ_OK)
400     return QString();
401   comment.resize(globalInfo.size_comment);
402   if((fakeThis->p->zipError=unzGetGlobalComment(p->unzFile_f, comment.data(),
      comment.size())) < 0)
403     return QString();
404   fakeThis->p->zipError = UNZ_OK;
405   return p->commentCodec->toUnicode(comment);
406 }
```

Here is the call graph for this function:



12.71.4.4 QTextCodec * QuaZip::getCommentCodec ( ) const

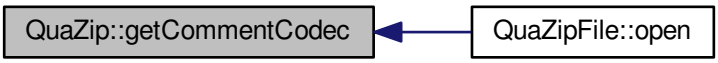
Definition at line 597 of file quazip.cpp.

References QuaZipPrivate::commentCodec, and p.

Referenced by QuaZipFile::open().

```
598 {
599   return p->commentCodec;
600 }
```

Here is the caller graph for this function:



12.71.4.5 bool QuaZip::getCurrentFileInfo (QuaZipFileInfo * *info*) const

Fills the structure pointed by *info*. Returns `true` on success, `false` otherwise. In the latter case structure pointed by *info* remains untouched. If there was an error, getZipError() returns error code.

Should be used only in QuaZip::mdUnzip mode.

Does nothing and returns `false` in any of the following cases.

- ZIP is not open;
- ZIP does not have current file.

In both cases getZipError() returns `UNZ_OK` since there is no ZIP/UNZIP API call.

This overload doesn't support zip64, but will work OK on zip64 archives except that if one of the sizes (compressed or uncompressed) is greater than 0xFFFFFFFFu, it will be set to exactly 0xFFFFFFFFu.

See Also

getCurrentFileInfo(QuaZipFileInfo64* info)const
QuaZipFileInfo64::toQuaZipFileInfo(QuaZipFileInfo&)const

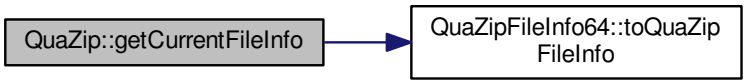
Definition at line 492 of file quazip.cpp.

References QuaZipFileInfo64::toQuaZipFileInfo().

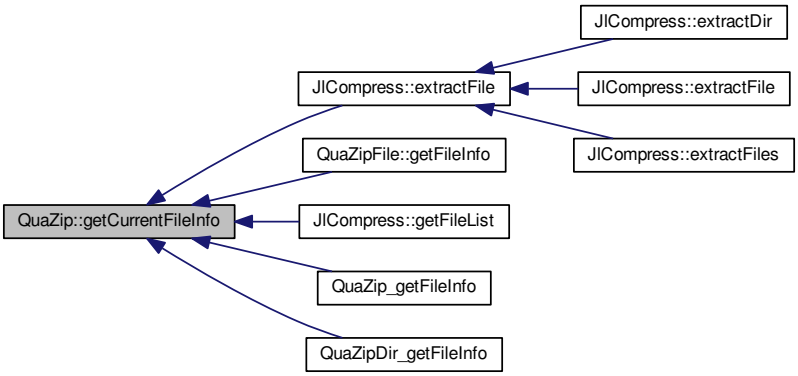
Referenced by JlCompress::extractFile(), QuaZipFile::getFileInfo(), JlCompress::getFileList(), QuaZip_getFileInfo(), and QuaZipDir_getFileInfo().

```
493 {
494     QuaZipFileInfo64 info64;
495     if (info == NULL) { // Very unlikely because of the overloads
496         return false;
497     }
498     if (getCurrentFileInfo(&info64)) {
499         info64.toQuaZipFileInfo(*info);
500         return true;
501     } else {
502         return false;
503     }
504 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.6 bool QuaZip::getCurrentFileInfo (QuaZipFileInfo64 * *info*) const

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

This function supports zip64. If the archive doesn't use zip64, it is completely equivalent to getCurrentFileInfo(QuaZipFileInfo* info) except for the argument type.

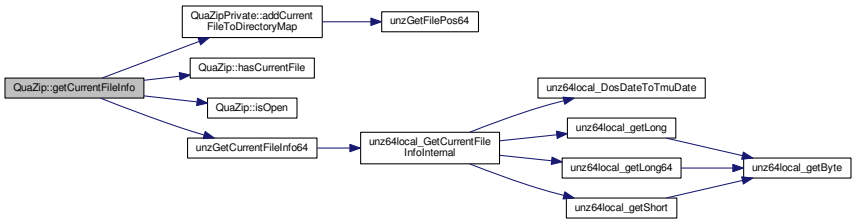
See Also

Definition at line 506 of file quazip.cpp.

References QuaZipPrivate::addCurrentFileToDirectoryMap(), QuaZipFileInfo64::comment, QuaZipPrivate::commentCodec, unz_file_info64_s::compressed_size, QuaZipFileInfo64::compressedSize, unz_file_info64_s::compression_method, QuaZipFileInfo64::crc, unz_file_info64_s::crc, QuaZipFileInfo64::dateTime, unz_file_info64_s::disk_num_start, QuaZipFileInfo64::diskNumberStart, unz_file_info64_s::external_fa, QuaZipFileInfo64::externalAttr, QuaZipFileInfo64::extra, QuaZipPrivate::fileNameCodec, unz_file_info64_s::flag, QuaZipFileInfo64::flags, hasCurrentFile(), unz_file_info64_s::internal_fa, QuaZipFileInfo64::internalAttr, isOpen(), mdUnzip, QuaZipFileInfo64::method, QuaZipPrivate::mode, QuaZipFileInfo64::name, p, unz_file_info64_s::size_file_comment, unz_file_info64_s::size_file_extra, unz_file_info64_s::size_filename, tm_unz_s::tm_hour, tm_unz_s::tm_mday, tm_unz_s::tm_min, tm_unz_s::tm_mon, tm_unz_s::tm_sec, tm_unz_s::tm_year, unz_file_info64_s::tmu_date, unz_file_info64_s::uncompressed_size, QuaZipFileInfo64::uncompressedSize, UNZ_OK, unzGetCurrentFileInfo64(), unz_file_info64_s::version, unz_file_info64_s::version_needed, QuaZipFileInfo64::versionCreated, QuaZipFileInfo64::versionNeeded, and QuaZipPrivate::zipError.

```
507 {
508   QuaZip *fakeThis=(QuaZip*)this; // non-const
509   fakeThis->p->zipError=UNZ_OK;
510   if(p->mode!=mdUnzip) {
511     qWarning("QuaZip::getCurrentFileInfo(): ZIP is not open in mdUnzip mode");
512     return false;
513   }
514   unz_file_info64 info_z;
515   QByteArray fileName;
516   QByteArray extra;
517   QByteArray comment;
518   if(info==NULL) return false;
519   if(!isOpen()||!hasCurrentFile()) return false;
520   if((fakeThis->p->zipError=unzGetCurrentFileInfo64(
      p->unzFile_f, &info_z, NULL, 0, NULL, 0, NULL, 0))!=UNZ_OK)
521     return false;
522   fileName.resize(info_z.size_filename);
523   extra.resize(info_z.size_file_extra);
524   comment.resize(info_z.size_file_comment);
525   if((fakeThis->p->zipError=unzGetCurrentFileInfo64(
      p->unzFile_f, NULL,
526       fileName.data(), fileName.size(),
527       extra.data(), extra.size(),
528       comment.data(), comment.size()))!=UNZ_OK)
529     return false;
530   info->versionCreated=info_z.version;
531   info->versionNeeded=info_z.version_needed;
532   info->flags=info_z.flag;
533   info->method=info_z.compression_method;
534   info->crc=info_z.crc;
535   info->compressedSize=info_z.compressed_size;
536   info->uncompressedSize=info_z.uncompressed_size;
537   info->diskNumberStart=info_z.disk_num_start;
538   info->internalAttr=info_z.internal_fa;
539   info->externalAttr=info_z.external_fa;
540   info->name=p->fileNameCodec->toUnicode(fileName);
541   info->comment=p->commentCodec->toUnicode(comment);
542   info->extra=extra;
543   info->dateTime=QDateTime(
544       QDate(info_z.tmu_date.tm_year, info_z.tmu_date.
      tm_mon+1, info_z.tmu_date.tm_mday),
545       QTime(info_z.tmu_date.tm_hour, info_z.tmu_date.
      tm_min, info_z.tmu_date.tm_sec));
546   // Add to directory map
547   p->addCurrentFileToDirectoryMap(info->name);
548   return true;
549 }
```

Here is the call graph for this function:



12.71.4.7 QString QuaZip::getCurrentFileName ( ) const

Equivalent to calling getCurrentFileInfo() and then getting `name` field of the QuaZipFileInfo structure, but faster and more convenient.

Should be used only in QuaZip::mdUnzip mode.

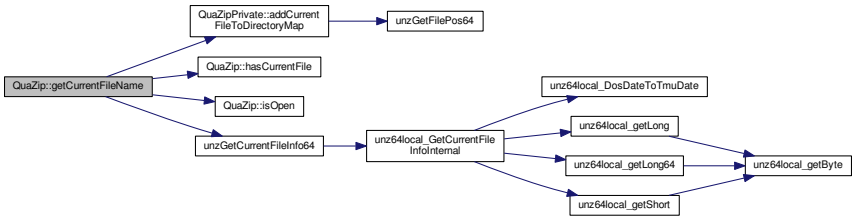
Definition at line 551 of file quazip.cpp.

References QuaZipPrivate::addCurrentFileToDirectoryMap(), QuaZipPrivate::fileNameCodec, hasCurrentFile(), isOpen(), MAX_FILE_NAME_LENGTH, mdUnzip, QuaZipPrivate::mode, p, UNZ_OK, unzGetCurrentFileInfo64(), and QuaZipPrivate::zipError.

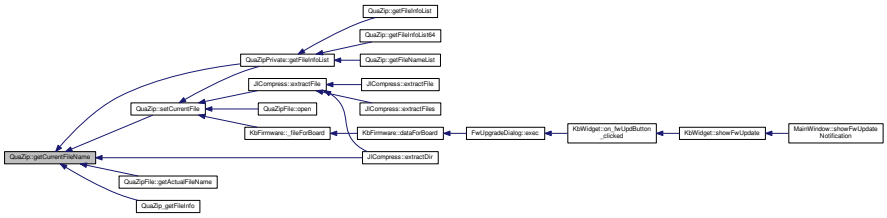
Referenced by JlCompress::extractDir(), QuaZipFile::getActualFileName(), QuaZipPrivate::getFileInfoList(), QuaZip_getFileInfo(), and setCurrentFile().

```
552 {
553   QuaZip *fakeThis=(QuaZip*)this; // non-const
554   fakeThis->p->zipError=UNZ_OK;
555   if(p->mode!=mdUnzip) {
556     qWarning("QuaZip::getCurrentFileName(): ZIP is not open in mdUnzip mode");
557     return QString();
558   }
559   if(!isOpen()||!hasCurrentFile()) return QString();
560   QByteArray fileName(MAX_FILE_NAME_LENGTH, 0);
561   if((fakeThis->p->zipError=unzGetCurrentFileInfo64(
      p->unzFile_f, NULL, fileName.data(), fileName.size(),
562       NULL, 0, NULL, 0))!=UNZ_OK)
563     return QString();
564   QString result = p->fileNameCodec->toUnicode(fileName.constData());
565   if (result.isEmpty())
566       return result;
567   // Add to directory map
568   p->addCurrentFileToDirectoryMap(result);
569   return result;
570 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.8 int QuaZip::getEntriesCount ( ) const

Returns negative error code in the case of error. The same error code will be returned by subsequent getZipError() call.

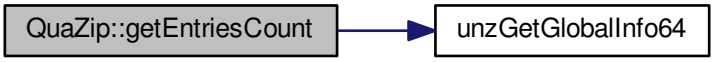
Definition at line 375 of file quazip.cpp.

References mdUnzip, QuaZipPrivate::mode, unz_global_info64_s::number_entry, p, UNZ_OK, unzGetGlobalInfo64(), and QuaZipPrivate::zipError.

```
376 {
377   QuaZip *fakeThis=(QuaZip*)this; // non-const
378   fakeThis->p->zipError=UNZ_OK;
379   if(p->mode!=mdUnzip) {
380     qWarning("QuaZip::getEntriesCount(): ZIP is not open in mdUnzip mode");
381     return -1;
```

```
382   }
383   unz_global_info64 globalInfo;
384   if((fakeThis->p->zipError=unzGetGlobalInfo64(p->unzFile_f, &globalInfo))!=
      UNZ_OK)
385     return p->zipError;
386   return (int)globalInfo.number_entry;
387 }
```

Here is the call graph for this function:



12.71.4.9 QList< QuaZipFileInfo > QuaZip::getFileInfoList ( ) const

Returns

A list of QuaZipFileInfo objects or an empty list if there was an error or if the archive is empty (call getZipError() to figure out which).

This function doesn't support zip64, but will still work with zip64 archives, converting results using QuaZipFileInfo64-::toQuaZipFileInfo(). If all file sizes are below 4 GB, it will work just fine.

See Also

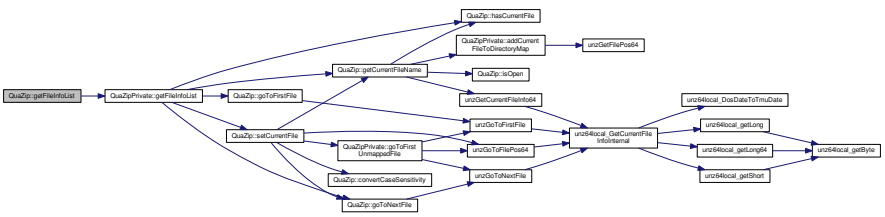
getFileNameList()
getFileInfoList64()

Definition at line 729 of file quazip.cpp.

References QuaZipPrivate::getFileInfoList(), and p.

```
730 {
731     QList<QuaZipFileInfo> list;
732     if (p->getFileInfoList(&list))
733         return list;
734     else
735         return QList<QuaZipFileInfo>();
736 }
```

Here is the call graph for this function:



12.71.4.10 QList< QuaZipFileInfo64 > QuaZip::getFileInfoList64 ( ) const

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

This function supports zip64.

See Also

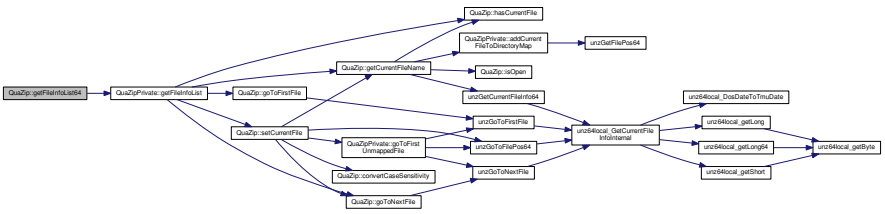
getFileNameList()
getFileInfoList()

Definition at line 738 of file quazip.cpp.

References QuaZipPrivate::getFileInfoList(), and p.

```
739 {
740     QList<QuaZipFileInfo64> list;
741     if (p->getFileInfoList(&list))
742         return list;
743     else
744         return QList<QuaZipFileInfo64>();
745 }
```

Here is the call graph for this function:



12.71.4.11 QTextCodec * QuaZip::getFileNameCodec ( ) const

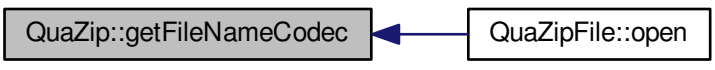
Definition at line 582 of file quazip.cpp.

References QuaZipPrivate::fileNameCodec, and p.

Referenced by QuaZipFile::open().

```
583 {
584   return p->fileNameCodec;
585 }
```

Here is the caller graph for this function:



12.71.4.12 QStringList QuaZip::getFileNameList ( ) const

Returns

A list of file names or an empty list if there was an error or if the archive is empty (call getZipError() to figure out which).

See Also

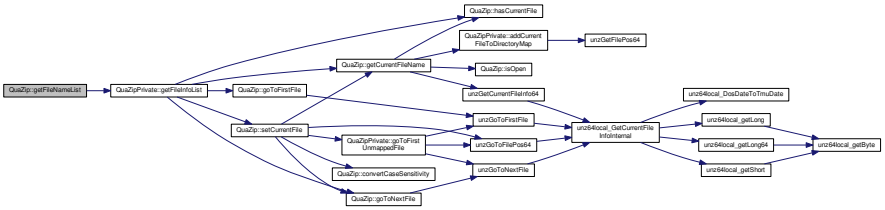
getFileInfoList()

Definition at line 720 of file quazip.cpp.

References QuaZipPrivate::getFileInfoList(), and p.

```
721 {
722     QStringList list;
723     if (p->getFileInfoList(&list))
724         return list;
725     else
726         return QStringList();
727 }
```

Here is the call graph for this function:



12.71.4.13 QIODevice * QuaZip::getIoDevice ( ) const

Returns null string if no device has been set explicitly, for example when opening a ZIP file by name.

See Also

setIoDevice(), getZipName(), setZipName()

Definition at line 607 of file quazip.cpp.

References QuaZipPrivate::ioDevice, p, and QuaZipPrivate::zipName.

```
608 {
609   if (!p->zipName.isEmpty()) // opened by name, using an internal QIODevice
610     return NULL;
611   return p->ioDevice;
612 }
```

12.71.4.14 QuaZip::Mode QuaZip::getMode ( ) const

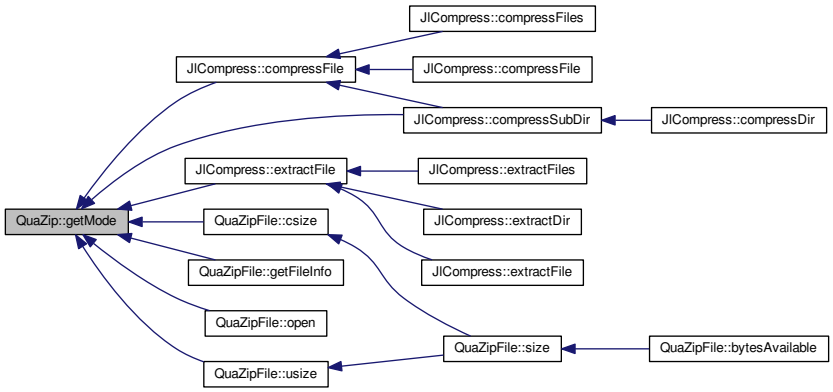
Definition at line 614 of file quazip.cpp.

References QuaZipPrivate::mode, and p.

Referenced by JlCompress::compressFile(), JlCompress::compressSubDir(), QuaZipFile::csize(), JlCompress::extractFile(), QuaZipFile::getFileInfo(), QuaZipFile::open(), and QuaZipFile::usize().

```
615 {
616   return p->mode;
617 }
```

Here is the caller graph for this function:



12.71.4.15 unzFile QuaZip::getUnzFile ( )

You can use this handle to directly call UNZIP part of the ZIP/UNZIP package functions (see unzip.h).

Warning

> When using the handle returned by this function, please keep in mind that QuaZip class is unable to detect any changes you make in the ZIP file state (e. g. changing current file, or closing the handle). So please do not do anything with this handle that is possible to do with the functions of this class. Or at least return the handle in the original state before calling some another function of this class (including implicit destructor calls and calls from the QuaZipFile objects that refer to this QuaZip instance!). So if you have changed the current file in the ZIP archive - then change it back or you may experience some strange behavior or even crashes.

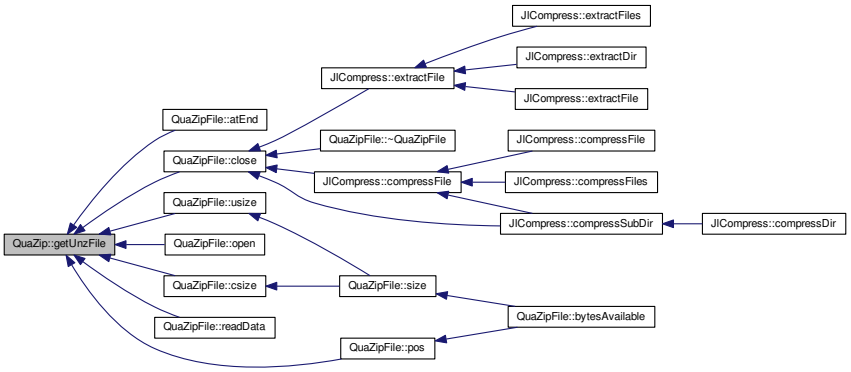
Definition at line 639 of file quazip.cpp.

References p.

Referenced by QuaZipFile::atEnd(), QuaZipFile::close(), QuaZipFile::csize(), QuaZipFile::open(), QuaZipFile::pos(), QuaZipFile::readData(), and QuaZipFile::usize().

```
640 {
641   return p->unzFile_f;
642 }
```

Here is the caller graph for this function:



12.71.4.16 int QuaZip::getZipError ( ) const

Returns `UNZ_OK` if the last operation was successful.

Error code resets to `UNZ_OK` every time you call any function that accesses something inside ZIP archive, even if it is `const` (like getEntriesCount()). open() and close() calls reset error code too. See documentation for the specific functions for details on error detection.

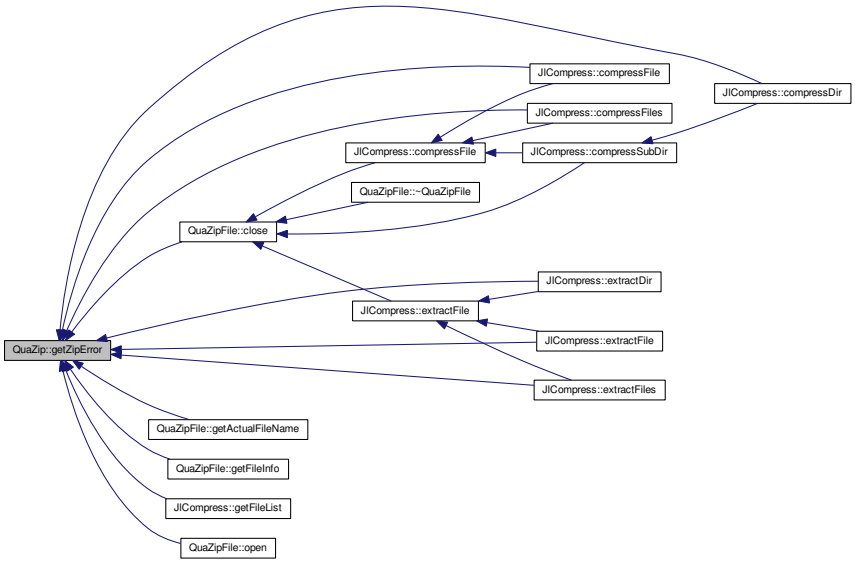
Definition at line 624 of file quazip.cpp.

References p, and QuaZipPrivate::zipError.

Referenced by QuaZipFile::close(), JlCompress::compressDir(), JlCompress::compressFile(), JlCompress::compressFiles(), JlCompress::extractDir(), JlCompress::extractFile(), JlCompress::extractFiles(), QuaZipFile::getActualFileName(), QuaZipFile::getFileInfo(), JlCompress::getFileList(), and QuaZipFile::open().

```
625 {
626   return p->zipError;
627 }
```

Here is the caller graph for this function:



12.71.4.17 zipFile QuaZip::getZipFile ( )

You can use this handle to directly call ZIP part of the ZIP/UNZIP package functions (see zip.h). Warnings about the getUnzFile() function also apply to this function.

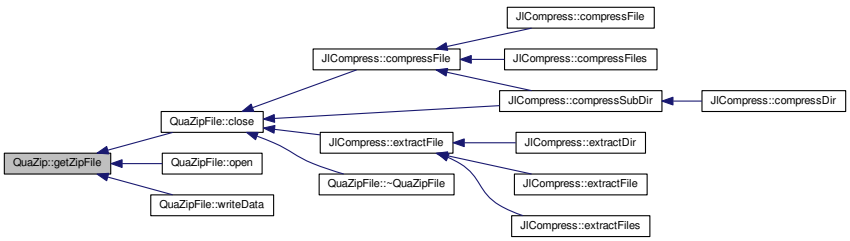
Definition at line 644 of file quazip.cpp.

References p.

Referenced by QuaZipFile::close(), QuaZipFile::open(), and QuaZipFile::writeData().

```
645 {
646   return p->zipFile_f;
647 }
```

Here is the caller graph for this function:



12.71.4.18 QString QuaZip::getZipName ( ) const

Returns null string if no ZIP file name has been set, for example when the QuaZip instance is set up to use a QIODevice instead.

See Also

setZipName(), setIoDevice(), getIoDevice()

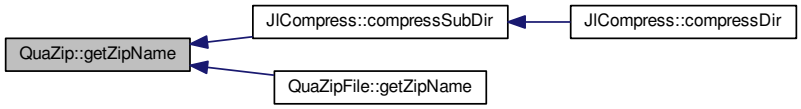
Definition at line 602 of file quazip.cpp.

References p, and QuaZipPrivate::zipName.

Referenced by JlCompress::compressSubDir(), and QuaZipFile::getZipName().

```
603 {
604   return p->zipName;
605 }
```

Here is the caller graph for this function:



12.71.4.19 bool QuaZip::goToFirstFile ( )

Returns `true` on success, `false` otherwise. Call getZipError() to get the error code.

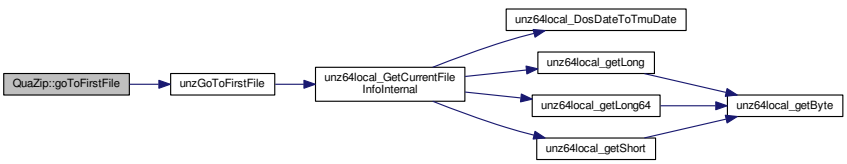
Definition at line 466 of file quazip.cpp.

References QuaZipPrivate::hasCurrentFile_f, mdUnzip, QuaZipPrivate::mode, p, UNZ_OK, unzGoToFirstFile(), and QuaZipPrivate::zipError.

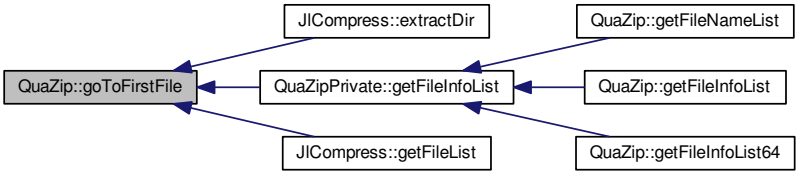
Referenced by JlCompress::extractDir(), QuaZipPrivate::getFileInfoList(), and JlCompress::getFileList().

```
467 {
468   p->zipError=UNZ_OK;
469   if(p->mode!=mdUnzip) {
470     qWarning("QuaZip::goToFirstFile(): ZIP is not open in mdUnzip mode");
471     return false;
472   }
473   p->zipError=unzGoToFirstFile(p->unzFile_f);
474   p->hasCurrentFile_f=p->zipError==UNZ_OK;
475   return p->hasCurrentFile_f;
476 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.20 bool QuaZip::goToNextFile ( )

Returns `true` on success, `false` otherwise. Call getZipError() to determine if there was an error.

Should be used only in QuaZip::mdUnzip mode.

Note

If the end of file was reached, getZipError() will return `UNZ_OK` instead of `UNZ_END_OF_LIST_OF_FILE`. This is to make things like this easier:

```
for(bool more=zip.goToFirstFile(); more; more=zip.goToNextFile()) {
  // do something
}
if(zip.getZipError()==UNZ_OK) {
  // ok, there was no error
}
```

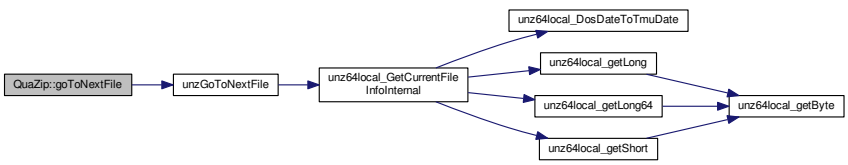
Definition at line 478 of file quazip.cpp.

References QuaZipPrivate::hasCurrentFile_f, mdUnzip, QuaZipPrivate::mode, p, UNZ_END_OF_LIST_OF_FILE, UNZ_OK, unzGoToNextFile(), and QuaZipPrivate::zipError.

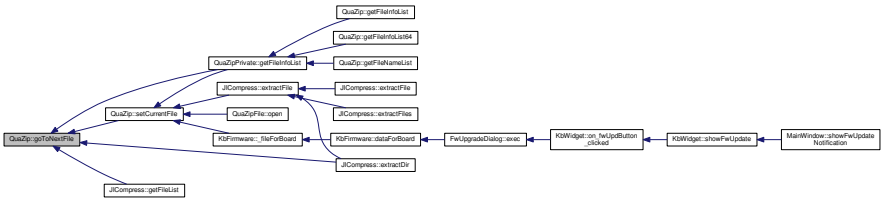
Referenced by JlCompress::extractDir(), QuaZipPrivate::getFileInfoList(), JlCompress::getFileList(), and setCurrentFile().

```
479 {
480   p->zipError=UNZ_OK;
481   if(p->mode!=mdUnzip) {
482     qWarning("QuaZip::goToFirstFile(): ZIP is not open in mdUnzip mode");
483     return false;
484   }
485   p->zipError=unzGoToNextFile(p->unzFile_f);
486   p->hasCurrentFile_f=p->zipError==UNZ_OK;
487   if(p->zipError==UNZ_END_OF_LIST_OF_FILE)
488     p->zipError=UNZ_OK;
489   return p->hasCurrentFile_f;
490 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.21 bool QuaZip::hasCurrentFile ( ) const

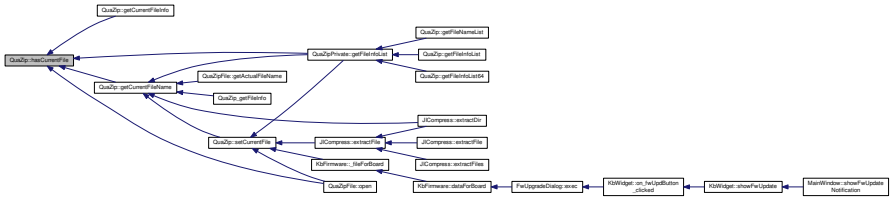
Definition at line 634 of file quazip.cpp.

References QuaZipPrivate::hasCurrentFile_f, and p.

Referenced by getCurrentFileInfo(), getCurrentFileName(), QuaZipPrivate::getFileInfoList(), and QuaZipFile::open().

```
635 {
636   return p->hasCurrentFile_f;
637 }
```

Here is the caller graph for this function:



12.71.4.22 bool QuaZip::isAutoClose ( ) const

See Also

setAutoClose()

Definition at line 780 of file quazip.cpp.

References QuaZipPrivate::autoClose, and p.

```
781 {
782     return p->autoClose;
783 }
```

12.71.4.23 bool QuaZip::isDataDescriptorWritingEnabled ( ) const

See Also

setDataDescriptorWritingEnabled()

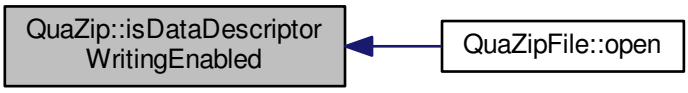
Definition at line 654 of file quazip.cpp.

References QuaZipPrivate::dataDescriptorWritingEnabled, and p.

Referenced by QuaZipFile::open().

```
655 {
656     return p->dataDescriptorWritingEnabled;
657 }
```

Here is the caller graph for this function:



12.71.4.24 bool QuaZip::isOpen ( ) const

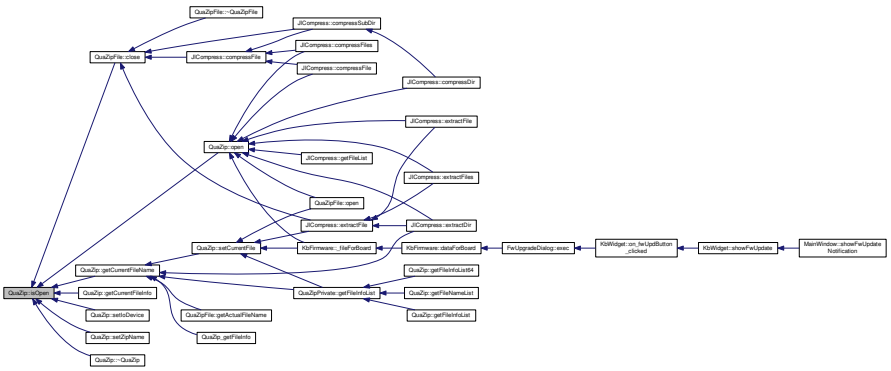
Definition at line 619 of file quazip.cpp.

References mdNotOpen, QuaZipPrivate::mode, and p.

Referenced by QuaZipFile::close(), getCurrentFileInfo(), getCurrentFileName(), open(), setIoDevice(), setZipName(), and ~QuaZip().

```
620 {
621   return p->mode!=mdNotOpen;
622 }
```

Here is the caller graph for this function:



12.71.4.25 bool QuaZip::isZip64Enabled ( ) const

Returns

`true` if and only if the zip64 mode is enabled.

See Also

setZip64Enabled()

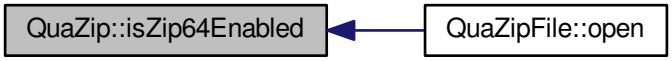
Definition at line 775 of file quazip.cpp.

References p, and QuaZipPrivate::zip64.

Referenced by QuaZipFile::open().

```
776 {
777     return p->zip64;
778 }
```

Here is the caller graph for this function:



12.71.4.26 bool QuaZip::open (Mode *mode*, zlib_filefunc_def * *ioApi* = NULL)

Argument *mode* specifies open mode of the ZIP archive. See Mode for details. Note that there is zipOpen2() function in the ZIP/UNZIP API which accepts *globalcomment* argument, but it does not use it anywhere, so this open() function does not have this argument. See setComment() if you need to set global comment.

If the ZIP file is accessed via explicitly set QIODevice, then this device is opened in the necessary mode. If the device was already opened by some other means, then QuaZIP checks if the open mode is compatible to the mode needed for the requested operation. If necessary, seeking is performed to position the device properly.

Returns

`true` if successful, `false` otherwise.

Note

ZIP/UNZIP API open calls do not return error code - they just return `NULL` indicating an error. But to make things easier, quazip.h header defines additional error code `UNZ_ERROROPEN` and getZipError() will return it if the open call of the ZIP/UNZIP API returns `NULL`.

Argument *ioApi* specifies IO function set for ZIP/UNZIP package to use. See unzip.h, zip.h and ioapi.h for details. Note that IO API for QuaZip is different from the original package. The file path argument was changed to be of type `voidpf`, and QuaZip passes a QIODevice pointer there. This QIODevice is either set explicitly via setIoDevice() or the QuaZip(QIODevice*) constructor, or it is created internally when opening the archive by its file name. The default API (qioapi.cpp) just delegates everything to the QIODevice API. Not only this allows to use a QIODevice instead of file name, but also has a nice side effect of raising the file size limit from 2G to 4G (in non-zip64 archives).

Note

If the zip64 support is needed, the ioApi argument *must* be NULL because due to the backwards compatibility issues it can be used to provide a 32-bit API only.
If the no-auto-close feature is used, then the *ioApi* argument *should* be NULL because the old API doesn't support the 'fake close' operation, causing slight memory leaks and other possible troubles (like closing the output device in case when an error occurs during opening).

In short: just forget about the *ioApi* argument and you'll be fine.

Definition at line 215 of file quazip.cpp.

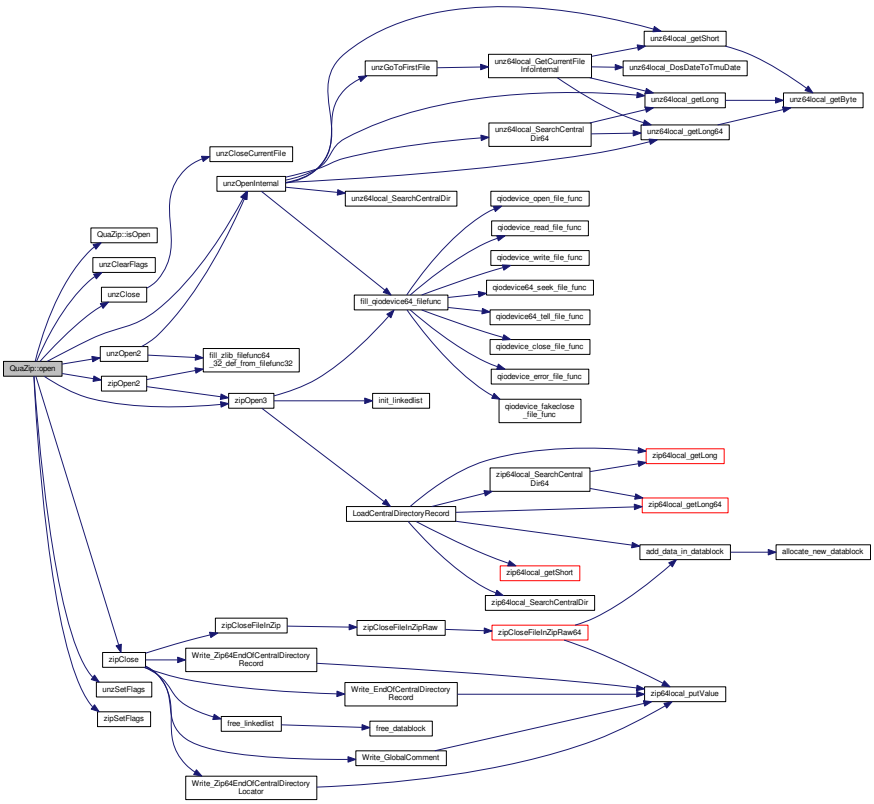
References APPEND_STATUS_ADDINZIP, APPEND_STATUS_CREATE, APPEND_STATUS_CREATEAFTER, QuaZipPrivate::autoClose, QuaZipPrivate::dataDescriptorWritingEnabled, QuaZipPrivate::ioDevice, isOpen(), mdAdd, mdAppend, mdCreate, mdUnzip, QuaZipPrivate::mode, p, UNZ_AUTO_CLOSE, UNZ_OK, UNZ_OPENERROR, unzClearFlags(), unzClose(), unzOpen2(), unzOpenInternal(), unzSetFlags(), ZIP_AUTO_CLOSE, ZIP_SEQUENTIAL, ZIP_WRITE_DATA_DESCRIPTOR, zipClose(), QuaZipPrivate::zipError, QuaZipPrivate::zipName, zipOpen2(), zipOpen3(), and zipSetFlags().

Referenced by KbFirmware::_fileForBoard(), JlCompress::compressDir(), JlCompress::compressFile(), JlCompress::compressFiles(), JlCompress::extractDir(), JlCompress::extractFile(), JlCompress::extractFiles(), JlCompress::getFileList(), and QuaZipFile::open().

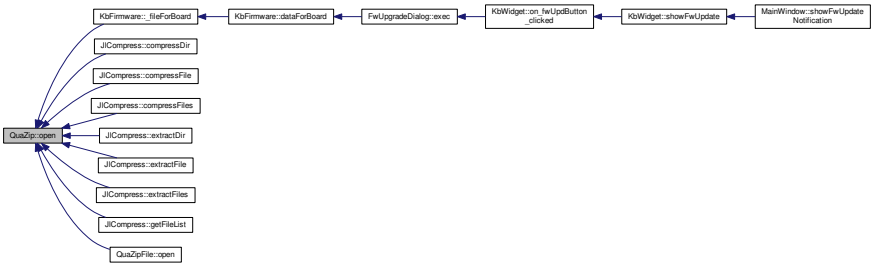
```
216 {
217   p->zipError=UNZ_OK;
218   if(isOpen()) {
219     qWarning("QuaZip::open(): ZIP already opened");
220     return false;
221   }
222   QIODevice *ioDevice = p->ioDevice;
223   if (ioDevice == NULL) {
224     if (p->zipName.isEmpty()) {
225       qWarning("QuaZip::open(): set either ZIP file name or IO device first");
226       return false;
227     } else {
228       ioDevice = new QFile(p->zipName);
229     }
230   }
231   unsigned flags = 0;
232   switch(mode) {
233     case mdUnzip:
234       if (ioApi == NULL) {
235           if (p->autoClose)
236               flags |= UNZ_AUTO_CLOSE;
237           p->unzFile_f=unzOpenInternal(ioDevice, NULL, 1, flags);
238       } else {
239           // QuaZIP pre-zip64 compatibility mode
240           p->unzFile_f=unzOpen2(ioDevice, ioApi);
241           if (p->unzFile_f != NULL) {
242               if (p->autoClose) {
243                   unzSetFlags(p->unzFile_f, UNZ_AUTO_CLOSE);
244               } else {
245                   unzClearFlags(p->unzFile_f, UNZ_AUTO_CLOSE);
246               }
247           }
248       }
249       if(p->unzFile_f!=NULL) {
250         if (ioDevice->isSequential()) {
251             unzClose(p->unzFile_f);
252             if (!p->zipName.isEmpty())
253                 delete ioDevice;
254             qWarning("QuaZip::open(): "
255                      "only mdCreate can be used with "
256                      "sequential devices");
257             return false;
258         }
259         p->mode=mode;
260         p->ioDevice = ioDevice;
261         return true;
262       } else {
263         p->zipError=UNZ_OPENERROR;
264         if (!p->zipName.isEmpty())
265           delete ioDevice;
266         return false;
267       }
268     case mdCreate:
269     case mdAppend:
270     case mdAdd:
271       if (ioApi == NULL) {
```

```
272             if (p->autoClose)
273                 flags |= ZIP_AUTO_CLOSE;
274             if (p->dataDescriptorWritingEnabled)
275                 flags |= ZIP_WRITE_DATA_DESCRIPTOR;
276             p->zipFile_f=zipOpen3(ioDevice,
277                 mode==mdCreate?APPEND_STATUS_CREATE:
278                 mode==mdAppend?APPEND_STATUS_CREATEAFTER:
279                 APPEND_STATUS_ADDINZIP,
280                 NULL, NULL, flags);
281         } else {
282             // QuaZIP pre-zip64 compatibility mode
283             p->zipFile_f=zipOpen2(ioDevice,
284                 mode==mdCreate?APPEND_STATUS_CREATE:
285                 mode==mdAppend?APPEND_STATUS_CREATEAFTER:
286                 APPEND_STATUS_ADDINZIP,
287                 NULL,
288                 ioApi);
289             if (p->zipFile_f != NULL) {
290                 zipSetFlags(p->zipFile_f, flags);
291             }
292         }
293         if(p->zipFile_f!=NULL) {
294           if (ioDevice->isSequential()) {
295               if (mode != mdCreate) {
296                   zipClose(p->zipFile_f, NULL);
297                   qWarning("QuaZip::open(): "
298                           "only mdCreate can be used with "
299                            "sequential devices");
300                   if (!p->zipName.isEmpty())
301                       delete ioDevice;
302                   return false;
303               }
304               zipSetFlags(p->zipFile_f, ZIP_SEQUENTIAL);
305           }
306           p->mode=mode;
307           p->ioDevice = ioDevice;
308           return true;
309         } else {
310           p->zipError=UNZ_OPENERROR;
311           if (!p->zipName.isEmpty())
312             delete ioDevice;
313           return false;
314         }
315       default:
316         qWarning("QuaZip::open(): unknown mode: %d", (int)mode);
317         if (!p->zipName.isEmpty())
318           delete ioDevice;
319         return false;
320         break;
321     }
322 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.27 QuaZip& QuaZip::operator= (const QuaZip & *that*) [private]

12.71.4.28 void QuaZip::setAutoClose (bool *autoClose*) const

By default, QuaZIP opens the underlying QIODevice when open() is called, and closes it when close() is called. In some cases, when the device is set explicitly using setIoDevice(), it may be desirable to leave the device open. If the auto-close flag is unset using this method, then the device isn't closed automatically if it was set explicitly.

If it is needed to clear this flag, it is recommended to do so before opening the archive because otherwise QuaZIP may close the device during the open() call if an error is encountered after the device is opened.

If the device was not set explicitly, but rather the setZipName() or the appropriate constructor was used to set the

ZIP file name instead, then the auto-close flag has no effect, and the internal device is closed nevertheless because there is no other way to close it.

See Also

isAutoClose()
setIoDevice()

Definition at line 785 of file quazip.cpp.

References QuaZipPrivate::autoClose, and p.

```
786 {
787     p->autoClose = autoClose;
788 }
```

12.71.4.29 void QuaZip::setComment (const QString & *comment*)

The comment will be written to the archive on close operation. QuaZip makes a distinction between a null QByteArray() comment and an empty "" comment in the QuaZip::mdAdd mode. A null comment is the default and it means "don't change the comment". An empty comment removes the original comment.

See Also

open()

Definition at line 629 of file quazip.cpp.

References QuaZipPrivate::comment, and p.

```
630 {
631   p->comment=comment;
632 }
```

12.71.4.30 void QuaZip::setCommentCodec (QTextCodec * *commentCodec*)

This codec defaults to locale codec, which is probably ok.

Definition at line 587 of file quazip.cpp.

References QuaZipPrivate::commentCodec, and p.

```
588 {
589   p->commentCodec=commentCodec;
590 }
```

12.71.4.31 void QuaZip::setCommentCodec (const char * *commentCodecName*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Equivalent to calling setCommentCodec(QTextCodec::codecForName(codecName));

Definition at line 592 of file quazip.cpp.

References QuaZipPrivate::commentCodec, and p.

```
593 {
594   p->commentCodec=QTextCodec::codecForName(commentCodecName);
595 }
```

12.71.4.32 bool QuaZip::setCurrentFile (const QString & *fileName*, CaseSensitivity *cs* = csDefault)

Returns `true` if successful, `false` otherwise. Argument *cs* specifies case sensitivity of the file name. Call getZipError() in the case of a failure to get error code.

This is not a wrapper to unzLocateFile() function. That is because I had to implement locale-specific case-insensitive comparison.

Here are the differences from the original implementation:

- If the file was not found, error code is `UNZ_OK`, not `UNZ_END_OF_LIST_OF_FILE` (see also goToNextFile()).
- If this function fails, it unsets the current file rather than resetting it back to what it was before the call.

If *fileName* is null string then this function unsets the current file and return `true`. Note that you should close the file first if it is open! See QuaZipFile::QuaZipFile(QuaZip*,QObject*) for the details.

Should be used only in QuaZip::mdUnzip mode.

See Also

setFileNameCodec(), CaseSensitivity

Definition at line 408 of file quazip.cpp.

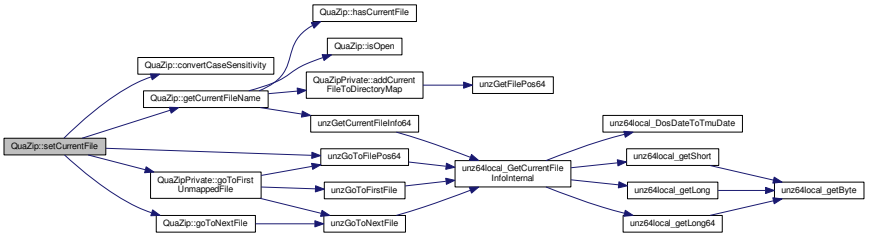
References convertCaseSensitivity(), current, QuaZipPrivate::directoryCaseInsensitive, QuaZipPrivate::directoryCaseSensitive, getCurrentFileName(), QuaZipPrivate::goToFirstUnmappedFile(), goToNextFile(), QuaZipPrivate::hasCurrentFile_f, MAX_FILE_NAME_LENGTH, mdUnzip, QuaZipPrivate::mode, p, unz64_file_pos_s::pos_in_zip_directory, UNZ_OK, UNZ_PARAMERROR, unzGoToFilePos64(), and QuaZipPrivate::zipError.

Referenced by KbFirmware::_fileForBoard(), JlCompress::extractFile(), QuaZipPrivate::getFileInfoList(), and QuaZipFile::open().

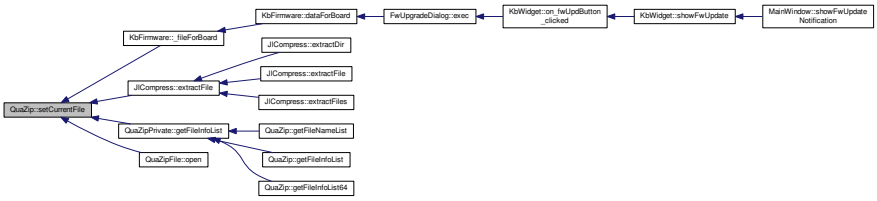
```
409 {
410   p->zipError=UNZ_OK;
411   if(p->mode!=mdUnzip) {
412     qWarning("QuaZip::setCurrentFile(): ZIP is not open in mdUnzip mode");
413     return false;
414   }
415   if(fileName.isEmpty()) {
416     p->hasCurrentFile_f=false;
417     return true;
418   }
419   // Unicode-aware reimplementation of the unzLocateFile function
420   if(p->unzFile_f==NULL) {
421     p->zipError=UNZ_PARAMERROR;
422     return false;
423   }
424   if(fileName.length()>MAX_FILE_NAME_LENGTH) {
425     p->zipError=UNZ_PARAMERROR;
426     return false;
427   }
428   // Find the file by name
429   bool sens = convertCaseSensitivity(cs) == Qt::CaseSensitive;
430   QString lower, current;
431   if(!sens) lower=fileName.toLower();
432   p->hasCurrentFile_f=false;
433
434   // Check the appropriate Map
435   unz64_file_pos fileDirPos;
436   fileDirPos.pos_in_zip_directory = 0;
437   if (sens) {
438       if (p->directoryCaseSensitive.contains(fileName))
439           fileDirPos = p->directoryCaseSensitive.value(fileName);
440   } else {
441       if (p->directoryCaseInsensitive.contains(lower))
442           fileDirPos = p->directoryCaseInsensitive.value(lower);
443   }
444
445   if (fileDirPos.pos_in_zip_directory != 0) {
446       p->zipError = unzGoToFilePos64(p->unzFile_f, &fileDirPos);
447       p->hasCurrentFile_f = p->zipError == UNZ_OK;
448   }
```

```
449
450   if (p->hasCurrentFile_f)
451       return p->hasCurrentFile_f;
452
453   // Not mapped yet, start from where we have got to so far
454   for(bool more=p->goToFirstUnmappedFile(); more; more=
      goToNextFile()) {
455     current=getCurrentFileName();
456     if(current.isEmpty()) return false;
457     if(sens) {
458       if(current==fileName) break;
459     } else {
460       if(current.toLower()==lower) break;
461     }
462   }
463   return p->hasCurrentFile_f;
464 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.71.4.33 void QuaZip::setDataDescriptorWritingEnabled (bool *enabled*)

According to the ZIP format specification, a file inside archive may have a data descriptor immediately following the file data. This is reflected by a special flag in the local file header and in the central directory. By default, QuaZIP sets this flag and writes the data descriptor unless both method and level were set to 0, in which case it operates in 1.0-compatible mode and never writes data descriptors.

By setting this flag to false, it is possible to disable data descriptor writing, thus increasing compatibility with archive readers that don't understand this feature of the ZIP file format.

Setting this flag affects all the QuaZipFile instances that are opened after this flag is set.

The data descriptor writing mode is enabled by default.

Note that if the ZIP archive is written into a QIODevice for which QIODevice::isSequential() returns `true`, then the data descriptor is mandatory and will be written even if this flag is set to false.

Parameters

enabled	If `true`, enable local descriptor writing, disable it otherwise.

See Also

QuaZipFile::isDataDescriptorWritingEnabled()

Definition at line 649 of file quazip.cpp.

References QuaZipPrivate::dataDescriptorWritingEnabled, and p.

```
650 {
651     p->dataDescriptorWritingEnabled = enabled;
652 }
```

12.71.4.34 void QuaZip::setDefaultFileNameCodec (QTextCodec * *codec*) [static]

The default codec is used by the constructors, so calling this function won't affect the QuaZip instances already created at that moment.

The codec specified here can be overriden by calling setFileNameCodec(). If neither function is called, QTextCodec-::codecForLocale() will be used to decode or encode file names. Use this function with caution if the application uses other libraries that depend on QuaZIP. Those libraries can either call this function by themselves, thus overriding your setting or can rely on the default encoding, thus failing mysteriously if you change it. For these reasons, it isn't recommended to use this function if you are developing a library, not an application. Instead, ask your library users to call it in case they need specific encoding.

In most cases, using setFileNameCodec() instead is the right choice. However, if you depend on third-party code that uses QuaZIP, then the reasons stated above can actually become a reason to use this function in case the third-party code in question fails because it doesn't understand the encoding you need and doesn't provide a way to specify it. This applies to the JlCompress class as well, as it was contributed and doesn't support explicit encoding parameters.

In short: use setFileNameCodec() when you can, resort to setDefaultFileNameCodec() when you don't have access to the QuaZip instance.

Parameters

codec	The codec to use by default. If NULL, resets to default.

Definition at line 760 of file quazip.cpp.

References QuaZipPrivate::defaultFileNameCodec.

Referenced by setDefaultFileNameCodec().

```
761 {
762     QuaZipPrivate::defaultFileNameCodec = codec;
763 }
```

Here is the caller graph for this function:

12.71.4.35 void QuaZip::setDefaultFileNameCodec (const char * *codecName*) [static]

Definition at line 765 of file quazip.cpp.

References setDefaultFileNameCodec().

```
766 {
767     setDefaultFileNameCodec(QTextCodec::codecForName(codecName));
768 }
```

Here is the call graph for this function:

12.71.4.36 void QuaZip::setFileNameCodec (QTextCodec * *fileNameCodec*)

This is necessary to access files in the ZIP archive created under Windows with non-latin characters in file names. For example, file names with cyrillic letters will be in `IBM866` encoding.

Definition at line 572 of file quazip.cpp.

References QuaZipPrivate::fileNameCodec, and p.

```
573 {
574   p->fileNameCodec=fileNameCodec;
575 }
```

12.71.4.37 void QuaZip::setFileNameCodec (const char * *fileNameCodecName*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Equivalent to calling setFileNameCodec(QTextCodec::codecForName(codecName));

Definition at line 577 of file quazip.cpp.

References QuaZipPrivate::fileNameCodec, and p.

```
578 {
579   p->fileNameCodec=QTextCodec::codecForName(fileNameCodecName);
580 }
```

12.71.4.38 void QuaZip::setIoDevice (QIODevice * *ioDevice*)

Does nothing if the ZIP file is open.

Does not reset error code returned by getZipError().

See Also

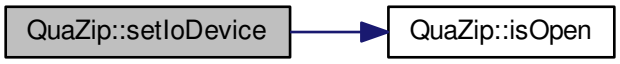
getIoDevice(), getZipName(), setZipName()

Definition at line 365 of file quazip.cpp.

References QuaZipPrivate::ioDevice, isOpen(), p, and QuaZipPrivate::zipName.

```
366 {
367   if(isOpen()) {
368     qWarning("QuaZip::setIoDevice(): ZIP is already open!");
369     return;
370   }
371   p->ioDevice = ioDevice;
372   p->zipName = QString();
373 }
```

Here is the call graph for this function:



12.71.4.39 void QuaZip::setZip64Enabled (bool *zip64*)

Parameters

zip64	If `true`, the zip64 mode is enabled, disabled otherwise.

Once this is enabled, all new files (until the mode is disabled again) will be created in the zip64 mode, thus enabling the ability to write files larger than 4 GB. By default, the zip64 mode is off due to compatibility reasons.

Note that this does not affect the ability to read zip64 archives in any way.

See Also

isZip64Enabled()

Definition at line 770 of file quazip.cpp.

References p, and QuaZipPrivate::zip64.

```
771 {
772     p->zip64 = zip64;
773 }
```

12.71.4.40 void QuaZip::setZipName (const QString & *zipName*)

Does nothing if the ZIP file is open.

Does not reset error code returned by getZipError().

See Also

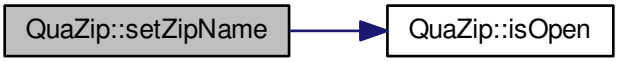
setIoDevice(), getIoDevice(), getZipName()

Definition at line 355 of file quazip.cpp.

References QuaZipPrivate::ioDevice, isOpen(), p, and QuaZipPrivate::zipName.

```
356 {
357   if(isOpen()) {
358     qWarning("QuaZip::setZipName(): ZIP is already open!");
359     return;
360   }
361   p->zipName=zipName;
362   p->ioDevice = NULL;
363 }
```

Here is the call graph for this function:



12.71.5 Friends And Related Function Documentation

12.71.5.1 friend class QuaZipPrivate [friend]

Definition at line 85 of file quazip.h.

12.71.6 Field Documentation

12.71.6.1 QuaZipPrivate* QuaZip::p [private]

Definition at line 129 of file quazip.h.

Referenced by close(), getComment(), getCommentCodec(), getCurrentFileInfo(), getCurrentFileName(), getEntriesCount(), getFileInfoList(), getFileInfoList64(), getFileNameCodec(), getFileNameList(), getIoDevice(), getMode(), getUnzFile(), getZipError(), getZipFile(), getZipName(), goToFirstFile(), goToNextFile(), hasCurrentFile(), isAutoClose(), isDataDescriptorWritingEnabled(), isOpen(), isZip64Enabled(), open(), setAutoClose(), setComment(), setCommentCodec(), setCurrentFile(), setDataDescriptorWritingEnabled(), setFileNameCodec(), setIoDevice(), setZip64Enabled(), setZipName(), and ~QuaZip().

The documentation for this class was generated from the following files:

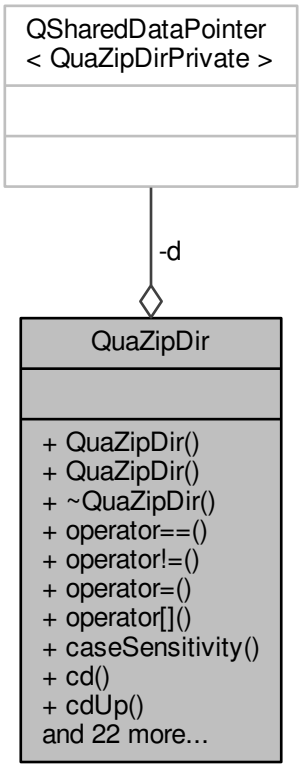
- src/ckb/quazip/quazip.h
- src/ckb/quazip/quazip.cpp

12.72 QuaZipDir Class Reference

Provides ZIP archive navigation.

`#include <src/ckb/quazip/quazipdir.h>`

Collaboration diagram for QuaZipDir:



Public Member Functions

- QuaZipDir (const QuaZipDir &that)

  The copy constructor.
- QuaZipDir (QuaZip ∗zip, const QString &dir=QString())

  Constructs a QuaZipDir instance pointing to the specified directory.
- ∼QuaZipDir ()

  Destructor.
- bool operator== (const QuaZipDir &that)

  The assignment operator.
- bool operator!= (const QuaZipDir &that)

  operator!=
- QuaZipDir & operator= (const QuaZipDir &that)

  operator==
- QString operator[] (int pos) const

  Returns the name of the entry at the specified position.
- QuaZip::CaseSensitivity caseSensitivity () const

  Returns the current case sensitivity mode.
- bool cd (const QString &dirName)

  Changes the 'current' directory.

- bool cdUp ()

  Goes up.
- uint count () const

  Returns the number of entries in the directory.
- QString dirName () const

  Returns the current directory name.
- QList< QuaZipFileInfo > entryInfoList (const QStringList &nameFilters, QDir::Filters filters=QDir::NoFilter, QDir::SortFlags sort=QDir::NoSort) const

  Returns the list of the entries in the directory.
- QList< QuaZipFileInfo > entryInfoList (QDir::Filters filters=QDir::NoFilter, QDir::SortFlags sort=QDir::NoSort) const

  Returns the list of the entries in the directory.
- QList< QuaZipFileInfo64 > entryInfoList64 (const QStringList &nameFilters, QDir::Filters filters=QDir::NoFilter, QDir::SortFlags sort=QDir::NoSort) const

  Returns the list of the entries in the directory with zip64 support.
- QList< QuaZipFileInfo64 > entryInfoList64 (QDir::Filters filters=QDir::NoFilter, QDir::SortFlags sort=QDir::NoSort) const

  Returns the list of the entries in the directory with zip64 support.
- QStringList entryList (const QStringList &nameFilters, QDir::Filters filters=QDir::NoFilter, QDir::SortFlags sort=QDir::NoSort) const

  Returns the list of the entry names in the directory.
- QStringList entryList (QDir::Filters filters=QDir::NoFilter, QDir::SortFlags sort=QDir::NoSort) const

  Returns the list of the entry names in the directory.
- bool exists (const QString &fileName) const

  Returns `true` if the entry with the specified name exists.
- bool exists () const

  Return `true` if the directory pointed by this QuaZipDir exists.
- QString filePath (const QString &fileName) const

  Returns the full path to the specified file.
- QDir::Filters filter ()

  Returns the default filter.
- bool isRoot () const

  Returns if the QuaZipDir points to the root of the archive.
- QStringList nameFilters () const

  Return the default name filter.
- QString path () const

  Returns the path to the current dir.
- QString relativeFilePath (const QString &fileName) const

  Returns the path to the specified file relative to the current dir.
- void setCaseSensitivity (QuaZip::CaseSensitivity caseSensitivity)

  Sets the default case sensitivity mode.
- void setFilter (QDir::Filters filters)

  Sets the default filter.
- void setNameFilters (const QStringList &nameFilters)

  Sets the default name filter.
- void setPath (const QString &path)

  Goes to the specified path.
- void setSorting (QDir::SortFlags sort)

  Sets the default sorting mode.
- QDir::SortFlags sorting () const

  Returns the default sorting mode.

Private Attributes

- QSharedDataPointer
  < QuaZipDirPrivate > d

12.72.1 Detailed Description

This class is modelled after QDir, and is designed to provide similar features for ZIP archives.

The only significant difference from QDir is that the root path is not '/', but an empty string since that's how the file paths are stored in the archive. However, QuaZipDir understands the paths starting with '/'. It is important in a few places:

- In the cd() function.
- In the constructor.
- In the exists() function.
- In the relativePath() function.

Note that since ZIP uses '/' on all platforms, the '\' separator is not supported.

Definition at line 54 of file quazipdir.h.

12.72.2 Constructor & Destructor Documentation

12.72.2.1 QuaZipDir::QuaZipDir (const QuaZipDir & *that*)

Definition at line 50 of file quazipdir.cpp.

Referenced by exists().

```
50                                          :
51     d(that.d)
52 {
53 }
```

Here is the caller graph for this function:

12.72.2.2 QuaZipDir::QuaZipDir (QuaZip * *zip*, const QString & *dir* = QString())

If *dir* is not specified, points to the root of the archive. The same happens if the *dir* is "/".

Definition at line 55 of file quazipdir.cpp.

References d.

```
55                                                  :
56     d(new QuaZipDirPrivate(zip, dir))
57 {
58     if (d->dir.startsWith('/'))
59         d->dir = d->dir.mid(1);
60 }
```

12.72.2.3 QuaZipDir::~QuaZipDir ( )

Definition at line 62 of file quazipdir.cpp.

```
63 {
64 }
```

12.72.3 Member Function Documentation

12.72.3.1 QuaZip::CaseSensitivity QuaZipDir::caseSensitivity ( ) const

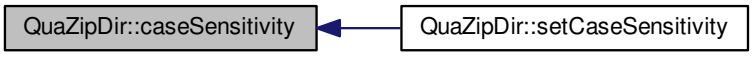
Definition at line 82 of file quazipdir.cpp.

References d.

Referenced by setCaseSensitivity().

```
83 {
84     return d->caseSensitivity;
85 }
```

Here is the caller graph for this function:



12.72.3.2 bool QuaZipDir::cd (const QString & *dirName*)

If the path starts with '/', it is interpreted as an absolute path from the root of the archive. Otherwise, it is interpreted as a path relative to the current directory as was set by the previous cd() or the constructor.

Note that the subsequent path() call will not return a path starting with '/' in all cases.

Definition at line 87 of file quazipdir.cpp.

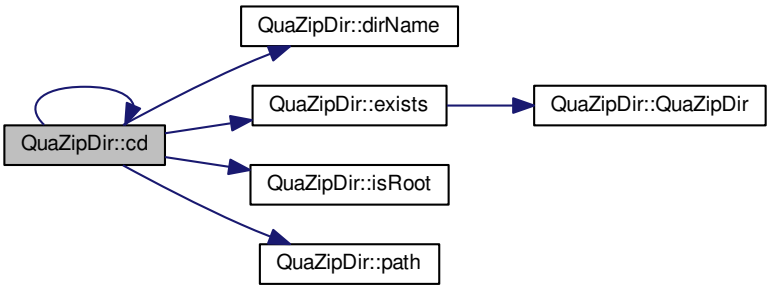
References cd(), d, dirName(), exists(), isRoot(), and path().

Referenced by cd(), and cdUp().

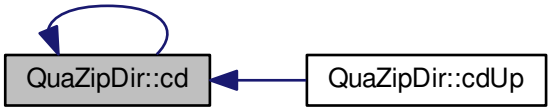
```
88 {
89     if (directoryName == "/") {
90         d->dir = "";
91         return true;
92     }
93     QString dirName = directoryName;
94     if (dirName.endsWith('/'))
95         dirName.chop(1);
96     if (dirName.contains('/')) {
97         QuaZipDir dir(*this);
98         if (dirName.startsWith('/')) {
99 #ifdef QUAZIP_QUAZIPDIR_DEBUG
100             qDebug("QuaZipDir::cd(%s): going to /",
101                    dirName.toUtf8().constData());
102 #endif
103             if (!dir.cd("/"))
104                 return false;
105         }
106         QStringList path = dirName.split('/', QString::SkipEmptyParts);
107         for (QStringList::const_iterator i = path.constBegin();
108                 i != path.end();
```

```
109                 ++i) {
110             const QString &step = *i;
111 #ifdef QUAZIP_QUAZIPDIR_DEBUG
112             qDebug("QuaZipDir::cd(%s): going to %s",
113                    dirName.toUtf8().constData(),
114                    step.toUtf8().constData());
115 #endif
116             if (!dir.cd(step))
117                 return false;
118         }
119         d->dir = dir.path();
120         return true;
121     } else { // no '/'
122         if (dirName == ".") {
123             return true;
124         } else if (dirName == "..") {
125             if (isRoot()) {
126                 return false;
127             } else {
128                 int slashPos = d->dir.lastIndexOf('/');
129                 if (slashPos == -1) {
130                     d->dir = "";
131                 } else {
132                     d->dir = d->dir.left(slashPos);
133                 }
134                 return true;
135             }
136         } else { // a simple subdirectory
137             if (exists(dirName)) {
138                 if (isRoot())
139                     d->dir = dirName;
140                 else
141                     d->dir += "/" + dirName;
142                 return true;
143             } else {
144                 return false;
145             }
146         }
147     }
148 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



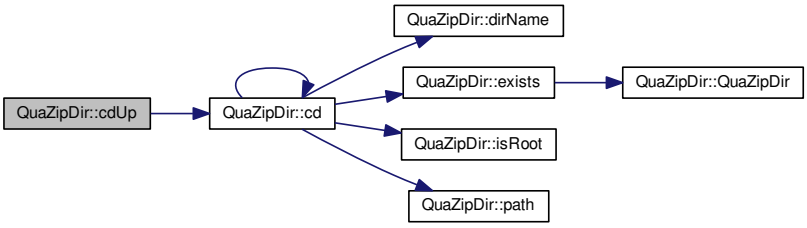
12.72.3.3 bool QuaZipDir::cdUp ( )

Definition at line 150 of file quazipdir.cpp.

References cd().

```
151 {
152     return cd("..");
153 }
```

Here is the call graph for this function:



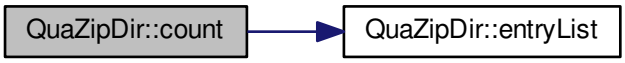
12.72.3.4 uint QuaZipDir::count ( ) const

Definition at line 155 of file quazipdir.cpp.

References entryList().

```
156 {
157     return entryList().count();
158 }
```

Here is the call graph for this function:



12.72.3.5 QString QuaZipDir::dirName ( ) const

The name doesn't include the path.

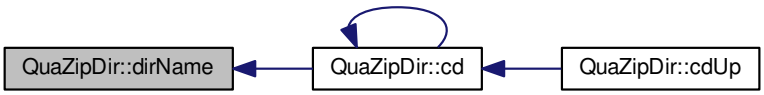
Definition at line 160 of file quazipdir.cpp.

References d.

Referenced by cd().

```
161 {
162     return QDir(d->dir).dirName();
163 }
```

Here is the caller graph for this function:



12.72.3.6 QList< QuaZipFileInfo > QuaZipDir::entryInfoList (const QStringList & *nameFilters*, QDir::Filters *filters* = QDir::NoFilter, QDir::SortFlags *sort* = QDir::NoSort) const

Parameters

nameFilters	The list of file patterns to list, uses the same syntax as QDir.
filters	The entry type filters, only Files and Dirs are accepted.
sort	Sorting mode.

Definition at line 401 of file quazipdir.cpp.

References d.

Referenced by entryInfoList().

```
403 {
404     QList<QuaZipFileInfo> result;
405     if (d->entryInfoList(nameFilters, filters, sort, result))
406         return result;
407     else
408         return QList<QuaZipFileInfo>();
409 }
```

Here is the caller graph for this function:

12.72.3.7 QList< QuaZipFileInfo > QuaZipDir::entryInfoList (QDir::Filters *filters* = QDir::NoFilter, QDir::SortFlags *sort* = QDir::NoSort) const

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

The same as entryInfoList(QStringList(), filters, sort).

Definition at line 411 of file quazipdir.cpp.

References entryInfoList().

```
413 {
414     return entryInfoList(QStringList(), filters, sort);
415 }
```

Here is the call graph for this function:

12.72.3.8 QList< QuaZipFileInfo64 > QuaZipDir::entryInfoList64 (const QStringList & *nameFilters*, QDir::Filters *filters* = QDir::NoFilter, QDir::SortFlags *sort* = QDir::NoSort) const

Parameters

nameFilters	The list of file patterns to list, uses the same syntax as QDir.
filters	The entry type filters, only Files and Dirs are accepted.
sort	Sorting mode.

Definition at line 417 of file quazipdir.cpp.

References d.

Referenced by entryInfoList64().

```
419 {
420     QList<QuaZipFileInfo64> result;
421     if (d->entryInfoList(nameFilters, filters, sort, result))
422         return result;
423     else
424         return QList<QuaZipFileInfo64>();
425 }
```

Here is the caller graph for this function:

12.72.3.9 QList< QuaZipFileInfo64 > QuaZipDir::entryInfoList64 (QDir::Filters *filters* = QDir::NoFilter, QDir::SortFlags *sort* = QDir::NoSort) const

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

The same as entryInfoList64(QStringList(), filters, sort).

Definition at line 427 of file quazipdir.cpp.

References entryInfoList64().

```
429 {
430     return entryInfoList64(QStringList(), filters, sort);
431 }
```

Here is the call graph for this function:

12.72.3.10 QStringList QuaZipDir::entryList (const QStringList & *nameFilters*, QDir::Filters *filters* = QDir::NoFilter, QDir::SortFlags *sort* = QDir::NoSort) const

The same as entryInfoList(nameFilters, filters, sort), but only returns entry names.

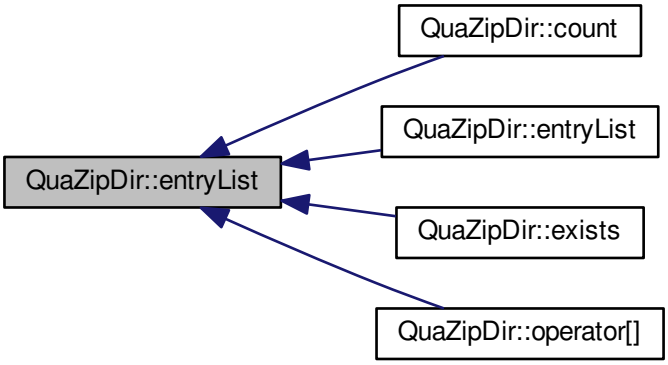
Definition at line 433 of file quazipdir.cpp.

References d.

Referenced by count(), entryList(), exists(), and operator[]().

```
435 {
436     QStringList result;
437     if (d->entryInfoList(nameFilters, filters, sort, result))
438         return result;
439     else
440         return QStringList();
441 }
```

Here is the caller graph for this function:



12.72.3.11 QStringList QuaZipDir::entryList (QDir::Filters *filters* = `QDir::NoFilter`, QDir::SortFlags *sort* = `QDir::NoSort`) const

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

The same as entryList(QStringList(), filters, sort).

Definition at line 443 of file quazipdir.cpp.

References entryList().

```
445 {
446     return entryList(QStringList(), filters, sort);
447 }
```

Here is the call graph for this function:

12.72.3.12 bool QuaZipDir::exists (const QString & *fileName*) const

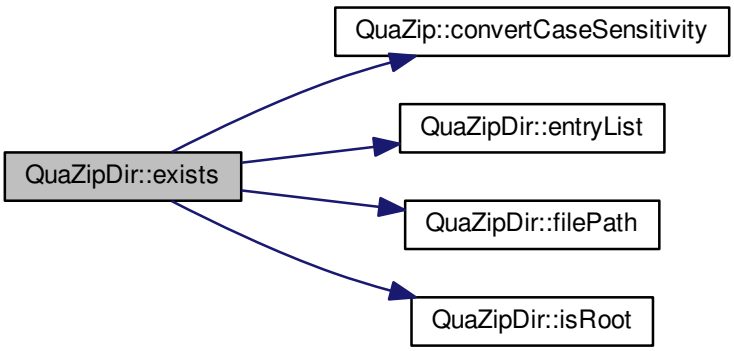
The ".." is considered to exist if the current directory is not root. The "." and "/" are considered to always exist. Paths starting with "/" are relative to the archive root, other paths are relative to the current dir.

Definition at line 449 of file quazipdir.cpp.

References QuaZip::convertCaseSensitivity(), d, entryList(), filePath(), and isRoot().

```
450 {
451     if (filePath == "/" || filePath.isEmpty())
452         return true;
453     QString fileName = filePath;
454     if (fileName.endsWith('/'))
455         fileName.chop(1);
456     if (fileName.contains('/')) {
457         QFileInfo fileInfo(fileName);
458 #ifdef QUAZIP_QUAZIPDIR_DEBUG
459         qDebug("QuaZipDir::exists(): fileName=%s, fileInfo.fileName()=%s, "
460                 "fileInfo.path()=%s", fileName.toUtf8().constData(),
461                 fileInfo.fileName().toUtf8().constData(),
462                 fileInfo.path().toUtf8().constData());
463 #endif
464         QuaZipDir dir(*this);
465         return dir.cd(fileInfo.path()) && dir.exists(fileInfo.fileName());
466     } else {
467         if (fileName == "..") {
468             return !isRoot();
469         } else if (fileName == ".") {
470             return true;
471         } else {
472             QStringList entries = entryList(QDir::AllEntries, QDir::NoSort);
473 #ifdef QUAZIP_QUAZIPDIR_DEBUG
474             qDebug("QuaZipDir::exists(): looking for %s",
475                     fileName.toUtf8().constData());
476             for (QStringList::const_iterator i = entries.constBegin();
477                     i != entries.constEnd();
478                     ++i) {
479                 qDebug("QuaZipDir::exists(): entry: %s",
480                         i->toUtf8().constData());
481             }
482 #endif
483             Qt::CaseSensitivity cs = QuaZip::convertCaseSensitivity(
484                     d->caseSensitivity);
485             if (filePath.endsWith('/')) {
486                 return entries.contains(filePath, cs);
487             } else {
488                 return entries.contains(fileName, cs)
489                     || entries.contains(fileName + "/", cs);
490             }
491         }
492     }
493 }
```

Here is the call graph for this function:



12.72.3.13 bool QuaZipDir::exists ( ) const

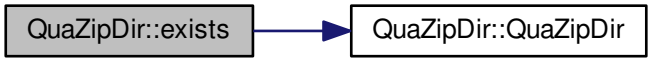
Definition at line 495 of file quazipdir.cpp.

References d, and QuaZipDir().

Referenced by cd().

```
496 {
497     return QuaZipDir(d->zip).exists(d->dir);
498 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.72.3.14 QString QuaZipDir::filePath (const QString & *fileName*) const

Doesn't check if the file actually exists.

Definition at line 500 of file quazipdir.cpp.

References d.

Referenced by exists().

```
501 {
502     return QDir(d->dir).filePath(fileName);
503 }
```

Here is the caller graph for this function:

12.72.3.15 QDir::Filters QuaZipDir::filter ( )

Definition at line 505 of file quazipdir.cpp.

References d.

```
506 {
507     return d->filter;
508 }
```

12.72.3.16 bool QuaZipDir::isRoot ( ) const

Not that the root path is the empty string, not '/'.

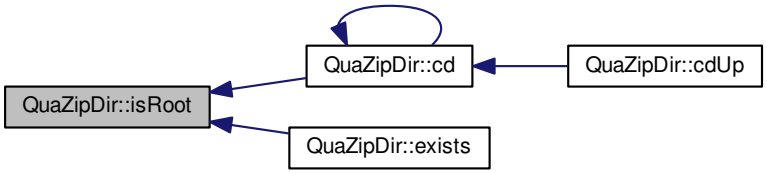
Definition at line 510 of file quazipdir.cpp.

References d.

Referenced by cd(), and exists().

```
511 {
512     return d->simplePath().isEmpty();
513 }
```

Here is the caller graph for this function:



12.72.3.17 QStringList QuaZipDir::nameFilters ( ) const

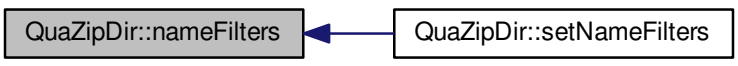
Definition at line 515 of file quazipdir.cpp.

References d.

Referenced by setNameFilters().

```
516 {
517     return d->nameFilters;
518 }
```

Here is the caller graph for this function:



12.72.3.18 bool QuaZipDir::operator!= (const QuaZipDir & *that*) [inline]

Returns

`true` if either this and *that* use different QuaZip instances or if they point to different directories.

Definition at line 75 of file quazipdir.h.

```
75 {return !operator==(that);}
```

12.72.3.19 QuaZipDir & QuaZipDir::operator= (const QuaZipDir & *that*)

Returns

`true` if both this and *that* use the same QuaZip instance and point to the same directory.

Definition at line 71 of file quazipdir.cpp.

References d.

```
72 {
73     this->d = that.d;
74     return *this;
75 }
```

12.72.3.20 bool QuaZipDir::operator== (const QuaZipDir & *that*)

Definition at line 66 of file quazipdir.cpp.

References d.

```
67 {
68     return d->zip == that.d->zip && d->dir == that.d->dir;
69 }
```

12.72.3.21 QString QuaZipDir::operator[] (int *pos*) const

Definition at line 77 of file quazipdir.cpp.

References entryList().

```
78 {
79     return entryList().at(pos);
80 }
```

Here is the call graph for this function:

12.72.3.22 QString QuaZipDir::path ( ) const

The path never starts with '/', and the root path is an empty string.

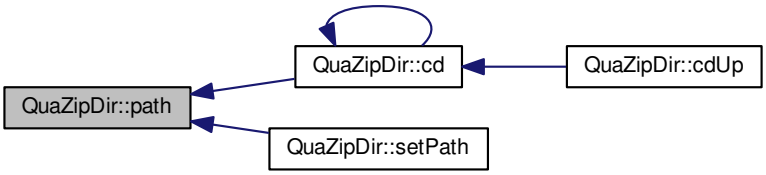
Definition at line 520 of file quazipdir.cpp.

References d.

Referenced by cd(), and setPath().

```
521 {
522     return d->dir;
523 }
```

Here is the caller graph for this function:



12.72.3.23 QString QuaZipDir::relativeFilePath (const QString & *fileName*) const

This function is mostly useless, provided only for the sake of completeness.

Parameters

fileName	The path to the file, should start with "/" if relative to the archive root.

Returns

Path relative to the current dir.

Definition at line 525 of file quazipdir.cpp.

References d.

```
526 {
527     return QDir("/" + d->dir).relativeFilePath(fileName);
528 }
```

12.72.3.24 void QuaZipDir::setCaseSensitivity (QuaZip::CaseSensitivity *caseSensitivity*)

Definition at line 530 of file quazipdir.cpp.

References caseSensitivity(), and d.

```
531 {
532     d->caseSensitivity = caseSensitivity;
533 }
```

Here is the call graph for this function:

12.72.3.25 void QuaZipDir::setFilter (QDir::Filters *filters*)

Definition at line 535 of file quazipdir.cpp.

References d.

```
536 {
537     d->filter = filters;
538 }
```

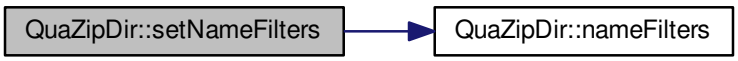
12.72.3.26 void QuaZipDir::setNameFilters (const QStringList & *nameFilters*)

Definition at line 540 of file quazipdir.cpp.

References d, and nameFilters().

```
541 {
542     d->nameFilters = nameFilters;
543 }
```

Here is the call graph for this function:



12.72.3.27 void QuaZipDir::setPath (const QString & *path*)

The difference from cd() is that this function never checks if the path actually exists and doesn't use relative paths, so it's possible to go to the root directory with setPath("").

Note that this function still chops the trailing and/or leading '/' and treats a single '/' as the root path (path() will still return an empty string).

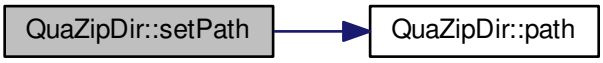
Definition at line 545 of file quazipdir.cpp.

References d, and path().

```
546 {
547     QString newDir = path;
```

```
548     if (newDir == "/") {
549         d->dir = "";
550     } else {
551         if (newDir.endsWith('/'))
552             newDir.chop(1);
553         if (newDir.startsWith('/'))
554             newDir = newDir.mid(1);
555         d->dir = newDir;
556     }
557 }
```

Here is the call graph for this function:



12.72.3.28 void QuaZipDir::setSorting (QDir::SortFlags *sort*)

Definition at line 559 of file quazipdir.cpp.

References d.

```
560 {
561     d->sorting = sort;
562 }
```

12.72.3.29 QDir::SortFlags QuaZipDir::sorting ( ) const

Definition at line 564 of file quazipdir.cpp.

References d.

```
565 {
566     return d->sorting;
567 }
```

12.72.4 Field Documentation

12.72.4.1 QSharedDataPointer<QuaZipDirPrivate> QuaZipDir::d [private]

Definition at line 56 of file quazipdir.h.

Referenced by caseSensitivity(), cd(), dirName(), entryInfoList(), entryInfoList64(), entryList(), exists(), filePath(), filter(), isRoot(), nameFilters(), operator=(), operator==(), path(), QuaZipDir(), relativeFilePath(), setCaseSensitivity(), setFilter(), setNameFilters(), setPath(), setSorting(), and sorting().

The documentation for this class was generated from the following files:

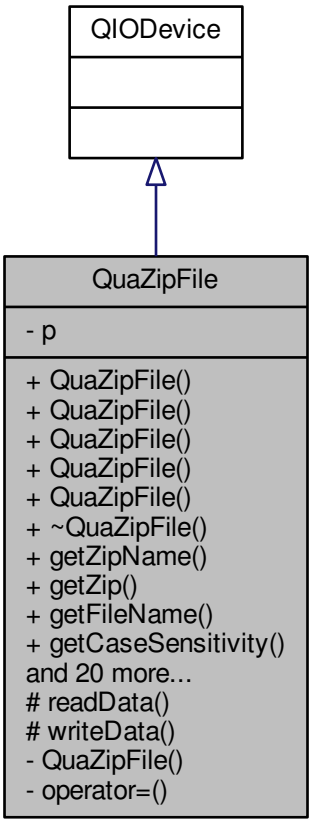
- src/ckb/quazip/quazipdir.h
- src/ckb/quazip/quazipdir.cpp

12.73 QuaZipFile Class Reference

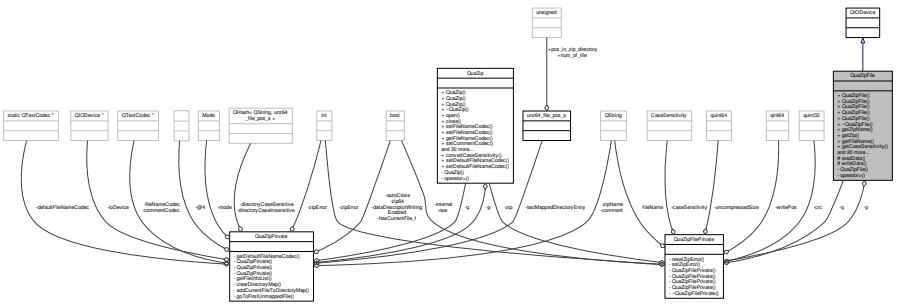
A file inside ZIP archive.

`#include <quazip/quazipfile.h>`

Inheritance diagram for QuaZipFile:



Collaboration diagram for QuaZipFile:



Public Member Functions

- QuaZipFile ()

  Constructs a QuaZipFile instance.
- QuaZipFile (QObject *parent)

  Constructs a QuaZipFile instance.
- QuaZipFile (const QString &zipName, QObject *parent=NULL)

  Constructs a QuaZipFile instance.
- QuaZipFile (const QString &zipName, const QString &fileName, QuaZip::CaseSensitivity cs=QuaZip::csDefault, QObject *parent=NULL)

  Constructs a QuaZipFile instance.
- QuaZipFile (QuaZip *zip, QObject *parent=NULL)

  Constructs a QuaZipFile instance.
- virtual ~QuaZipFile ()

  Destroys a QuaZipFile instance.
- QString getZipName () const

  Returns the ZIP archive file name.
- QuaZip * getZip () const

  Returns a pointer to the associated QuaZip object.
- QString getFileName () const

  Returns file name.
- QuaZip::CaseSensitivity getCaseSensitivity () const

  Returns case sensitivity of the file name.
- QString getActualFileName () const

  Returns the actual file name in the archive.
- void setZipName (const QString &zipName)

  Sets the ZIP archive file name.
- bool isRaw () const

  Returns `true` if the file was opened in raw mode.
- void setZip (QuaZip *zip)

  Binds to the existing QuaZip instance.
- void setFileName (const QString &fileName, QuaZip::CaseSensitivity cs=QuaZip::csDefault)

  Sets the file name.
- virtual bool open (OpenMode mode)

  Opens a file for reading.
- bool open (OpenMode mode, const char *password)

  Opens a file for reading.
- bool open (OpenMode mode, int *method, int *level, bool raw, const char *password=NULL)

  Opens a file for reading.
- bool open (OpenMode mode, const QuaZipNewInfo &info, const char *password=NULL, quint32 crc=0, int method=Z_DEFLATED, int level=Z_DEFAULT_COMPRESSION, bool raw=false, int windowBits=-MAX_WBITS, int memLevel=MAX_MEM_LEVEL, int strategy=Z_DEFAULT_STRATEGY)

  Opens a file for writing.
- virtual bool isSequential () const

  Returns `true`, but beware!
- virtual qint64 pos () const

  Returns current position in the file.
- virtual bool atEnd () const

  Returns `true` if the end of file was reached.
- virtual qint64 size () const

  Returns file size.

- qint64 csize () const

  Returns compressed file size.
- qint64 usize () const

  Returns uncompressed file size.
- bool getFileInfo (QuaZipFileInfo *info)

  Gets information about current file.
- bool getFileInfo (QuaZipFileInfo64 *info)

  Gets information about current file with zip64 support.
- virtual void close ()

  Closes the file.
- int getZipError () const

  Returns the error code returned by the last ZIP/UNZIP API call.
- virtual qint64 bytesAvailable () const

  Returns the number of bytes available for reading.

Protected Member Functions

- qint64 readData (char *data, qint64 maxSize)

  Implementation of the QIODevice::readData().
- qint64 writeData (const char *data, qint64 maxSize)

  Implementation of the QIODevice::writeData().

Private Member Functions

- QuaZipFile (const QuaZipFile &that)
- QuaZipFile & operator= (const QuaZipFile &that)

Private Attributes

- QuaZipFilePrivate * p

Friends

- class QuaZipFilePrivate

12.73.1 Detailed Description

This is the most interesting class. Not only it provides C++ interface to the ZIP/UNZIP package, but also integrates it with Qt by subclassing QIODevice. This makes possible to access files inside ZIP archive using QTextStream or QDataStream, for example. Actually, this is the main purpose of the whole QuaZIP library.

You can either use existing QuaZip instance to create instance of this class or pass ZIP archive file name to this class, in which case it will create internal QuaZip object. See constructors' descriptions for details. Writing is only possible with the existing instance.

Note that due to the underlying library's limitation it is not possible to use multiple QuaZipFile instances to open several files in the same archive at the same time. If you need to write to multiple files in parallel, then you should write to temporary files first, then pack them all at once when you have finished writing. If you need to read multiple files inside the same archive in parallel, you should extract them all into a temporary directory first.

12.73.2 Sequential or random-access?

At the first thought, QuaZipFile has fixed size, the start and the end and should be therefore considered random-access device. But there is one major obstacle to making it random-access: ZIP/UNZIP API does not support seek() operation and the only way to implement it is through reopening the file and re-reading to the required position, but this is prohibitively slow.

Therefore, QuaZipFile is considered to be a sequential device. This has advantage of availability of the ungetChar() operation (QIODevice does not implement it properly for non-sequential devices unless they support seek()). Disadvantage is a somewhat strange behaviour of the size() and pos() functions. This should be kept in mind while using this class.

Definition at line 74 of file quazipfile.h.

12.73.3 Constructor & Destructor Documentation

12.73.3.1 QuaZipFile::QuaZipFile (const QuaZipFile & *that*) [private]

12.73.3.2 QuaZipFile::QuaZipFile ( )

You should use setZipName() and setFileName() or setZip() before trying to call open() on the constructed object.

Definition at line 108 of file quazipfile.cpp.

```
108                       :
109   p(new QuaZipFilePrivate(this))
110 {
111 }
```

12.73.3.3 QuaZipFile::QuaZipFile (QObject * *parent*)

parent argument specifies this object's parent object.

You should use setZipName() and setFileName() or setZip() before trying to call open() on the constructed object.

Definition at line 113 of file quazipfile.cpp.

```
113                                       :
114   QIODevice(parent),
115   p(new QuaZipFilePrivate(this))
116 {
117 }
```

12.73.3.4 QuaZipFile::QuaZipFile (const QString & *zipName*, QObject * *parent* = NULL)

parent argument specifies this object's parent object and *zipName* specifies ZIP archive file name.

You should use setFileName() before trying to call open() on the constructed object.

QuaZipFile constructed by this constructor can be used for read only access. Use QuaZipFile(QuaZip*,QObject*) for writing.

Definition at line 119 of file quazipfile.cpp.

```
119                                                           :
120   QIODevice(parent),
121   p(new QuaZipFilePrivate(this, zipName))
122 {
123 }
```

12.73.3.5 QuaZipFile::QuaZipFile (const QString & *zipName*, const QString & *fileName*, QuaZip::CaseSensitivity *cs* = QuaZip::csDefault, QObject * *parent* = NULL)

parent argument specifies this object's parent object, *zipName* specifies ZIP archive file name and *fileName* and *cs* specify a name of the file to open inside archive.

QuaZipFile constructed by this constructor can be used for read only access. Use QuaZipFile(QuaZip*,QObject*) for writing.

See Also

QuaZip::setCurrentFile()

Definition at line 125 of file quazipfile.cpp.

```
126                                             :
127     QIODevice(parent),
128     p(new QuaZipFilePrivate(this, zipName, fileName, cs))
129 {
130 }
```

12.73.3.6 QuaZipFile::QuaZipFile (QuaZip * *zip*, QObject * *parent* = NULL)

parent argument specifies this object's parent object.

zip is the pointer to the existing QuaZip object. This QuaZipFile object then can be used to read current file in the *zip* or to write to the file inside it.

Warning

Using this constructor for reading current file can be tricky. Let's take the following example:

```
QuaZip zip("archive.zip");
zip.open(QuaZip::mdUnzip);
zip.setCurrentFile("file-in-archive");
QuaZipFile file(&zip);
file.open(QIODevice::ReadOnly);
// ok, now we can read from the file
file.read(somewhere, some);
zip.setCurrentFile("another-file-in-archive"); // oops...
QuaZipFile anotherFile(&zip);
anotherFile.open(QIODevice::ReadOnly);
anotherFile.read(somewhere, some); // this is still ok...
file.read(somewhere, some); // and this is NOT
```

So, what exactly happens here? When we change current file in the `zip` archive, `file` that references it becomes invalid (actually, as far as I understand ZIP/UNZIP sources, it becomes closed, but QuaZipFile has no means to detect it).

Summary: do not close `zip` object or change its current file as long as QuaZipFile is open. Even better - use another constructors which create internal QuaZip instances, one per object, and therefore do not cause unnecessary trouble. This constructor may be useful, though, if you already have a QuaZip instance and do not want to access several files at once. Good example:

```
QuaZip zip("archive.zip");
zip.open(QuaZip::mdUnzip);
// first, we need some information about archive itself
QByteArray comment=zip.getComment();
// and now we are going to access files inside it
QuaZipFile file(&zip);
for(bool more=zip.goToFirstFile(); more; more=zip.goToNextFile()) {
  file.open(QIODevice::ReadOnly);
  // do something cool with file here
  file.close(); // do not forget to close!
}
zip.close();
```

Definition at line 132 of file quazipfile.cpp.

```
132                                                 :
133   QIODevice(parent),
134   p(new QuaZipFilePrivate(this, zip))
135 {
136 }
```

12.73.3.7 QuaZipFile::~QuaZipFile ( ) [virtual]

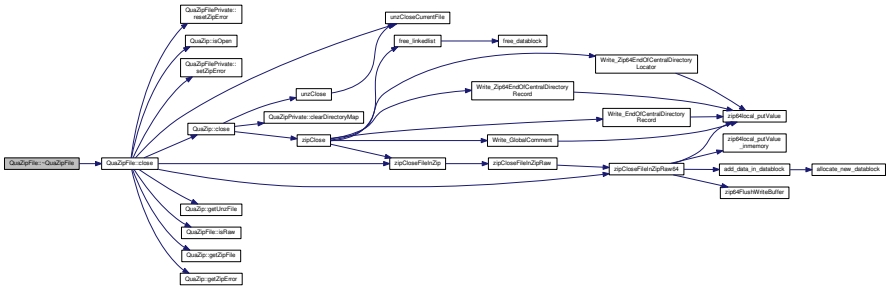
Closes file if open, destructs internal QuaZip object (if it exists and *is* internal, of course).

Definition at line 138 of file quazipfile.cpp.

References close(), and p.

```
139 {
140   if (isOpen())
141     close();
142   delete p;
143 }
```

Here is the call graph for this function:



12.73.4 Member Function Documentation

12.73.4.1 bool QuaZipFile::atEnd ( ) const [virtual]

This function returns `false` in the case of error. This means that you called this function on either not open file, or a file in the not open archive or even on a QuaZipFile instance that does not even have QuaZip instance associated. Do not do that because there is no means to determine whether `false` is returned because of error or because end of file was reached. Well, on the other side you may interpret `false` return value as "there is no file open to check for end of file and there is no end of file therefore".

When writing, this function always returns `true` (because you are always writing to the end of file).

Error code returned by getZipError() is not affected by this function call.

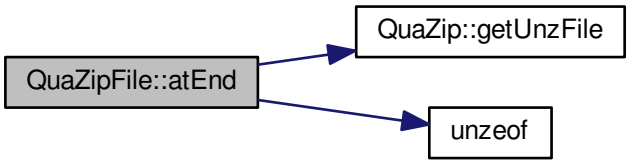
Definition at line 361 of file quazipfile.cpp.

References QuaZip::getUnzFile(), p, unzeof(), and QuaZipFilePrivate::zip.

```
362 {
363   if(p->zip==NULL) {
364     qWarning("QuaZipFile::atEnd(): call setZipName() or setZip() first");
365     return false;
366   }
367   if(!isOpen()) {
368     qWarning("QuaZipFile::atEnd(): file is not open");
369     return false;
370   }
371   if(openMode()&ReadOnly)
372       // the same problem as with pos()
373     return QIODevice::bytesAvailable() == 0
374         && unzeof(p->zip->getUnzFile())==1;
```

```
375   else
376     return true;
377 }
```

Here is the call graph for this function:



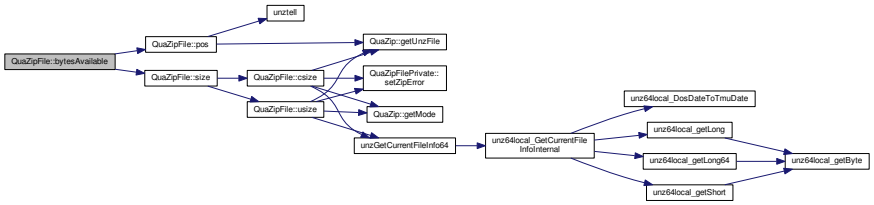
12.73.4.2 qint64 QuaZipFile::bytesAvailable ( ) const [virtual]

Definition at line 499 of file quazipfile.cpp.

References pos(), and size().

```
500 {
501     return size() - pos();
502 }
```

Here is the call graph for this function:



12.73.4.3 void QuaZipFile::close ( ) [virtual]

Call getZipError() to determine if the close was successful.

Definition at line 432 of file quazipfile.cpp.

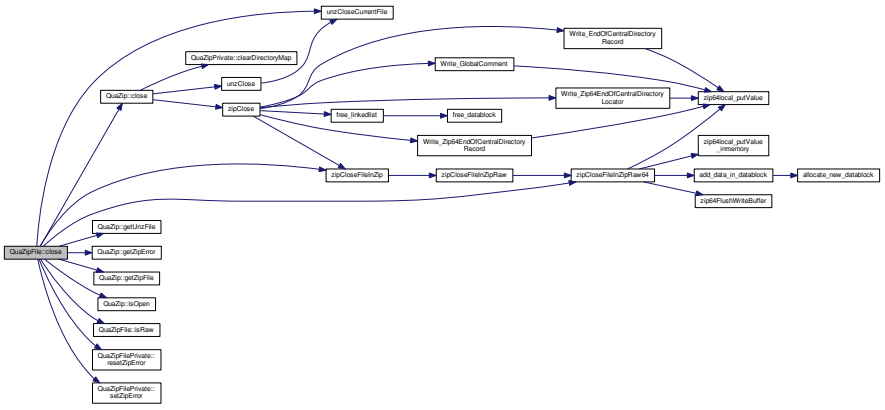
References QuaZip::close(), QuaZipFilePrivate::crc, QuaZip::getUnzFile(), QuaZip::getZipError(), QuaZip::getZipFile(), QuaZipFilePrivate::internal, QuaZip::isOpen(), isRaw(), p, QuaZipFilePrivate::resetZipError(), QuaZipFilePrivate::setZipError(), QuaZipFilePrivate::uncompressedSize, UNZ_OK, unzCloseCurrentFile(), QuaZipFilePrivate::zip, zipCloseFileInZip(), zipCloseFileInZipRaw64(), and QuaZipFilePrivate::zipError.

Referenced by JlCompress::compressFile(), JlCompress::compressSubDir(), JlCompress::extractFile(), and ~QuaZipFile().

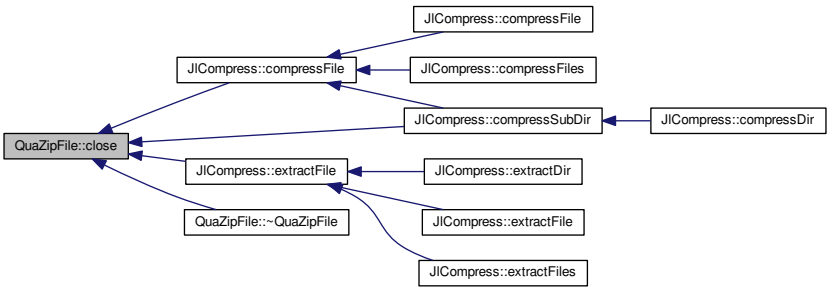
```
433 {
434   p->resetZipError();
435   if(p->zip==NULL||!p->zip->isOpen()) return;
```

```
436   if(!isOpen()) {
437     qWarning("QuaZipFile::close(): file isn't open");
438     return;
439   }
440   if(openMode()&ReadOnly)
441     p->setZipError(unzCloseCurrentFile(p->zip->
      getUnzFile()));
442   else if(openMode()&WriteOnly)
443     if(isRaw()) p->setZipError(zipCloseFileInZipRaw64(
      p->zip->getZipFile(), p->uncompressedSize, p->
      crc));
444     else p->setZipError(zipCloseFileInZip(p->zip->
      getZipFile()));
445   else {
446     qWarning("Wrong open mode: %d", (int)openMode());
447     return;
448   }
449   if(p->zipError==UNZ_OK) setOpenMode(QIODevice::NotOpen);
450   else return;
451   if(p->internal) {
452     p->zip->close();
453     p->setZipError(p->zip->getZipError());
454   }
455 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.73.4.4 qint64 QuaZipFile::csize ( ) const

Equivalent to calling getFileInfo() and then getting compressedSize field, but more convenient and faster.

File must be open for reading before calling this function.

Returns -1 on error, call getZipError() to get error code.

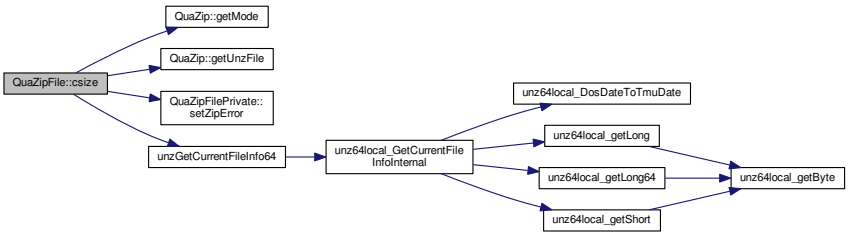
Definition at line 391 of file quazipfile.cpp.

References unz_file_info64_s::compressed_size, QuaZip::getMode(), QuaZip::getUnzFile(), QuaZip::mdUnzip, p, QuaZipFilePrivate::setZipError(), UNZ_OK, unzGetCurrentFileInfo64(), QuaZipFilePrivate::zip, and QuaZipFilePrivate::zipError.

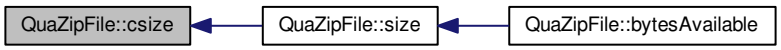
Referenced by size().

```
392 {
393   unz_file_info64 info_z;
394   p->setZipError(UNZ_OK);
395   if(p->zip==NULL||p->zip->getMode()!=QuaZip::mdUnzip) return -1;
396   p->setZipError(unzGetCurrentFileInfo64(p->
      zip->getUnzFile(), &info_z, NULL, 0, NULL, 0, NULL, 0));
397   if(p->zipError!=UNZ_OK)
398     return -1;
399   return info_z.compressed_size;
400 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.73.4.5 QString QuaZipFile::getActualFileName ( ) const

This is *not* a ZIP archive file name, but a name of file inside archive. It is not necessary the same name that you have passed to the QuaZipFile(const QString&,const QString&,QuaZip::CaseSensitivity,QObject*), setFileName() or QuaZip::setCurrentFile() - this is the real file name inside archive, so it may differ in case if the file name search was case-insensitive.

Equivalent to calling getCurrentFileName() on the associated QuaZip object. Returns null string if there is no associated QuaZip object or if it does not have a current file yet. And this is the case if you called setFileName() but did not open the file yet. So this is perfectly fine:

```
QuaZipFile file("somezip.zip");
file.setFileName("somefile");
QString name=file.getName(); // name=="somefile"
QString actual=file.getActualFileName(); // actual is null string
file.open(QIODevice::ReadOnly);
QString actual=file.getActualFileName(); // actual can be "SoMeFiLe" on Windows
```

See Also

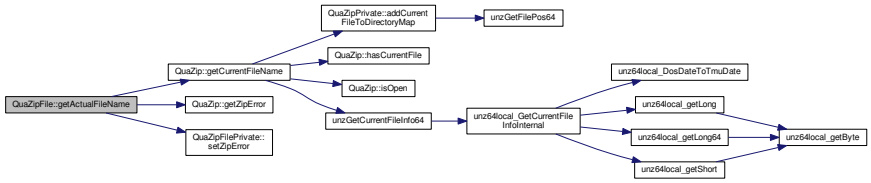
getZipName(), getFileName(), QuaZip::CaseSensitivity

Definition at line 155 of file quazipfile.cpp.

References QuaZip::getCurrentFileName(), QuaZip::getZipError(), p, QuaZipFilePrivate::setZipError(), UNZ_OK, and QuaZipFilePrivate::zip.

```
156 {
157   p->setZipError(UNZ_OK);
158   if (p->zip == NULL || (openMode() & WriteOnly))
159     return QString();
160   QString name=p->zip->getCurrentFileName();
161   if(name.isNull())
162     p->setZipError(p->zip->getZipError());
163   return name;
164 }
```

Here is the call graph for this function:



12.73.4.6 QuaZip::CaseSensitivity QuaZipFile::getCaseSensitivity ( ) const

This function returns case sensitivity argument you passed to this object either by using QuaZipFile(const QString&,const QString&,QuaZip::CaseSensitivity,QObject*) or by calling setFileName().

Returns unpredictable value if getFileName() returns null string (this is the case when you did not used setFileName() or constructor above).

See Also

getFileName

Definition at line 484 of file quazipfile.cpp.

References QuaZipFilePrivate::caseSensitivity, and p.

```
485 {
486   return p->caseSensitivity;
487 }
```

12.73.4.7 bool QuaZipFile::getFileInfo (QuaZipFileInfo * *info*)

This function does the same thing as calling QuaZip::getCurrentFileInfo() on the associated QuaZip object, but you can not call getCurrentFileInfo() if the associated QuaZip is internal (because you do not have access to it), while you still can call this function in that case.

File must be open for reading before calling this function.

Returns

`false` in the case of an error.

This function doesn't support zip64, but will still work fine on zip64 archives if file sizes are below 4 GB, otherwise the values will be set as if converted using QuaZipFileInfo64::toQuaZipFileInfo().

See Also

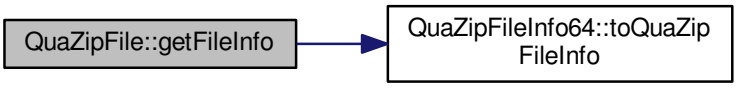
getFileInfo(QuaZipFileInfo64*)

Definition at line 413 of file quazipfile.cpp.

References QuaZipFileInfo64::toQuaZipFileInfo().

```
414 {
415     QuaZipFileInfo64 info64;
416     if (getFileInfo(&info64)) {
417         info64.toQuaZipFileInfo(*info);
418         return true;
419     } else {
420         return false;
421     }
422 }
```

Here is the call graph for this function:



12.73.4.8 bool QuaZipFile::getFileInfo (QuaZipFileInfo64 * *info*)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

See Also

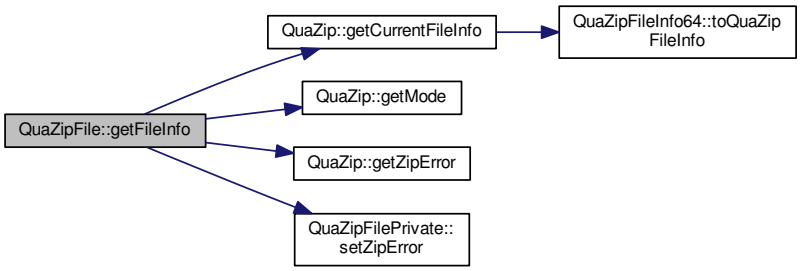
getFileInfo(QuaZipFileInfo*)

Definition at line 424 of file quazipfile.cpp.

References QuaZip::getCurrentFileInfo(), QuaZip::getMode(), QuaZip::getZipError(), QuaZip::mdUnzip, p, QuaZipFilePrivate::setZipError(), UNZ_OK, QuaZipFilePrivate::zip, and QuaZipFilePrivate::zipError.

```
425 {
426     if(p->zip==NULL||p->zip->getMode()!=QuaZip::mdUnzip) return false;
427     p->zip->getCurrentFileInfo(info);
428     p->setZipError(p->zip->getZipError());
429     return p->zipError==UNZ_OK;
430 }
```

Here is the call graph for this function:



12.73.4.9 QString QuaZipFile::getFileName ( ) const

This function returns file name you passed to this object either by using QuaZipFile(const QString&,const QString&,-QuaZip::CaseSensitivity,QObject*) or by calling setFileName(). Real name of the file may differ in case if you used case-insensitivity.

Returns null string if there is no file name set yet. This is the case when this QuaZipFile operates on the existing QuaZip object (constructor QuaZipFile(QuaZip*,QObject*) or setZip() was used).

See Also

getActualFileName

Definition at line 479 of file quazipfile.cpp.

References QuaZipFilePrivate::fileName, and p.

```
480 {
481   return p->fileName;
482 }
```

12.73.4.10 QuaZip * QuaZipFile::getZip ( ) const

Returns NULL if there is no associated QuaZip or it is internal (so you will not mess with it).

Definition at line 150 of file quazipfile.cpp.

References QuaZipFilePrivate::internal, p, and QuaZipFilePrivate::zip.

```
151 {
152     return p->internal ? NULL : p->zip;
153 }
```

12.73.4.11 int QuaZipFile::getZipError ( ) const

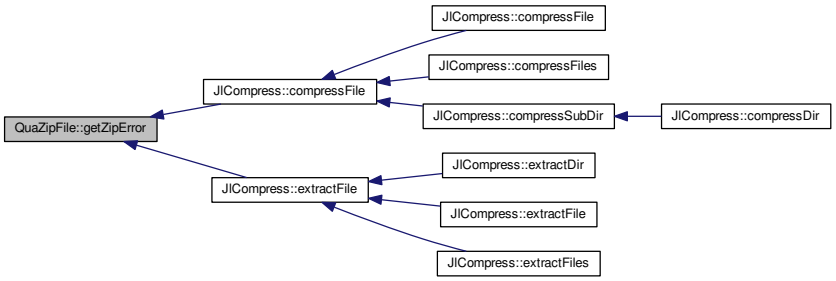
Definition at line 494 of file quazipfile.cpp.

References p, and QuaZipFilePrivate::zipError.

Referenced by JlCompress::compressFile(), and JlCompress::extractFile().

```
495 {
496   return p->zipError;
497 }
```

Here is the caller graph for this function:



12.73.4.12 QString QuaZipFile::getZipName ( ) const

If this object was created by passing QuaZip pointer to the constructor, this function will return that QuaZip's file name (or null string if that object does not have file name yet).

Otherwise, returns associated ZIP archive file name or null string if there are no name set yet.

See Also

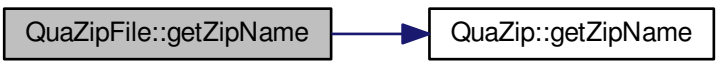
setZipName() getFileName()

Definition at line 145 of file quazipfile.cpp.

References QuaZip::getZipName(), p, and QuaZipFilePrivate::zip.

```
146 {
147   return p->zip==NULL ? QString() : p->zip->getZipName();
148 }
```

Here is the call graph for this function:



12.73.4.13 bool QuaZipFile::isRaw ( ) const

If the file is not open, the returned value is undefined.

See Also

open(OpenMode,int*,int*,bool,const char*)

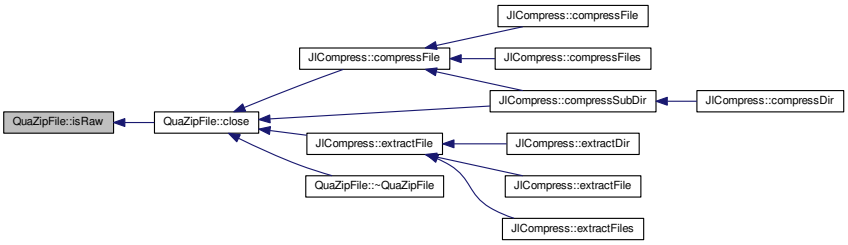
Definition at line 489 of file quazipfile.cpp.

References p, and QuaZipFilePrivate::raw.

Referenced by close().

```
490 {
491   return p->raw;
492 }
```

Here is the caller graph for this function:



12.73.4.14 bool QuaZipFile::isSequential ( ) const [virtual]

Definition at line 337 of file quazipfile.cpp.

```
338 {
339   return true;
340 }
```

12.73.4.15 bool QuaZipFile::open (OpenMode *mode*) [virtual]

Returns `true` on success, `false` otherwise. Call getZipError() to get error code.

Note

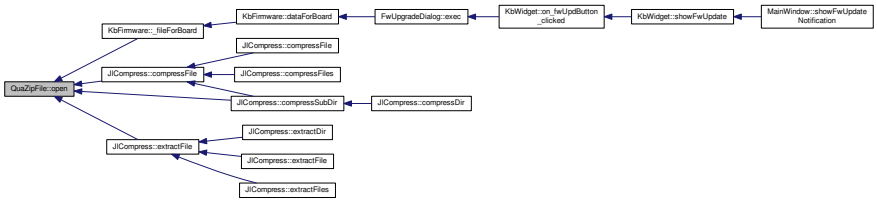
Since ZIP/UNZIP API provides buffered reading only, QuaZipFile does not support unbuffered reading. So do not pass QIODevice::Unbuffered flag in *mode*, or open will fail.

Definition at line 221 of file quazipfile.cpp.

Referenced by KbFirmware::_fileForBoard(), JICompress::compressFile(), JICompress::compressSubDir(), JICompress::extractFile().

```
222 {
223   return open(mode, NULL);
224 }
```

Here is the caller graph for this function:



12.73.4.16 bool QuaZipFile::open (OpenMode *mode*, const char * *password*) [inline]

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Argument *password* specifies a password to decrypt the file. If it is NULL then this function behaves just like open(OpenMode).

Definition at line 298 of file quazipfile.h.

```
299     {return open(mode, NULL, NULL, false, password);}
```

12.73.4.17 bool QuaZipFile::open (OpenMode *mode*, int * *method*, int * *level*, bool *raw*, const char * *password* = NULL)

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts. Argument *password* specifies a password to decrypt the file.

An integers pointed by *method* and *level* will receive codes of the compression method and level used. See unzip.h.

If raw is `true` then no decompression is performed.

method should not be `NULL`. *level* can be `NULL` if you don't want to know the compression level.

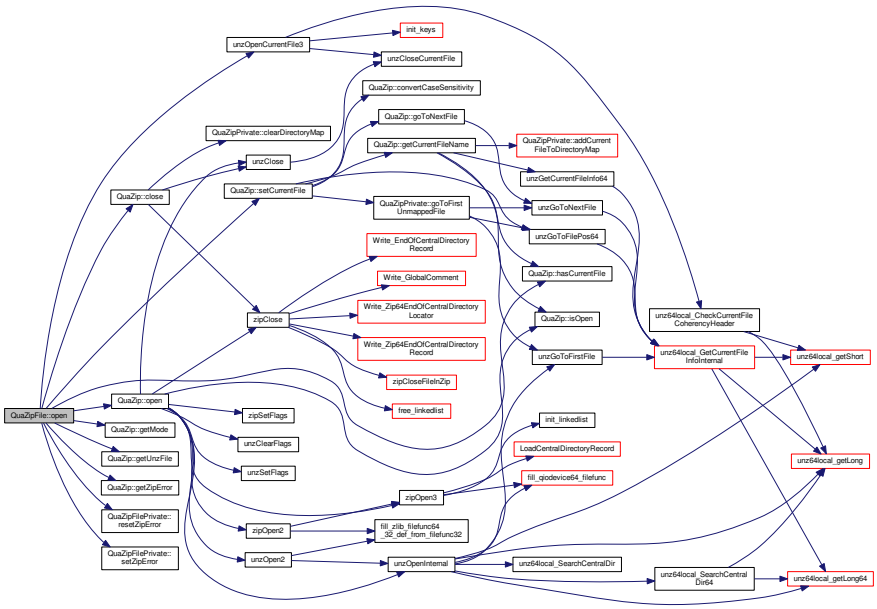
Definition at line 226 of file quazipfile.cpp.

References QuaZipFilePrivate::caseSensitivity, QuaZip::close(), QuaZipFilePrivate::fileName, QuaZip::getMode(), QuaZip::getUnzFile(), QuaZip::getZipError(), QuaZip::hasCurrentFile(), QuaZipFilePrivate::internal, QuaZip::mdUnzip, QuaZip::open(), p, QuaZipFilePrivate::raw, QuaZipFilePrivate::resetZipError(), QuaZip::setCurrentFile(), QuaZipFilePrivate::setZipError(), UNZ_OK, unzOpenCurrentFile3(), QuaZipFilePrivate::zip, and QuaZipFilePrivate::zipError.

```
227 {
228   p->resetZipError();
229   if(isOpen()) {
230     qWarning("QuaZipFile::open(): already opened");
231     return false;
232   }
233   if(mode&Unbuffered) {
234     qWarning("QuaZipFile::open(): Unbuffered mode is not supported");
235     return false;
236   }
237   if((mode&ReadOnly)&&!(mode&WriteOnly)) {
238     if(p->internal) {
239       if(!p->zip->open(QuaZip::mdUnzip)) {
240         p->setZipError(p->zip->getZipError());
241         return false;
242       }
243       if(!p->zip->setCurrentFile(p->fileName, p->
      caseSensitivity)) {
244         p->setZipError(p->zip->getZipError());
245         p->zip->close();
246         return false;
247       }
248     } else {
249       if(p->zip==NULL) {
250         qWarning("QuaZipFile::open(): zip is NULL");
251         return false;
```

```
252       }
253       if(p->zip->getMode()!=QuaZip::mdUnzip) {
254         qWarning("QuaZipFile::open(): file open mode %d incompatible with ZIP open mode %d",
255             (int)mode, (int)p->zip->getMode());
256         return false;
257       }
258       if(!p->zip->hasCurrentFile()) {
259         qWarning("QuaZipFile::open(): zip does not have current file");
260         return false;
261       }
262     }
263     p->setZipError(unzOpenCurrentFile3(p->zip->
    getUnzFile(), method, level, (int)raw, password));
264     if(p->zipError==UNZ_OK) {
265       setOpenMode(mode);
266       p->raw=raw;
267       return true;
268     } else
269       return false;
270   }
271   qWarning("QuaZipFile::open(): open mode %d not supported by this function", (int)mode);
272   return false;
273 }
```

Here is the call graph for this function:



12.73.4.18 bool QuaZipFile::open (OpenMode *mode*, const QuaZipNewInfo & *info*, const char * *password* = NULL, quint32 *crc* = 0, int *method* = Z_DEFLATED, int *level* = Z_DEFAULT_COMPRESSION, bool *raw* = false, int *windowBits* = -MAX_WBITS, int *memLevel* = MAX_MEM_LEVEL, int *strategy* = Z_DEFAULT_STRATEGY)

info argument specifies information about file. It should at least specify a correct file name. Also, it is a good idea to specify correct timestamp (by default, current time will be used). See QuaZipNewInfo.

The *password* argument specifies the password for crypting. Pass NULL if you don't need any crypting. The *crc* argument was supposed to be used for crypting too, but then it turned out that it's false information, so you need to set it to 0 unless you want to use the raw mode (see below).

Arguments *method* and *level* specify compression method and level. The only method supported is Z_DEFLATED, but you may also specify 0 for no compression. If all of the files in the archive use both method 0 and either level 0 is explicitly specified or data descriptor writing is disabled with QuaZip::setDataDescriptorWritingEnabled(), then the resulting archive is supposed to be compatible with the 1.0 ZIP format version, should you need that. Except for this, *level* has no other effects with method 0.

If *raw* is `true`, no compression is performed. In this case, *crc* and uncompressedSize field of the *info* are required.

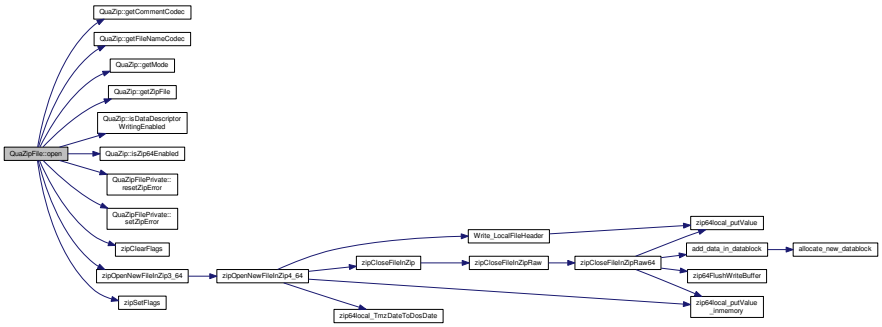
Arguments *windowBits*, *memLevel*, *strategy* provide zlib algorithms tuning. See deflateInit2() in zlib.

Definition at line 275 of file quazipfile.cpp.

References QuaZipNewInfo::comment, QuaZipFilePrivate::crc, QuaZipNewInfo::dateTime, zip_fileinfo::dosDate, zip_fileinfo::external_fa, QuaZipNewInfo::externalAttr, QuaZipNewInfo::extraGlobal, QuaZipNewInfo::extraLocal, QuaZip::getCommentCodec(), QuaZip::getFileNameCodec(), QuaZip::getMode(), QuaZip::getZipFile(), QuaZipFilePrivate::internal, zip_fileinfo::internal_fa, QuaZipNewInfo::internalAttr, QuaZip::isDataDescriptorWritingEnabled(), QuaZip::isZip64Enabled(), QuaZip::mdAdd, QuaZip::mdAppend, QuaZip::mdCreate, QuaZipNewInfo::name, p, QuaZipFilePrivate::raw, QuaZipFilePrivate::resetZipError(), QuaZipFilePrivate::setZipError(), tm_zip_s::tm_hour, tm_zip_s::tm_mday, tm_zip_s::tm_min, tm_zip_s::tm_mon, tm_zip_s::tm_sec, tm_zip_s::tm_year, zip_fileinfo::tmz_date, QuaZipFilePrivate::uncompressedSize, QuaZipNewInfo::uncompressedSize, UNZ_OK, QuaZipFilePrivate::writePos, QuaZipFilePrivate::zip, ZIP_WRITE_DATA_DESCRIPTOR, zipClearFlags(), QuaZipFilePrivate::zipError, zipOpenNewFileInZip3_64(), and zipSetFlags().

```
279 {
280   zip_fileinfo info_z;
281   p->resetZipError();
282   if(isOpen()) {
283     qWarning("QuaZipFile::open(): already opened");
284     return false;
285   }
286   if((mode&WriteOnly)&&!(mode&ReadOnly)) {
287     if(p->internal) {
288       qWarning("QuaZipFile::open(): write mode is incompatible with internal QuaZip approach");
289       return false;
290     }
291     if(p->zip==NULL) {
292       qWarning("QuaZipFile::open(): zip is NULL");
293       return false;
294     }
295     if(p->zip->getMode()!=QuaZip::mdCreate&&p->zip->
    getMode()!=QuaZip::mdAppend&&p->zip->getMode()!=
    QuaZip::mdAdd) {
296       qWarning("QuaZipFile::open(): file open mode %d incompatible with ZIP open mode %d",
297           (int)mode, (int)p->zip->getMode());
298       return false;
299     }
300     info_z.tmz_date.tm_year=info.dateTime.date().year();
301     info_z.tmz_date.tm_mon=info.dateTime.date().month() - 1;
302     info_z.tmz_date.tm_mday=info.dateTime.date().day();
303     info_z.tmz_date.tm_hour=info.dateTime.time().hour();
304     info_z.tmz_date.tm_min=info.dateTime.time().minute();
305     info_z.tmz_date.tm_sec=info.dateTime.time().second();
306     info_z.dosDate = 0;
307     info_z.internal_fa=(uLong)info.internalAttr;
308     info_z.external_fa=(uLong)info.externalAttr;
309     if (p->zip->isDataDescriptorWritingEnabled())
310         zipSetFlags(p->zip->getZipFile(),
    ZIP_WRITE_DATA_DESCRIPTOR);
311     else
312         zipClearFlags(p->zip->getZipFile(),
    ZIP_WRITE_DATA_DESCRIPTOR);
313     p->setZipError(zipOpenNewFileInZip3_64(p->
    zip->getZipFile(),
314           p->zip->getFileNameCodec()->fromUnicode(info.name).constData(), &info_z,
315           info.extraLocal.constData(), info.extraLocal.length(),
316           info.extraGlobal.constData(), info.extraGlobal.length(),
317           p->zip->getCommentCodec()->fromUnicode(info.comment).constData(),
318           method, level, (int)raw,
319           windowBits, memLevel, strategy,
320           password, (uLong)crc, p->zip->isZip64Enabled()));
321     if(p->zipError==UNZ_OK) {
322       p->writePos=0;
323       setOpenMode(mode);
324       p->raw=raw;
325       if(raw) {
326         p->crc=crc;
327         p->uncompressedSize=info.uncompressedSize;
328       }
329       return true;
330     } else
331       return false;
332   }
333   qWarning("QuaZipFile::open(): open mode %d not supported by this function", (int)mode);
334   return false;
335 }
```

Here is the call graph for this function:



12.73.4.19 QuaZipFile& QuaZipFile::operator= (const QuaZipFile & *that*) [private]

12.73.4.20 qint64 QuaZipFile::pos ( ) const [virtual]

Implementation of the QIODevice::pos(). When reading, this function is a wrapper to the ZIP/UNZIP unztell(), therefore it is unable to keep track of the ungetChar() calls (which is non-virtual and therefore is dangerous to reimplement). So if you are using ungetChar() feature of the QIODevice, this function reports incorrect value until you get back characters which you ungot.

When writing, pos() returns number of bytes already written (uncompressed unless you use raw mode).

Note

Although QuaZipFile is a sequential device and therefore pos() should always return zero, it does not, because it would be misguiding. Keep this in mind.

This function returns -1 if the file or archive is not open.

Error code returned by getZipError() is not affected by this function call.

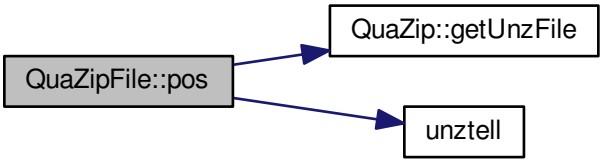
Definition at line 342 of file quazipfile.cpp.

References QuaZip::getUnzFile(), p, unztell(), QuaZipFilePrivate::writePos, and QuaZipFilePrivate::zip.

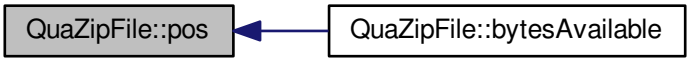
Referenced by bytesAvailable().

```
343 {
344   if(p->zip==NULL) {
345     qWarning("QuaZipFile::pos(): call setZipName() or setZip() first");
346     return -1;
347   }
348   if(!isOpen()) {
349     qWarning("QuaZipFile::pos(): file is not open");
350     return -1;
351   }
352   if(openMode()&ReadOnly)
353       // QIODevice::pos() is broken for sequential devices,
354       // but thankfully bytesAvailable() returns the number of
355       // bytes buffered, so we know how far ahead we are.
356     return unztell(p->zip->getUnzFile()) - QIODevice::bytesAvailable();
357   else
358     return p->writePos;
359 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



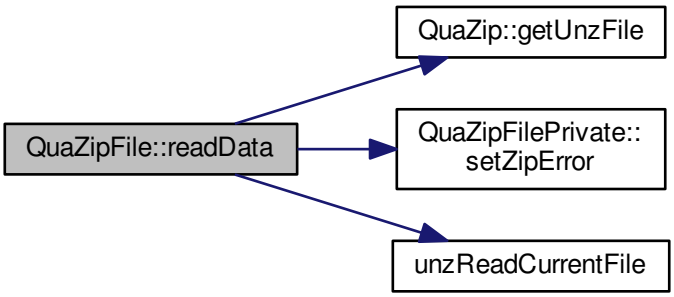
12.73.4.21 qint64 QuaZipFile::readData (char * *data*, qint64 *maxSize*) [protected]

Definition at line 457 of file quazipfile.cpp.

References QuaZip::getUnzFile(), p, QuaZipFilePrivate::setZipError(), UNZ_OK, unzReadCurrentFile(), and QuaZipFilePrivate::zip.

```
458 {
459   p->setZipError(UNZ_OK);
460   qint64 bytesRead=unzReadCurrentFile(p->zip->getUnzFile(), data, (unsigned
      )maxSize);
461   if (bytesRead < 0) {
462     p->setZipError((int) bytesRead);
463     return -1;
464   }
465   return bytesRead;
466 }
```

Here is the call graph for this function:



12.73.4.22 void QuaZipFile::setFileName (const QString & *fileName,* QuaZip::CaseSensitivity *cs* = QuaZip::csDefault)

Will do nothing if at least one of the following conditions is met:

- ZIP name has not been set yet (getZipName() returns null string).
- This QuaZipFile is associated with external QuaZip. In this case you should call that QuaZip's setCurrentFile() function instead!
- File is already open so setting the name is meaningless.

See Also

QuaZip::setCurrentFile

Definition at line 191 of file quazipfile.cpp.

References QuaZipFilePrivate::caseSensitivity, QuaZipFilePrivate::fileName, QuaZipFilePrivate::internal, p, and QuaZipFilePrivate::zip.

```
192 {
193   if(p->zip==NULL) {
194     qWarning("QuaZipFile::setFileName(): call setZipName() first");
195     return;
196   }
197   if(!p->internal) {
198     qWarning("QuaZipFile::setFileName(): should not be used when not using internal QuaZip");
199     return;
200   }
201   if(isOpen()) {
202     qWarning("QuaZipFile::setFileName(): can not set file name for already opened file");
203     return;
204   }
205   p->fileName=fileName;
206   if (p->fileName.startsWith('/'))
207       p->fileName = p->fileName.mid(1);
208   p->caseSensitivity=cs;
209 }
```

12.73.4.23 void QuaZipFile::setZip (QuaZip * *zip*)

This function destroys internal QuaZip object, if any, and makes this QuaZipFile to use current file in the *zip* object for any further operations. See QuaZipFile(QuaZip*,QObject*) for the possible pitfalls.

Will do nothing if the file is currently open. You must close() it first.

Definition at line 178 of file quazipfile.cpp.

References QuaZipFilePrivate::fileName, QuaZipFilePrivate::internal, p, and QuaZipFilePrivate::zip.

```
179 {
180   if(isOpen()) {
181     qWarning("QuaZipFile::setZip(): file is already open - can not set ZIP");
182     return;
183   }
184   if(p->zip!=NULL && p->internal)
185     delete p->zip;
186   p->zip=zip;
187   p->fileName=QString();
188   p->internal=false;
189 }
```

12.73.4.24 void QuaZipFile::setZipName (const QString & *zipName*)

Automatically creates internal QuaZip object and destroys previously created internal QuaZip object, if any.

Will do nothing if this file is already open. You must close() it first.

Definition at line 166 of file quazipfile.cpp.

References QuaZipFilePrivate::internal, p, and QuaZipFilePrivate::zip.

```
167 {
168   if(isOpen()) {
169     qWarning("QuaZipFile::setZipName(): file is already open - can not set ZIP name");
170     return;
171   }
172   if(p->zip!=NULL && p->internal)
173     delete p->zip;
174   p->zip=new QuaZip(zipName);
175   p->internal=true;
176 }
```

12.73.4.25 qint64 QuaZipFile::size ( ) const [virtual]

This function returns csize() if the file is open for reading in raw mode, usize() if it is open for reading in normal mode and pos() if it is open for writing.

Returns -1 on error, call getZipError() to get error code.

Note

> This function returns file size despite that QuaZipFile is considered to be sequential device, for which size() should return bytesAvailable() instead. But its name would be very misguiding otherwise, so just keep in mind this inconsistence.

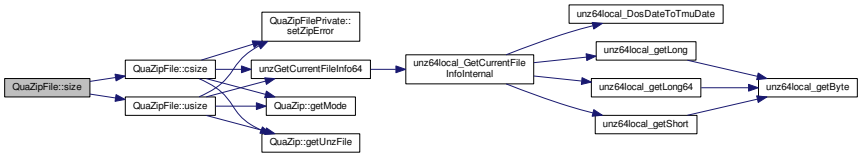
Definition at line 379 of file quazipfile.cpp.

References csize(), p, QuaZipFilePrivate::raw, usize(), and QuaZipFilePrivate::writePos.

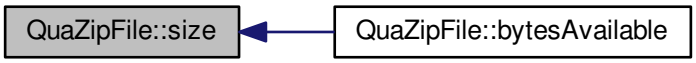
Referenced by bytesAvailable().

```
380 {
381   if(!isOpen()) {
382     qWarning("QuaZipFile::atEnd(): file is not open");
383     return -1;
384   }
385   if(openMode()&ReadOnly)
386     return p->raw?csize():usize();
387   else
388     return p->writePos;
389 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.73.4.26 qint64 QuaZipFile::usize ( ) const

Equivalent to calling getFileInfo() and then getting uncompressedSize field, but more convenient and faster. See getFileInfo() for a warning.

File must be open for reading before calling this function.

Returns -1 on error, call getZipError() to get error code.

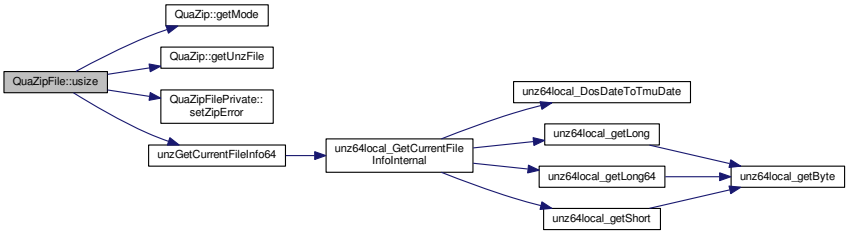
Definition at line 402 of file quazipfile.cpp.

References QuaZip::getMode(), QuaZip::getUnzFile(), QuaZip::mdUnzip, p, QuaZipFilePrivate::setZipError(), unz_file_info64_s::uncompressed_size, UNZ_OK, unzGetCurrentFileInfo64(), QuaZipFilePrivate::zip, and QuaZipFilePrivate::zipError.

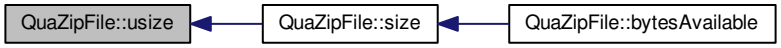
Referenced by size().

```
403 {
404   unz_file_info64 info_z;
405   p->setZipError(UNZ_OK);
406   if(p->zip==NULL||p->zip->getMode()!=QuaZip::mdUnzip) return -1;
407   p->setZipError(unzGetCurrentFileInfo64(p->
      zip->getUnzFile(), &info_z, NULL, 0, NULL, 0, NULL, 0));
408   if(p->zipError!=UNZ_OK)
409     return -1;
410   return info_z.uncompressed_size;
411 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



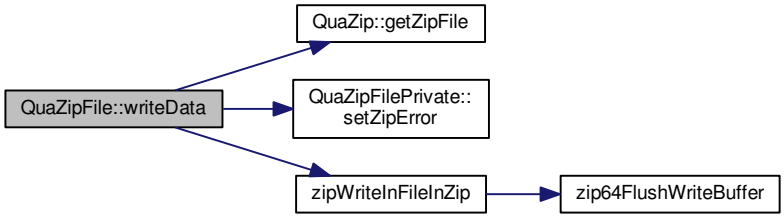
12.73.4.27 qint64 QuaZipFile::writeData (const char * *data*, qint64 *maxSize*) [protected]

Definition at line 468 of file quazipfile.cpp.

References QuaZip::getZipFile(), p, QuaZipFilePrivate::setZipError(), QuaZipFilePrivate::writePos, QuaZipFilePrivate::zip, ZIP_OK, QuaZipFilePrivate::zipError, and zipWriteInFileInZip().

```
469 {
470   p->setZipError(ZIP_OK);
471   p->setZipError(zipWriteInFileInZip(p->zip->
      getZipFile(), data, (uint)maxSize));
472   if(p->zipError!=ZIP_OK) return -1;
473   else {
474     p->writePos+=maxSize;
475     return maxSize;
476   }
477 }
```

Here is the call graph for this function:



12.73.5 Friends And Related Function Documentation

12.73.5.1 friend class QuaZipFilePrivate [friend]

Definition at line 75 of file quazipfile.h.

12.73.6 Field Documentation

12.73.6.1 QuaZipFilePrivate* QuaZipFile::p [private]

Definition at line 78 of file quazipfile.h.

Referenced by atEnd(), close(), csize(), getActualFileName(), getCaseSensitivity(), getFileInfo(), getFileName(), getZip(), getZipError(), getZipName(), isRaw(), open(), pos(), readData(), setFileName(), setZip(), setZipName(), size(), usize(), writeData(), and ~QuaZipFile().

The documentation for this class was generated from the following files:

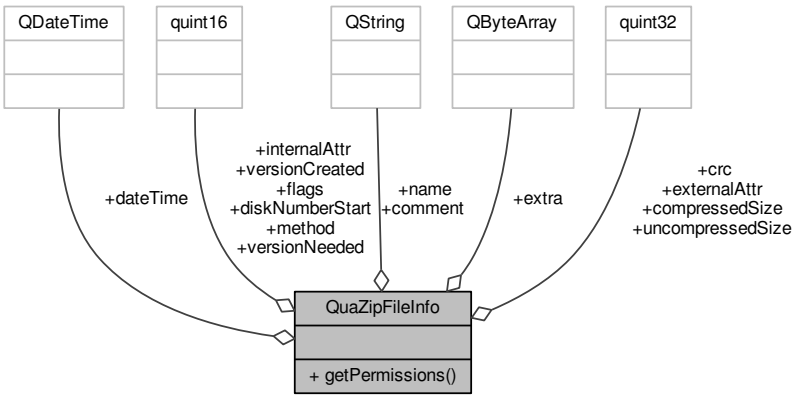
- src/ckb/quazip/quazipfile.h
- src/ckb/quazip/quazipfile.cpp

12.74 QuaZipFileInfo Struct Reference

Information about a file inside archive.

`#include <src/ckb/quazip/quazipfileinfo.h>`

Collaboration diagram for QuaZipFileInfo:



Public Member Functions

- QFile::Permissions getPermissions () const

  Get the file permissions.

Data Fields

- QString name

  File name.
- quint16 versionCreated

Version created by.

- quint16 versionNeeded

  Version needed to extract.

- quint16 flags

  General purpose flags.

- quint16 method

  Compression method.

- QDateTime dateTime

  Last modification date and time.

- quint32 crc

  CRC.

- quint32 compressedSize

  Compressed file size.

- quint32 uncompressedSize

  Uncompressed file size.

- quint16 diskNumberStart

  Disk number start.

- quint16 internalAttr

  Internal file attributes.

- quint32 externalAttr

  External file attributes.

- QString comment

  Comment.

- QByteArray extra

  Extra field.

12.74.1 Detailed Description

Deprecated Use QuaZipFileInfo64 instead. Not only it supports large files, but also more convenience methods as well.

Call QuaZip::getCurrentFileInfo() or QuaZipFile::getFileInfo() to fill this structure.

Definition at line 41 of file quazipfileinfo.h.

12.74.2 Member Function Documentation

12.74.2.1 QFile::Permissions QuaZipFileInfo::getPermissions ( ) const

Returns the high 16 bits of external attributes converted to QFile::Permissions.

Definition at line 52 of file quazipfileinfo.cpp.

References externalAttr, and permissionsFromExternalAttr().

```
53 {
54     return permissionsFromExternalAttr(externalAttr);
55 }
```

Here is the call graph for this function:

12.74.3 Field Documentation

12.74.3.1 QString QuaZipFileInfo::comment

Definition at line 67 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.2 quint32 QuaZipFileInfo::compressedSize

Definition at line 57 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.3 quint32 QuaZipFileInfo::crc

Definition at line 55 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.4 QDateTime QuaZipFileInfo::dateTime

Definition at line 53 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.5 quint16 QuaZipFileInfo::diskNumberStart

Definition at line 61 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.6 quint32 QuaZipFileInfo::externalAttr

Definition at line 65 of file quazipfileinfo.h.

Referenced by getPermissions(), and QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.7 QByteArray QuaZipFileInfo::extra

Definition at line 69 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.8 quint16 QuaZipFileInfo::flags

Definition at line 49 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.9 quint16 QuaZipFileInfo::internalAttr

Definition at line 63 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.10 quint16 QuaZipFileInfo::method

Definition at line 51 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.11 QString QuaZipFileInfo::name

Definition at line 43 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.12 quint32 QuaZipFileInfo::uncompressedSize

Definition at line 59 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.13 quint16 QuaZipFileInfo::versionCreated

Definition at line 45 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

12.74.3.14 quint16 QuaZipFileInfo::versionNeeded

Definition at line 47 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

The documentation for this struct was generated from the following files:

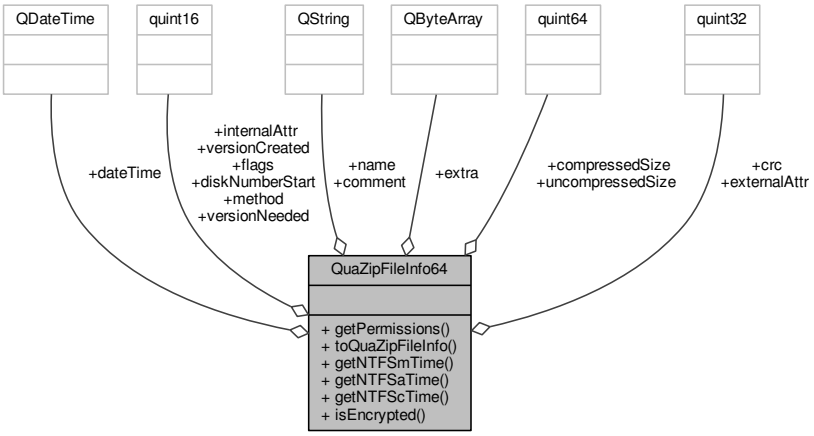
- src/ckb/quazip/quazipfileinfo.h
- src/ckb/quazip/quazipfileinfo.cpp

12.75 QuaZipFileInfo64 Struct Reference

Information about a file inside archive (with zip64 support).

`#include <src/ckb/quazip/quazipfileinfo.h>`

Collaboration diagram for QuaZipFileInfo64:



Public Member Functions

- QFile::Permissions getPermissions () const

  Get the file permissions.
- bool toQuaZipFileInfo (QuaZipFileInfo &info) const

  Converts to QuaZipFileInfo.
- QDateTime getNTFSmTime (int *fineTicks=NULL) const

  Returns the NTFS modification time.
- QDateTime getNTFSaTime (int *fineTicks=NULL) const

  Returns the NTFS access time.
- QDateTime getNTFScTime (int *fineTicks=NULL) const

  Returns the NTFS creation time.
- bool isEncrypted () const

  Checks whether the file is encrypted.

Data Fields

- QString name

  File name.
- quint16 versionCreated

  Version created by.
- quint16 versionNeeded

  Version needed to extract.
- quint16 flags

  General purpose flags.
- quint16 method

  Compression method.
- QDateTime dateTime

  Last modification date and time.
- quint32 crc

  CRC.

- quint64 compressedSize

  Compressed file size.
- quint64 uncompressedSize

  Uncompressed file size.
- quint16 diskNumberStart

  Disk number start.
- quint16 internalAttr

  Internal file attributes.
- quint32 externalAttr

  External file attributes.
- QString comment

  Comment.
- QByteArray extra

  Extra field.

12.75.1 Detailed Description

Call QuaZip::getCurrentFileInfo() or QuaZipFile::getFileInfo() to fill this structure.

Definition at line 81 of file quazipfileinfo.h.

12.75.2 Member Function Documentation

12.75.2.1 QDateTime QuaZipFileInfo64::getNTFSaTime (int * *fineTicks* = NULL) const

The getNTFS*Time() functions only work if there is an NTFS extra field present. Otherwise, they all return invalid null timestamps.

Parameters

fineTicks	If not NULL, the fractional part of milliseconds returned there, measured in 100-nanosecond ticks. Will be set to zero if there is no NTFS extra field.

See Also

dateTime
getNTFSmTime()
getNTFScTime()

Returns

The NTFS access time, UTC

Definition at line 168 of file quazipfileinfo.cpp.

References extra, and getNTFSTime().

```
169 {
170     return getNTFSTime(extra, 8, fineTicks);
171 }
```

Here is the call graph for this function:

12.75.2.2 QDateTime QuaZipFileInfo64::getNTFScTime (int * *fineTicks* = NULL) const

The getNTFS*Time() functions only work if there is an NTFS extra field present. Otherwise, they all return invalid null timestamps.

Parameters

fineTicks	If not NULL, the fractional part of milliseconds returned there, measured in 100-nanosecond ticks. Will be set to zero if there is no NTFS extra field.

See Also

dateTime
getNTFSmTime()
getNTFSaTime()

Returns

The NTFS creation time, UTC

Definition at line 173 of file quazipfileinfo.cpp.

References extra, and getNTFSTime().

```
174 {
175     return getNTFSTime(extra, 16, fineTicks);
176 }
```

Here is the call graph for this function:

12.75.2.3 QDateTime QuaZipFileInfo64::getNTFSmTime (int * *fineTicks* = NULL) const

The getNTFS*Time() functions only work if there is an NTFS extra field present. Otherwise, they all return invalid null timestamps.

Parameters

fineTicks	If not NULL, the fractional part of milliseconds returned there, measured in 100-nanosecond ticks. Will be set to zero if there is no NTFS extra field.

See Also

dateTime
getNTFSaTime()
getNTFScTime()

Returns

The NTFS modification time, UTC

Definition at line 163 of file quazipfileinfo.cpp.

References extra, and getNTFSTime().

```
164 {
165     return getNTFSTime(extra, 0, fineTicks);
166 }
```

Here is the call graph for this function:

12.75.2.4 QFile::Permissions QuaZipFileInfo64::getPermissions ( ) const

Returns the high 16 bits of external attributes converted to QFile::Permissions.

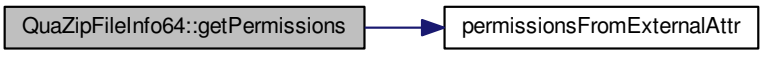
Definition at line 57 of file quazipfileinfo.cpp.

References externalAttr, and permissionsFromExternalAttr().

Referenced by JlCompress::extractFile().

```
58 {
59     return permissionsFromExternalAttr(externalAttr);
60 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.75.2.5 bool QuaZipFileInfo64::isEncrypted ( ) const [inline]

Definition at line 175 of file quazipfileinfo.h.

```
175 {return (flags & 1) != 0;}
```

12.75.2.6 bool QuaZipFileInfo64::toQuaZipFileInfo (QuaZipFileInfo & *info*) const

If any of the fields are greater than 0xFFFFFFFFu, they are set to 0xFFFFFFFFu exactly, not just truncated. This function should be mainly used for compatibility with the old code expecting QuaZipFileInfo, in the cases when it's impossible or otherwise unadvisable (due to ABI compatibility reasons, for example) to modify that old code to use QuaZipFileInfo64.

Returns

`true` if all fields converted correctly, `false` if an overflow occured.

Definition at line 62 of file quazipfileinfo.cpp.

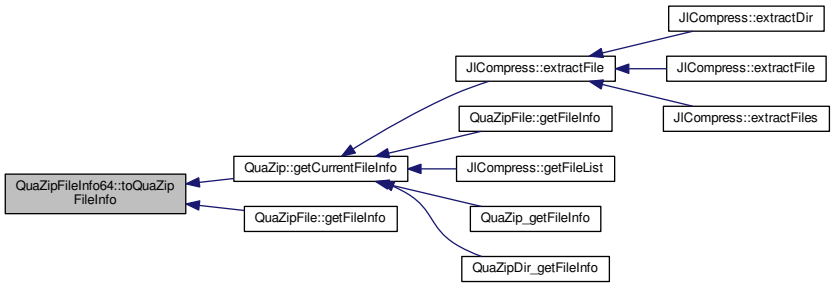
References QuaZipFileInfo::comment, comment, QuaZipFileInfo::compressedSize, compressedSize, QuaZipFileInfo::crc, crc, QuaZipFileInfo::dateTime, dateTime, QuaZipFileInfo::diskNumberStart, diskNumberStart, QuaZipFileInfo::externalAttr, externalAttr, QuaZipFileInfo::extra, extra, QuaZipFileInfo::flags, flags, QuaZipFileInfo::internalAttr, internalAttr, QuaZipFileInfo::method, method, QuaZipFileInfo::name, name, QuaZipFileInfo::uncompressedSize, uncompressedSize, QuaZipFileInfo::versionCreated, versionCreated, QuaZipFileInfo::versionNeeded, and versionNeeded.

Referenced by QuaZip::getCurrentFileInfo(), and QuaZipFile::getFileInfo().

```
63 {
64     bool noOverflow = true;
65     info.name = name;
66     info.versionCreated = versionCreated;
67     info.versionNeeded = versionNeeded;
68     info.flags = flags;
69     info.method = method;
70     info.dateTime = dateTime;
71     info.crc = crc;
72     if (compressedSize > 0xFFFFFFFFu) {
73         info.compressedSize = 0xFFFFFFFFu;
74         noOverflow = false;
75     } else {
76         info.compressedSize = compressedSize;
77     }
78     if (uncompressedSize > 0xFFFFFFFFu) {
79         info.uncompressedSize = 0xFFFFFFFFu;
80         noOverflow = false;
81     } else {
82         info.uncompressedSize = uncompressedSize;
83     }
84     info.diskNumberStart = diskNumberStart;
85     info.internalAttr = internalAttr;
```

```
86     info.externalAttr = externalAttr;
87     info.comment = comment;
88     info.extra = extra;
89     return noOverflow;
90 }
```

Here is the caller graph for this function:



12.75.3 Field Documentation

12.75.3.1 QString QuaZipFileInfo64::comment

Definition at line 114 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), and toQuaZipFileInfo().

12.75.3.2 quint64 QuaZipFileInfo64::compressedSize

Definition at line 104 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.3 quint32 QuaZipFileInfo64::crc

Definition at line 102 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.4 QDateTime QuaZipFileInfo64::dateTime

This is the time stored in the standard ZIP header. This format only allows to store time with 2-second precision, so the seconds will always be even and the milliseconds will always be zero. If you need more precise date and time, you can try to call the getNTFSmTime() function or its siblings, provided that the archive itself contains these NTFS times.

Definition at line 100 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), and toQuaZipFileInfo().

12.75.3.5 quint16 QuaZipFileInfo64::diskNumberStart

Definition at line 108 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.6 quint32 QuaZipFileInfo64::externalAttr

Definition at line 112 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), getPermissions(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.7 QByteArray QuaZipFileInfo64::extra

Definition at line 116 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), getNTFSaTime(), getNTFScTime(), getNTFSmTime(), and toQuaZipFileInfo().

12.75.3.8 quint16 QuaZipFileInfo64::flags

Definition at line 89 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.9 quint16 QuaZipFileInfo64::internalAttr

Definition at line 110 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.10 quint16 QuaZipFileInfo64::method

Definition at line 91 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.11 QString QuaZipFileInfo64::name

Definition at line 83 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), JlCompress::getFileList(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.12 quint64 QuaZipFileInfo64::uncompressedSize

Definition at line 106 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.13 quint16 QuaZipFileInfo64::versionCreated

Definition at line 85 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

12.75.3.14 quint16 QuaZipFileInfo64::versionNeeded

Definition at line 87 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

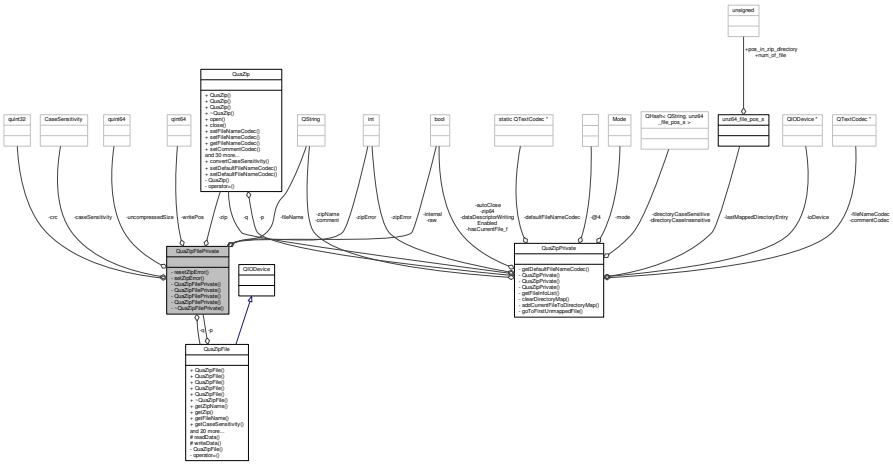
The documentation for this struct was generated from the following files:

- src/ckb/quazip/quazipfileinfo.h
- src/ckb/quazip/quazipfileinfo.cpp

12.76 QuaZipFilePrivate Class Reference

The implementation class for QuaZip.

Collaboration diagram for QuaZipFilePrivate:



Private Member Functions

- void resetZipError () const

  Resets zipError.
- void setZipError (int zipError) const

  Sets the zip error.
- QuaZipFilePrivate (QuaZipFile *q)

  The constructor for the corresponding QuaZipFile constructor.
- QuaZipFilePrivate (QuaZipFile *q, const QString &zipName)

  The constructor for the corresponding QuaZipFile constructor.
- QuaZipFilePrivate (QuaZipFile *q, const QString &zipName, const QString &fileName, QuaZip::CaseSensitivity cs)

  The constructor for the corresponding QuaZipFile constructor.
- QuaZipFilePrivate (QuaZipFile *q, QuaZip *zip)

  The constructor for the QuaZipFile constructor accepting a file name.
- ~QuaZipFilePrivate ()

  The destructor.

Private Attributes

- QuaZipFile * q

  The pointer to the associated QuaZipFile instance.
- QuaZip * zip

  The QuaZip object to work with.
- QString fileName

  The file name.

- QuaZip::CaseSensitivity caseSensitivity

  Case sensitivity mode.
- bool raw

  Whether this file is opened in the raw mode.
- qint64 writePos

  Write position to keep track of.
- quint64 uncompressedSize

  Uncompressed size to write along with a raw file.
- quint32 crc

  CRC to write along with a raw file.
- bool internal

  Whether zip points to an internal QuaZip instance.
- int zipError

  The last error.

Friends

- class QuaZipFile

12.76.1 Detailed Description

This class contains all the private stuff for the QuaZipFile class, thus allowing to preserve binary compatibility between releases, the technique known as the Pimpl (private implementation) idiom.

Definition at line 37 of file quazipfile.cpp.

12.76.2 Constructor & Destructor Documentation

12.76.2.1 QuaZipFilePrivate::QuaZipFilePrivate (QuaZipFile * *q*) [inline], [private]

Definition at line 78 of file quazipfile.cpp.

```
78                                          :
79       q(q), zip(NULL), internal(true), zipError(UNZ_OK) {}
```

12.76.2.2 QuaZipFilePrivate::QuaZipFilePrivate (QuaZipFile * *q*, const QString & *zipName*) [inline], [private]

Definition at line 81 of file quazipfile.cpp.

```
81                                                                  :
82       q(q), internal(true), zipError(UNZ_OK)
83       {
84         zip=new QuaZip(zipName);
85       }
```

12.76.2.3 QuaZipFilePrivate::QuaZipFilePrivate (QuaZipFile * *q*, const QString & *zipName*, const QString & *fileName*, QuaZip::CaseSensitivity *cs*) [inline], [private]

Definition at line 87 of file quazipfile.cpp.

```
88                                  :
89       q(q), internal(true), zipError(UNZ_OK)
90       {
91         zip=new QuaZip(zipName);
92         this->fileName=fileName;
93         if (this->fileName.startsWith('/'))
94             this->fileName = this->fileName.mid(1);
95         this->caseSensitivity=cs;
96       }
```

12.76.2.4 QuaZipFilePrivate::QuaZipFilePrivate (QuaZipFile * *q*, QuaZip * *zip*) [inline], [private]

Definition at line 98 of file quazipfile.cpp.

```
98                                                          :
99       q(q), zip(zip), internal(false), zipError(UNZ_OK) {}
```

12.76.2.5 QuaZipFilePrivate::~QuaZipFilePrivate ( ) [inline], [private]

Definition at line 101 of file quazipfile.cpp.

```
102     {
103       if (internal)
104         delete zip;
105     }
```

12.76.3 Member Function Documentation

12.76.3.1 void QuaZipFilePrivate::resetZipError ( ) const [inline], [private]

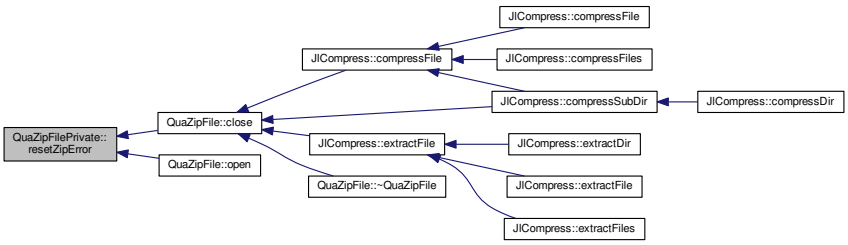
Definition at line 69 of file quazipfile.cpp.

References UNZ_OK.

Referenced by QuaZipFile::close(), and QuaZipFile::open().

```
69 {setZipError(UNZ_OK);}
```

Here is the caller graph for this function:



12.76.3.2 void QuaZipFilePrivate::setZipError (int *zipError*) const [private]

This function is marked as const although it changes one field. This allows to call it from const functions that don't change anything by themselves.

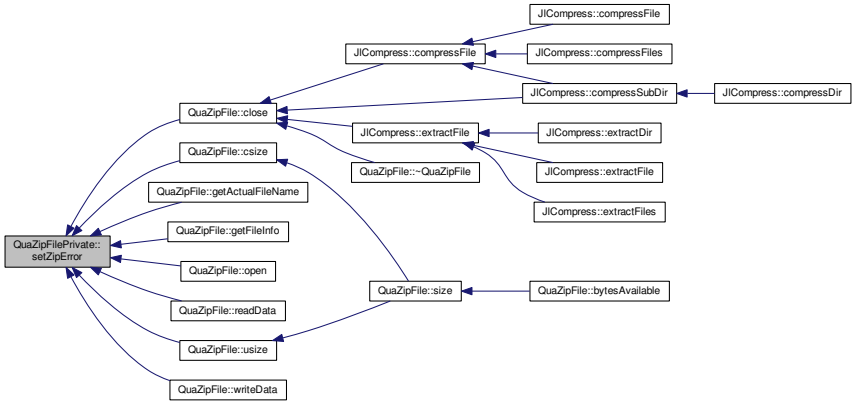
Definition at line 211 of file quazipfile.cpp.

References q, UNZ_OK, and zipError.

Referenced by QuaZipFile::close(), QuaZipFile::csize(), QuaZipFile::getActualFileName(), QuaZipFile::getFileInfo(), QuaZipFile::open(), QuaZipFile::readData(), QuaZipFile::usize(), and QuaZipFile::writeData().

```
212 {
213   QuaZipFilePrivate *fakeThis = const_cast<QuaZipFilePrivate*>(this); //
       non-const
214   fakeThis->zipError=zipError;
215   if(zipError==UNZ_OK)
216     q->setErrorString(QString());
217   else
218     q->setErrorString(QuaZipFile::tr("ZIP/UNZIP API error %1").arg(zipError));
219 }
```

Here is the caller graph for this function:



12.76.4 Friends And Related Function Documentation

12.76.4.1 friend class QuaZipFile [friend]

Definition at line 38 of file quazipfile.cpp.

12.76.5 Field Documentation

12.76.5.1 QuaZip::CaseSensitivity QuaZipFilePrivate::caseSensitivity [private]

Definition at line 47 of file quazipfile.cpp.

Referenced by QuaZipFile::getCaseSensitivity(), QuaZipFile::open(), and QuaZipFile::setFileName().

12.76.5.2 quint32 QuaZipFilePrivate::crc [private]

Definition at line 59 of file quazipfile.cpp.

Referenced by QuaZipFile::close(), and QuaZipFile::open().

12.76.5.3 QString QuaZipFilePrivate::fileName [private]

Definition at line 45 of file quazipfile.cpp.

Referenced by QuaZipFile::getFileName(), QuaZipFile::open(), QuaZipFile::setFileName(), and QuaZipFile::setZip().

12.76.5.4 bool QuaZipFilePrivate::internal [private]

This is true if the archive was opened by name, rather than by supplying an existing QuaZip instance.

Definition at line 65 of file quazipfile.cpp.

Referenced by QuaZipFile::close(), QuaZipFile::getZip(), QuaZipFile::open(), QuaZipFile::setFileName(), QuaZipFile::setZip(), and QuaZipFile::setZipName().

12.76.5.5 QuaZipFile* QuaZipFilePrivate::q [private]

Definition at line 41 of file quazipfile.cpp.

Referenced by setZipError().

12.76.5.6 bool QuaZipFilePrivate::raw [private]

Definition at line 49 of file quazipfile.cpp.

Referenced by QuaZipFile::isRaw(), QuaZipFile::open(), and QuaZipFile::size().

12.76.5.7 quint64 QuaZipFilePrivate::uncompressedSize [private]

Definition at line 57 of file quazipfile.cpp.

Referenced by QuaZipFile::close(), and QuaZipFile::open().

12.76.5.8 qint64 QuaZipFilePrivate::writePos [private]

QIODevice::pos() is broken for non-seekable devices, so we need our own position.

Definition at line 55 of file quazipfile.cpp.

Referenced by QuaZipFile::open(), QuaZipFile::pos(), QuaZipFile::size(), and QuaZipFile::writeData().

12.76.5.9 QuaZip* QuaZipFilePrivate::zip [private]

Definition at line 43 of file quazipfile.cpp.

Referenced by QuaZipFile::atEnd(), QuaZipFile::close(), QuaZipFile::csize(), QuaZipFile::getActualFileName(), QuaZipFile::getFileInfo(), QuaZipFile::getZip(), QuaZipFile::getZipName(), QuaZipFile::open(), QuaZipFile::pos(), QuaZipFile::readData(), QuaZipFile::setFileName(), QuaZipFile::setZip(), QuaZipFile::setZipName(), QuaZipFile::usize(), and QuaZipFile::writeData().

12.76.5.10 int QuaZipFilePrivate::zipError [private]

Definition at line 67 of file quazipfile.cpp.

Referenced by QuaZipFile::close(), QuaZipFile::csize(), QuaZipFile::getFileInfo(), QuaZipFile::getZipError(), QuaZipFile::open(), setZipError(), QuaZipFile::usize(), and QuaZipFile::writeData().

The documentation for this class was generated from the following file:

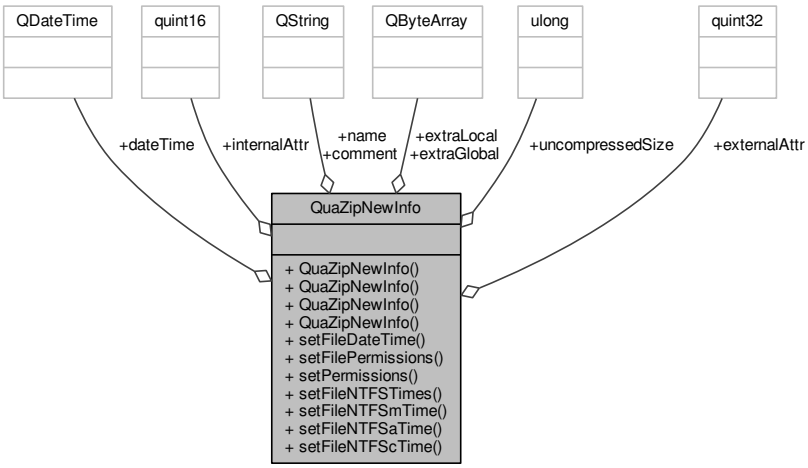
- src/ckb/quazip/quazipfile.cpp

12.77 QuaZipNewInfo Struct Reference

Information about a file to be created.

`#include <src/ckb/quazip/quazipnewinfo.h>`

Collaboration diagram for QuaZipNewInfo:



Public Member Functions

- QuaZipNewInfo (const QString &name)

  Constructs QuaZipNewInfo instance.
- QuaZipNewInfo (const QString &name, const QString &file)

  Constructs QuaZipNewInfo instance.
- QuaZipNewInfo (const QuaZipFileInfo &existing)

  Initializes the new instance from existing file info.
- QuaZipNewInfo (const QuaZipFileInfo64 &existing)

  Initializes the new instance from existing file info.
- void setFileDateTime (const QString &file)

  Sets the file timestamp from the existing file.
- void setFilePermissions (const QString &file)

  Sets the file permissions from the existing file.
- void setPermissions (QFile::Permissions permissions)

  Sets the file permissions.
- void setFileNTFSTimes (const QString &fileName)

  Sets the NTFS times from an existing file.
- void setFileNTFSmTime (const QDateTime &mTime, int fineTicks=0)

  Sets the NTFS modification time.
- void setFileNTFSaTime (const QDateTime &aTime, int fineTicks=0)

  Sets the NTFS access time.
- void setFileNTFScTime (const QDateTime &cTime, int fineTicks=0)

  Sets the NTFS creation time.

Data Fields

- QString name

  File name.
- QDateTime dateTime

  File timestamp.
- quint16 internalAttr

  File internal attributes.
- quint32 externalAttr

  File external attributes.
- QString comment

  File comment.
- QByteArray extraLocal

  File local extra field.
- QByteArray extraGlobal

  File global extra field.
- ulong uncompressedSize

  Uncompressed file size.

12.77.1 Detailed Description

This structure holds information about a file to be created inside ZIP archive. At least name should be set to something correct before passing this structure to QuaZipFile::open(OpenMode,const QuaZipNewInfo&,int,int,bool).

Zip64 support of this structure is slightly limited: in the raw mode (when a pre-compressed file is written into a ZIP file as-is), it is necessary to specify the uncompressed file size and the appropriate field is 32 bit. Since the raw mode is used extremely rare, there is no real need to have a separate QuaZipNewInfo64 structure like QuaZipFileInfo64. It may be added in the future though, if there is a demand for the raw mode with zip64 archives.

Definition at line 50 of file quazipnewinfo.h.

12.77.2 Constructor & Destructor Documentation

12.77.2.1 QuaZipNewInfo::QuaZipNewInfo (const QString & *name*)

Initializes name with *name*, dateTime with current date and time. Attributes are initialized with zeros, comment and extra field with null values.

Definition at line 79 of file quazipnewinfo.cpp.

```
79                                                :
80   name(name), dateTime(QDateTime::currentDateTime()),
      internalAttr(0), externalAttr(0),
81   uncompressedSize(0)
82 {
83 }
```

12.77.2.2 QuaZipNewInfo::QuaZipNewInfo (const QString & *name*, const QString & *file*)

Initializes name with *name*. Timestamp and permissions are taken from the specified file. If the *file* does not exists or its timestamp is inaccessible (e. g. you do not have read permission for the directory file in), uses current time and zero permissions. Other attributes are initialized with zeros, comment and extra field with null values.

See Also

setFileDateTime()

Definition at line 85 of file quazipnewinfo.cpp.

References dateTime, and QuaZipNewInfo_setPermissions().

```
85                                                                    :
86   name(name), internalAttr(0), externalAttr(0),
      uncompressedSize(0)
87 {
88   QFileInfo info(file);
89   QDateTime lm = info.lastModified();
90   if (!info.exists()) {
91     dateTime = QDateTime::currentDateTime();
92   } else {
93     dateTime = lm;
94     QuaZipNewInfo_setPermissions(this, info.permissions(), info.isDir());
95   }
96 }
```

Here is the call graph for this function:

12.77.2.3 QuaZipNewInfo::QuaZipNewInfo (const QuaZipFileInfo & *existing*)

Mainly used when copying files between archives.

Both extra fields are initialized to existing.extra. QuaZipNewInfo

Parameters

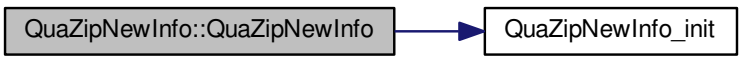
existing	

Definition at line 69 of file quazipnewinfo.cpp.

References QuaZipNewInfo_init().

```
70 {
71     QuaZipNewInfo_init(*this, existing);
72 }
```

Here is the call graph for this function:



12.77.2.4 QuaZipNewInfo::QuaZipNewInfo (const QuaZipFileInfo64 & *existing*)

Mainly used when copying files between archives.

Both extra fields are initialized to existing.extra. QuaZipNewInfo

Parameters

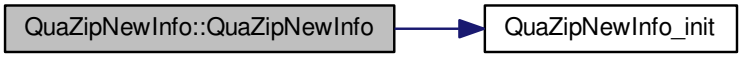
existing	

Definition at line 74 of file quazipnewinfo.cpp.

References QuaZipNewInfo_init().

```
75 {
76     QuaZipNewInfo_init(*this, existing);
77 }
```

Here is the call graph for this function:



12.77.3 Member Function Documentation

12.77.3.1 void QuaZipNewInfo::setFileDateTime (const QString & *file*)

Use this function to set the file timestamp from the existing file. Use it like this:

```
QuaZipFile zipFile(&zip);
QFile file("file-to-add");
file.open(QIODevice::ReadOnly);
QuaZipNewInfo info("file-name-in-archive");
info.setFileDateTime("file-to-add"); // take the timestamp from file
zipFile.open(QIODevice::WriteOnly, info);
```

This function does not change dateTime if some error occured (e. g. file is inaccessible).

Definition at line 98 of file quazipnewinfo.cpp.

References dateTime.

```
99 {
100   QFileInfo info(file);
101   QDateTime lm = info.lastModified();
102   if (info.exists())
103     dateTime = lm;
104 }
```

12.77.3.2 void QuaZipNewInfo::setFileNTFSaTime (const QDateTime & *aTime,* int *fineTicks* = 0)

The time is written into the NTFS record in both the local and the global extra fields, updating the existing record if there is one, or creating a new one and appending it to the end of each extra field. When updating an existing record, all other fields are left intact.

Parameters

aTime	The new access time.
fineTicks	The fractional part of milliseconds, in 100-nanosecond ticks (i. e. 9999 ticks = 999.9 microsecond). Values greater than 9999 will add milliseconds or even seconds, but this can be confusing and therefore is discouraged.

Definition at line 267 of file quazipnewinfo.cpp.

References extraGlobal, extraLocal, and setNTFSTime().

Referenced by setFileNTFSTimes().

```
268 {
269     setNTFSTime(extraLocal, aTime, 8, fineTicks);
270     setNTFSTime(extraGlobal, aTime, 8, fineTicks);
271 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.77.3.3 void QuaZipNewInfo::setFileNTFScTime (const QDateTime & *cTime*, int *fineTicks* = 0)

The time is written into the NTFS record in both the local and the global extra fields, updating the existing record if there is one, or creating a new one and appending it to the end of each extra field. When updating an existing record, all other fields are left intact.

Parameters

cTime	The new creation time.
fineTicks	The fractional part of milliseconds, in 100-nanosecond ticks (i. e. 9999 ticks = 999.9 microsecond). Values greater than 9999 will add milliseconds or even seconds, but this can be confusing and therefore is discouraged.

Definition at line 273 of file quazipnewinfo.cpp.

References extraGlobal, extraLocal, and setNTFSTime().

Referenced by setFileNTFSTimes().

```
274 {
275     setNTFSTime(extraLocal, cTime, 16, fineTicks);
276     setNTFSTime(extraGlobal, cTime, 16, fineTicks);
277 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.77.3.4 void QuaZipNewInfo::setFileNTFSmTime (const QDateTime & *mTime*, int *fineTicks* = 0)

The time is written into the NTFS record in both the local and the global extra fields, updating the existing record if there is one, or creating a new one and appending it to the end of each extra field. When updating an existing record, all other fields are left intact.

Parameters

mTime	The new modification time.
fineTicks	The fractional part of milliseconds, in 100-nanosecond ticks (i. e. 9999 ticks = 999.9 microsecond). Values greater than 9999 will add milliseconds or even seconds, but this can be confusing and therefore is discouraged.

Definition at line 261 of file quazipnewinfo.cpp.

References extraGlobal, extraLocal, and setNTFSTime().

Referenced by setFileNTFSTimes().

```
262 {
263     setNTFSTime(extraLocal, mTime, 0, fineTicks);
264     setNTFSTime(extraGlobal, mTime, 0, fineTicks);
265 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

12.77.3.5 void QuaZipNewInfo::setFileNTFSTimes (const QString & *fileName*)

If the file doesn't exist, a warning is printed to the stderr and nothing is done. Otherwise, all three times, as reported by QFileInfo::lastModified(), QFileInfo::lastRead() and QFileInfo::created(), are written to the NTFS extra field record.

The NTFS record is written to both the local and the global extra fields, updating the existing record if there is one, or creating a new one and appending it to the end of each extra field.

The microseconds will be zero, as they aren't reported by QFileInfo.

Parameters

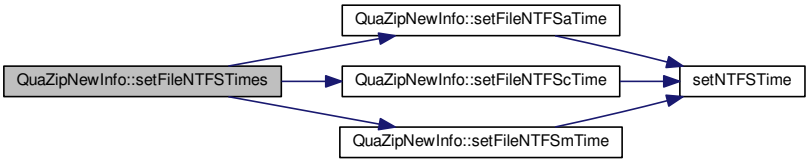
fileName	

Definition at line 118 of file quazipnewinfo.cpp.

References setFileNTFSaTime(), setFileNTFScTime(), and setFileNTFSmTime().

```
119 {
120     QFileInfo fi(fileName);
121     if (!fi.exists()) {
122         qWarning("QuaZipNewInfo::setFileNTFSTimes(): '%s' doesn't exist",
123                  fileName.toUtf8().constData());
124         return;
125     }
126     setFileNTFSmTime(fi.lastModified());
127     setFileNTFSaTime(fi.lastRead());
128     setFileNTFScTime(fi.created());
129 }
```

Here is the call graph for this function:



12.77.3.6 void QuaZipNewInfo::setFilePermissions (const QString & *file*)

Takes permissions from the file and sets the high 16 bits of external attributes. Uses QFileInfo to get permissions on all platforms.

Definition at line 106 of file quazipnewinfo.cpp.

References QuaZipNewInfo_setPermissions().

```
107 {
108     QFileInfo info = QFileInfo(file);
109     QFile::Permissions perm = info.permissions();
110     QuaZipNewInfo_setPermissions(this, perm, info.isDir());
111 }
```

Here is the call graph for this function:

12.77.3.7 void QuaZipNewInfo::setPermissions (QFile::Permissions *permissions*)

Modifies the highest 16 bits of external attributes. The type part is set to dir if the name ends with a slash, and to regular file otherwise.

Definition at line 113 of file quazipnewinfo.cpp.

References name, and QuaZipNewInfo_setPermissions().

```
114 {
115     QuaZipNewInfo_setPermissions(this, permissions,
    name.endsWith('/'));
116 }
```

Here is the call graph for this function:

12.77.4 Field Documentation

12.77.4.1 QString QuaZipNewInfo::comment

Will be encoded using QuaZip::getCommentCodec().

Definition at line 75 of file quazipnewinfo.h.

Referenced by QuaZipFile::open().

12.77.4.2 QDateTime QuaZipNewInfo::dateTime

This is the last file modification date and time. Will be stored in the archive central directory. It is a good practice to set it to the source file timestamp instead of archive creating time. Use setFileDateTime() or QuaZipNewInfo(const QString&, const QString&).

Definition at line 62 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), QuaZipNewInfo(), and setFileDateTime().

12.77.4.3 quint32 QuaZipNewInfo::externalAttr

The highest 16 bits contain Unix file permissions and type (dir or file). The constructor QuaZipNewInfo(const QString&, const QString&) takes permissions from the provided file.

Definition at line 71 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), and QuaZipNewInfo_setPermissions().

12.77.4.4 QByteArray QuaZipNewInfo::extraGlobal

Definition at line 79 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), setFileNTFSaTime(), setFileNTFScTime(), and setFileNTFSmTime().

12.77.4.5 QByteArray QuaZipNewInfo::extraLocal

Definition at line 77 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), setFileNTFSaTime(), setFileNTFScTime(), and setFileNTFSmTime().

12.77.4.6 quint16 QuaZipNewInfo::internalAttr

Definition at line 64 of file quazipnewinfo.h.

Referenced by QuaZipFile::open().

12.77.4.7 QString QuaZipNewInfo::name

This field holds file name inside archive, including path relative to archive root.

Definition at line 55 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), and setPermissions().

12.77.4.8 ulong QuaZipNewInfo::uncompressedSize

This is only needed if you are using raw file zipping mode, i. e. adding precompressed file in the zip archive.

Definition at line 84 of file quazipnewinfo.h.

Referenced by QuaZipFile::open().

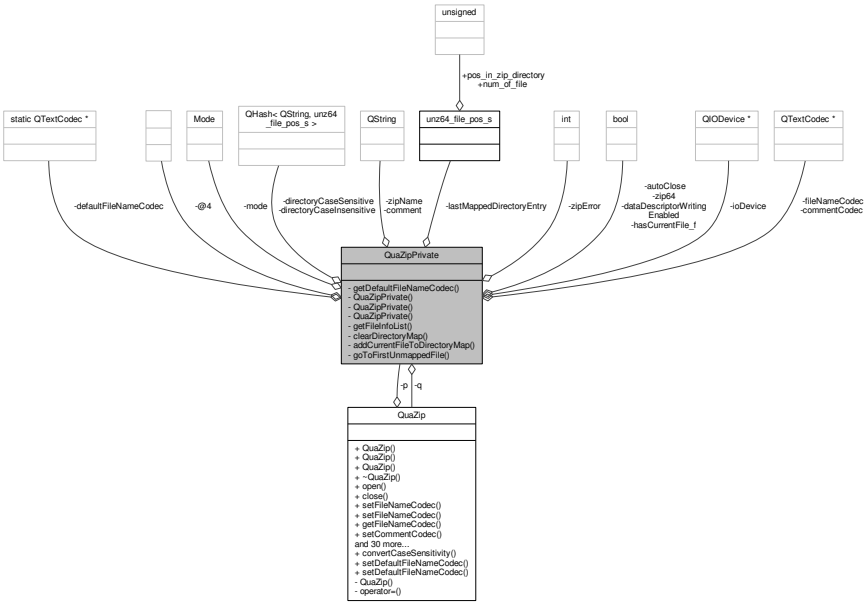
The documentation for this struct was generated from the following files:

- src/ckb/quazip/quazipnewinfo.h
- src/ckb/quazip/quazipnewinfo.cpp

12.78 QuaZipPrivate Class Reference

All the internal stuff for the QuaZip class.

Collaboration diagram for QuaZipPrivate:



Private Member Functions

- QTextCodec * getDefaultFileNameCodec ()
- QuaZipPrivate (QuaZip *q)

  The constructor for the corresponding QuaZip constructor.
- QuaZipPrivate (QuaZip *q, const QString &zipName)

  The constructor for the corresponding QuaZip constructor.
- QuaZipPrivate (QuaZip *q, QIODevice *ioDevice)

  The constructor for the corresponding QuaZip constructor.
- template<typename TFileInfo >
  bool getFileInfoList (QList< TFileInfo > *result) const

  Returns either a list of file names or a list of QuaZipFileInfo.
- void clearDirectoryMap ()

  Stores map of filenames and file locations for unzipping.
- void addCurrentFileToDirectoryMap (const QString &fileName)
- bool goToFirstUnmappedFile ()

Private Attributes

- QuaZip * q

  The pointer to the corresponding QuaZip instance.
- QTextCodec * fileNameCodec

  The codec for file names.
- QTextCodec * commentCodec

  The codec for comments.
- QString zipName

  The archive file name.
- QIODevice * ioDevice

  The device to access the archive.

- QString comment

  The global comment.
- QuaZip::Mode mode

  The open mode.
- union {
  unzFile **unzFile_f**
  The internal handle for UNZIP modes.
  zipFile **zipFile_f**
  The internal handle for ZIP modes.
  };
- bool hasCurrentFile_f

  Whether a current file is set.
- int zipError

  The last error.
- bool dataDescriptorWritingEnabled

  Whether the data descriptor writing mode is enabled.
- bool zip64

  The zip64 mode.
- bool autoClose

  The auto-close flag.
- QHash< QString, unz64_file_pos > directoryCaseSensitive
- QHash< QString, unz64_file_pos > directoryCaseInsensitive
- unz64_file_pos lastMappedDirectoryEntry

Static Private Attributes

- static QTextCodec * defaultFileNameCodec = NULL

Friends

- class QuaZip

12.78.1 Detailed Description

This class keeps all the private stuff for the QuaZip class so it can be changed without breaking binary compatibility, according to the Pimpl idiom.

Definition at line 39 of file quazip.cpp.

12.78.2 Constructor & Destructor Documentation

12.78.2.1 QuaZipPrivate::QuaZipPrivate (QuaZip * *q*) [inline], [private]

Definition at line 81 of file quazip.cpp.

References lastMappedDirectoryEntry, unz64_file_pos_s::num_of_file, and unz64_file_pos_s::pos_in_zip_directory.

```
81                                  :
82       q(q),
83       fileNameCodec(getDefaultFileNameCodec()),
84       commentCodec(QTextCodec::codecForLocale()),
85       ioDevice(NULL),
86       mode(QuaZip::mdNotOpen),
```

```
87        hasCurrentFile_f(false),
88        zipError(UNZ_OK),
89        dataDescriptorWritingEnabled(true),
90        zip64(false),
91        autoClose(true)
92    {
93        lastMappedDirectoryEntry.num_of_file = 0;
94        lastMappedDirectoryEntry.pos_in_zip_directory = 0;
95    }
```

12.78.2.2 QuaZipPrivate::QuaZipPrivate (QuaZip * *q*, const QString & *zipName*) [inline], [private]

Definition at line 97 of file quazip.cpp.

References lastMappedDirectoryEntry, unz64_file_pos_s::num_of_file, and unz64_file_pos_s::pos_in_zip_directory.

```
97                                                    :
98        q(q),
99        fileNameCodec(getDefaultFileNameCodec()),
100       commentCodec(QTextCodec::codecForLocale()),
101       zipName(zipName),
102       ioDevice(NULL),
103       mode(QuaZip::mdNotOpen),
104       hasCurrentFile_f(false),
105       zipError(UNZ_OK),
106       dataDescriptorWritingEnabled(true),
107       zip64(false),
108       autoClose(true)
109   {
110       lastMappedDirectoryEntry.num_of_file = 0;
111       lastMappedDirectoryEntry.pos_in_zip_directory = 0;
112   }
```

12.78.2.3 QuaZipPrivate::QuaZipPrivate (QuaZip * *q*, QIODevice * *ioDevice*) [inline], [private]

Definition at line 114 of file quazip.cpp.

References lastMappedDirectoryEntry, unz64_file_pos_s::num_of_file, and unz64_file_pos_s::pos_in_zip_directory.

```
114                                                    :
115       q(q),
116       fileNameCodec(getDefaultFileNameCodec()),
117       commentCodec(QTextCodec::codecForLocale()),
118       ioDevice(ioDevice),
119       mode(QuaZip::mdNotOpen),
120       hasCurrentFile_f(false),
121       zipError(UNZ_OK),
122       dataDescriptorWritingEnabled(true),
123       zip64(false),
124       autoClose(true)
125   {
126       lastMappedDirectoryEntry.num_of_file = 0;
127       lastMappedDirectoryEntry.pos_in_zip_directory = 0;
128   }
```

12.78.3 Member Function Documentation

12.78.3.1 void QuaZipPrivate::addCurrentFileToDirectoryMap (const QString & *fileName*) [inline], [private]

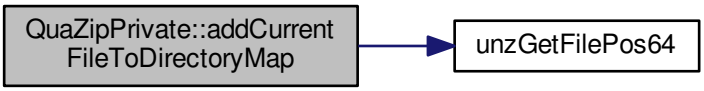
Definition at line 153 of file quazip.cpp.

References directoryCaseInsensitive, directoryCaseSensitive, hasCurrentFile_f, lastMappedDirectoryEntry, unz64_file_pos_s::pos_in_zip_directory, and unzGetFilePos64().

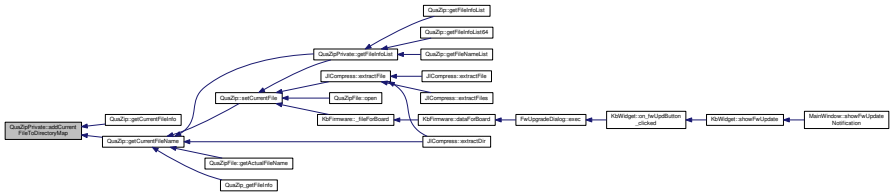
Referenced by QuaZip::getCurrentFileInfo(), and QuaZip::getCurrentFileName().

```
154 {
155     if (!hasCurrentFile_f || fileName.isEmpty()) {
156         return;
157     }
158     // Adds current file to filename map as fileName
159     unz64_file_pos fileDirectoryPos;
160     unzGetFilePos64(unzFile_f, &fileDirectoryPos);
161     directoryCaseSensitive.insert(fileName, fileDirectoryPos);
162     // Only add lowercase to directory map if not already there
163     // ensures only map the first one seen
164     QString lower = fileName.toLower();
165     if (!directoryCaseInsensitive.contains(lower))
166         directoryCaseInsensitive.insert(lower, fileDirectoryPos);
167     // Mark last one
168     if (fileDirectoryPos.pos_in_zip_directory >
      lastMappedDirectoryEntry.pos_in_zip_directory)
169         lastMappedDirectoryEntry = fileDirectoryPos;
170 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.78.3.2 void QuaZipPrivate::clearDirectoryMap ( ) [inline],[private]

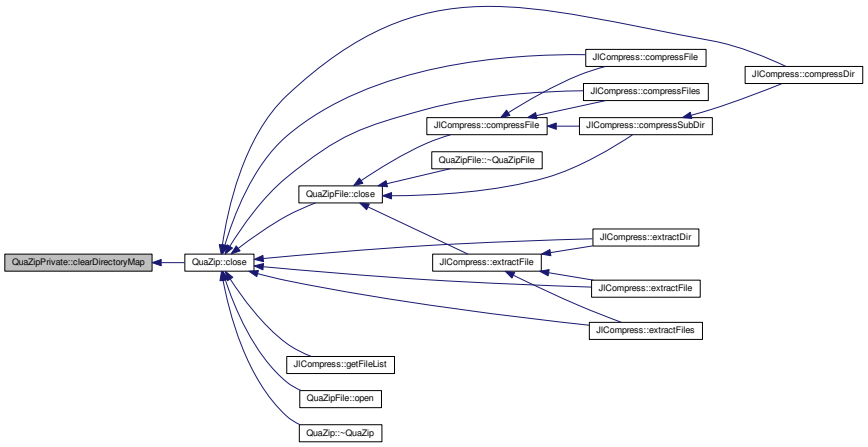
Definition at line 145 of file quazip.cpp.

References directoryCaseInsensitive, directoryCaseSensitive, lastMappedDirectoryEntry, unz64_file_pos_s::num_of_file, and unz64_file_pos_s::pos_in_zip_directory.

Referenced by QuaZip::close().

```
146 {
147     directoryCaseInsensitive.clear();
148     directoryCaseSensitive.clear();
149     lastMappedDirectoryEntry.num_of_file = 0;
150     lastMappedDirectoryEntry.pos_in_zip_directory = 0;
151 }
```

Here is the caller graph for this function:



12.78.3.3 QTextCodec* QuaZipPrivate::getDefaultFileNameCodec ( ) [inline], [private]

Definition at line 72 of file quazip.cpp.

References defaultFileNameCodec.

```
73     {
74         if (defaultFileNameCodec == NULL) {
75             return QTextCodec::codecForLocale();
76         } else {
77             return defaultFileNameCodec;
78         }
79     }
```

12.78.3.4 template<typename TFileInfo > bool QuaZipPrivate::getFileInfoList (QList< TFileInfo > * *result*) const [private]

Definition at line 687 of file quazip.cpp.

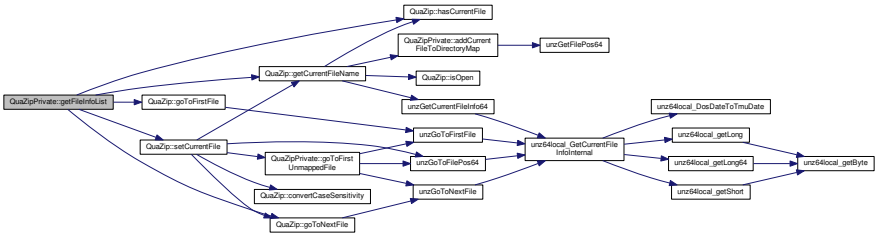
References QuaZip::getCurrentFileName(), QuaZip::goToFirstFile(), QuaZip::goToNextFile(), QuaZip::hasCurrentFile(), QuaZip::mdUnzip, mode, q, QuaZip::setCurrentFile(), UNZ_OK, and zipError.

Referenced by QuaZip::getFileInfoList(), QuaZip::getFileInfoList64(), and QuaZip::getFileNameList().

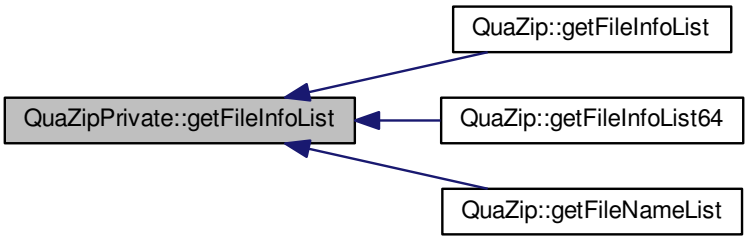
```
688 {
689   QuaZipPrivate *fakeThis=const_cast<QuaZipPrivate*>(this);
690   fakeThis->zipError=UNZ_OK;
691   if (mode!=QuaZip::mdUnzip) {
692     qWarning("QuaZip::getFileNameList/getFileInfoList(): "
693              "ZIP is not open in mdUnzip mode");
694     return false;
695   }
696   QString currentFile;
697   if (q->hasCurrentFile()) {
698       currentFile = q->getCurrentFileName();
699   }
700   if (q->goToFirstFile()) {
701       do {
702           bool ok;
703           result->append(QuaZip_getFileInfo<TFileInfo>(q, &ok));
704           if (!ok)
705               return false;
706       } while (q->goToNextFile());
707   }
708   if (zipError != UNZ_OK)
709       return false;
```

```
710   if (currentFile.isEmpty()) {
711       if (!q->goToFirstFile())
712           return false;
713   } else {
714       if (!q->setCurrentFile(currentFile))
715           return false;
716   }
717   return true;
718 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.78.3.5 bool QuaZipPrivate::goToFirstUnmappedFile ( ) [private]

Definition at line 172 of file quazip.cpp.

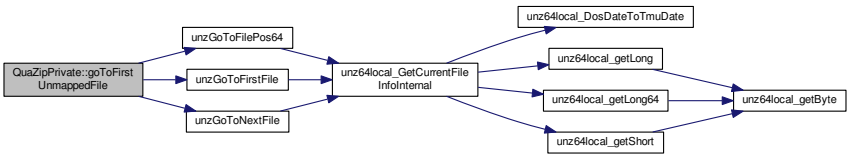
References hasCurrentFile_f, lastMappedDirectoryEntry, QuaZip::mdUnzip, mode, unz64_file_pos_s::pos_in_zip_directory, UNZ_END_OF_LIST_OF_FILE, UNZ_OK, unzGoToFilePos64(), unzGoToFirstFile(), unzGoToNextFile(), and zipError.

Referenced by QuaZip::setCurrentFile().

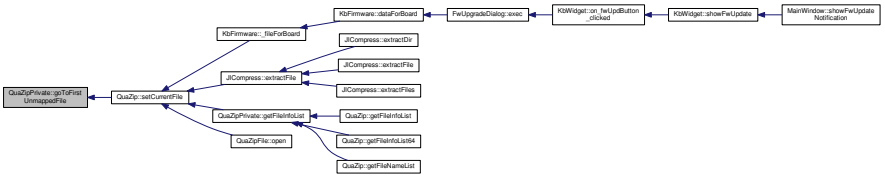
```
173 {
174     zipError = UNZ_OK;
175     if (mode != QuaZip::mdUnzip) {
176         qWarning("QuaZipPrivate::goToNextUnmappedFile(): ZIP is not open in mdUnzip mode");
177         return false;
178     }
179     // If not mapped anything, go to beginning
180     if (lastMappedDirectoryEntry.pos_in_zip_directory == 0) {
181         unzGoToFirstFile(unzFile_f);
182     } else {
183         // Goto the last one mapped, plus one
184         unzGoToFilePos64(unzFile_f, &lastMappedDirectoryEntry);
185         unzGoToNextFile(unzFile_f);
186     }
```

```
187     hasCurrentFile_f=zipError==UNZ_OK;
188     if(zipError==UNZ_END_OF_LIST_OF_FILE)
189       zipError=UNZ_OK;
190     return hasCurrentFile_f;
191 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.78.4 Friends And Related Function Documentation

12.78.4.1 friend class QuaZip [friend]

Definition at line 40 of file quazip.cpp.

12.78.5 Field Documentation

12.78.5.1 union { ... } [private]

12.78.5.2 bool QuaZipPrivate::autoClose [private]

Definition at line 71 of file quazip.cpp.

Referenced by QuaZip::isAutoClose(), QuaZip::open(), and QuaZip::setAutoClose().

12.78.5.3 QString QuaZipPrivate::comment [private]

Definition at line 53 of file quazip.cpp.

Referenced by QuaZip::close(), and QuaZip::setComment().

12.78.5.4 QTextCodec* QuaZipPrivate::commentCodec [private]

Definition at line 47 of file quazip.cpp.

Referenced by QuaZip::close(), QuaZip::getComment(), QuaZip::getCommentCodec(), QuaZip::getCurrentFileInfo(), and QuaZip::setCommentCodec().

12.78.5.5 bool QuaZipPrivate::dataDescriptorWritingEnabled [private]

Definition at line 67 of file quazip.cpp.

Referenced by QuaZip::isDataDescriptorWritingEnabled(), QuaZip::open(), and QuaZip::setDataDescriptorWritingEnabled().

12.78.5.6 QTextCodec * QuaZipPrivate::defaultFileNameCodec = NULL [static], [private]

Definition at line 140 of file quazip.cpp.

Referenced by getDefaultFileNameCodec(), and QuaZip::setDefaultFileNameCodec().

12.78.5.7 QHash<QString, unz64_file_pos> QuaZipPrivate::directoryCaseInsensitive [private]

Definition at line 138 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), clearDirectoryMap(), and QuaZip::setCurrentFile().

12.78.5.8 QHash<QString, unz64_file_pos> QuaZipPrivate::directoryCaseSensitive [private]

Definition at line 137 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), clearDirectoryMap(), and QuaZip::setCurrentFile().

12.78.5.9 QTextCodec* QuaZipPrivate::fileNameCodec [private]

Definition at line 45 of file quazip.cpp.

Referenced by QuaZip::getCurrentFileInfo(), QuaZip::getCurrentFileName(), QuaZip::getFileNameCodec(), and QuaZip::setFileNameCodec().

12.78.5.10 bool QuaZipPrivate::hasCurrentFile_f [private]

Definition at line 63 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), QuaZip::goToFirstFile(), goToFirstUnmappedFile(), QuaZip::goToNextFile(), QuaZip::hasCurrentFile(), and QuaZip::setCurrentFile().

12.78.5.11 QIODevice* QuaZipPrivate::ioDevice [private]

Definition at line 51 of file quazip.cpp.

Referenced by QuaZip::close(), QuaZip::getIoDevice(), QuaZip::open(), QuaZip::setIoDevice(), and QuaZip::setZipName().

12.78.5.12 unz64_file_pos QuaZipPrivate::lastMappedDirectoryEntry [private]

Definition at line 139 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), clearDirectoryMap(), goToFirstUnmappedFile(), and QuaZipPrivate().

12.78.5.13 QuaZip::Mode QuaZipPrivate::mode [private]

Definition at line 55 of file quazip.cpp.

Referenced by QuaZip::close(), QuaZip::getComment(), QuaZip::getCurrentFileInfo(), QuaZip::getCurrentFileName(), QuaZip::getEntriesCount(), getFileInfoList(), QuaZip::getMode(), QuaZip::goToFirstFile(), goToFirstUnmappedFile(), QuaZip::goToNextFile(), QuaZip::isOpen(), QuaZip::open(), and QuaZip::setCurrentFile().

12.78.5.14 QuaZip* QuaZipPrivate::q [private]

Definition at line 43 of file quazip.cpp.

Referenced by getFileInfoList().

12.78.5.15 bool QuaZipPrivate::zip64 [private]

Definition at line 69 of file quazip.cpp.

Referenced by QuaZip::isZip64Enabled(), and QuaZip::setZip64Enabled().

12.78.5.16 int QuaZipPrivate::zipError [private]

Definition at line 65 of file quazip.cpp.

Referenced by QuaZip::close(), QuaZip::getComment(), QuaZip::getCurrentFileInfo(), QuaZip::getCurrentFileName(), QuaZip::getEntriesCount(), getFileInfoList(), QuaZip::getZipError(), QuaZip::goToFirstFile(), goToFirstUnmappedFile(), QuaZip::goToNextFile(), QuaZip::open(), and QuaZip::setCurrentFile().

12.78.5.17 QString QuaZipPrivate::zipName [private]

Definition at line 49 of file quazip.cpp.

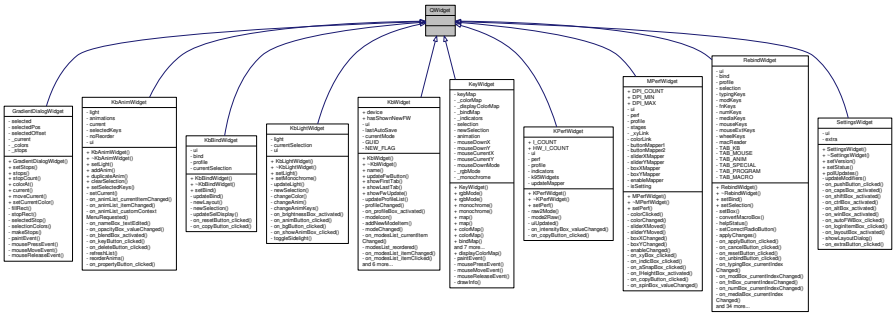
Referenced by QuaZip::close(), QuaZip::getIoDevice(), QuaZip::getZipName(), QuaZip::open(), QuaZip::setIoDevice(), and QuaZip::setZipName().

The documentation for this class was generated from the following file:

- src/ckb/quazip/quazip.cpp

12.79 QWidget Class Reference

Inheritance diagram for QWidget:



Collaboration diagram for QWidget:

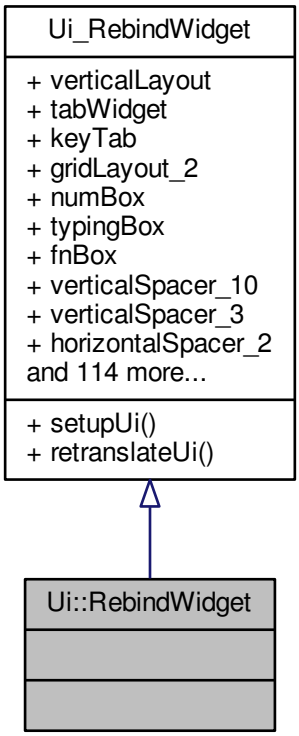
The documentation for this class was generated from the following file:

- src/ckb/keywidget.h

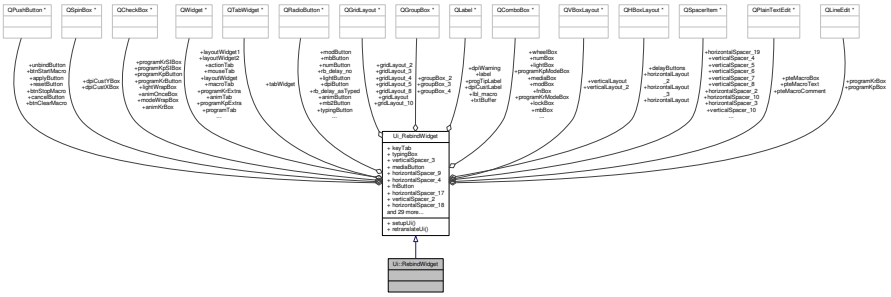
12.80 Ui::RebindWidget Class Reference

`#include <src/ckb/ui_rebindwidget.h>`

Inheritance diagram for Ui::RebindWidget:



Collaboration diagram for Ui::RebindWidget:



Public Member Functions

- void setupUi (QWidget *RebindWidget)
- void retranslateUi (QWidget *RebindWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QTabWidget * tabWidget
- QWidget * keyTab
- QGridLayout * gridLayout_2
- QComboBox * numBox
- QComboBox * typingBox
- QComboBox * fnBox
- QSpacerItem * verticalSpacer_10
- QSpacerItem * verticalSpacer_3
- QSpacerItem * horizontalSpacer_2
- QRadioButton * typingButton
- QComboBox * modBox
- QSpacerItem * horizontalSpacer_3
- QRadioButton * mediaButton
- QSpacerItem * verticalSpacer_4
- QSpacerItem * horizontalSpacer_9
- QSpacerItem * horizontalSpacer_4
- QRadioButton * modButton
- QRadioButton * fnButton
- QComboBox * mediaBox
- QRadioButton * numButton
- QSpacerItem * horizontalSpacer_17
- QWidget * mouseTab
- QGridLayout * gridLayout
- QRadioButton * dpiButton
- QSpacerItem * verticalSpacer_2
- QSpacerItem * verticalSpacer_8
- QSpacerItem * horizontalSpacer_18
- QSpacerItem * horizontalSpacer_20
- QComboBox * mb2Box
- QComboBox * dpiBox
- QSpinBox * dpiCustYBox
- QComboBox * mbBox

- QComboBox * wheelBox
- QRadioButton * mbButton
- QSpinBox * dpiCustXBox
- QLabel * dpiWarning
- QRadioButton * wheelButton
- QSpacerItem * horizontalSpacer_22
- QRadioButton * mb2Button
- QLabel * dpiCustLabel
- QSpacerItem * horizontalSpacer_19
- QSpacerItem * horizontalSpacer_21
- QWidget * animTab
- QGridLayout * gridLayout_5
- QSpacerItem * verticalSpacer_11
- QComboBox * animBox
- QSpacerItem * horizontalSpacer_24
- QSpacerItem * horizontalSpacer_25
- QCheckBox * animKrBox
- QCheckBox * animOnceBox
- QSpacerItem * horizontalSpacer_26
- QRadioButton * animButton
- QSpacerItem * horizontalSpacer_23
- QWidget * actionTab
- QGridLayout * gridLayout_3
- QSpacerItem * horizontalSpacer_5
- QRadioButton * modeButton
- QCheckBox * modeWrapBox
- QCheckBox * lightWrapBox
- QSpacerItem * verticalSpacer_7
- QSpacerItem * verticalSpacer_6
- QRadioButton * lockButton
- QSpacerItem * verticalSpacer_5
- QComboBox * lightBox
- QComboBox * lockBox
- QRadioButton * lightButton
- QComboBox * modeBox
- QSpacerItem * horizontalSpacer_8
- QSpacerItem * horizontalSpacer_10
- QSpacerItem * horizontalSpacer_7
- QSpacerItem * horizontalSpacer_6
- QWidget * programTab
- QGridLayout * gridLayout_4
- QSpacerItem * verticalSpacer
- QSpacerItem * horizontalSpacer
- QCheckBox * programKpButton
- QLineEdit * programKpBox
- QSpacerItem * horizontalSpacer_11
- QSpacerItem * horizontalSpacer_13
- QLineEdit * programKrBox
- QCheckBox * programKrButton
- QSpacerItem * horizontalSpacer_12
- QWidget * programKpExtra
- QHBoxLayout * horizontalLayout_2
- QCheckBox * programKpSIBox
- QComboBox * programKpModeBox
- QSpacerItem * horizontalSpacer_15

- QWidget * programKrExtra
- QHBoxLayout * horizontalLayout_3
- QCheckBox * programKrSIBox
- QComboBox * programKrModeBox
- QSpacerItem * horizontalSpacer_16
- QLabel * progTipLabel
- QWidget * macroTab
- QWidget * layoutWidget1
- QGridLayout * gridLayout_8
- QGroupBox * groupBox_2
- QPlainTextEdit * pteMacroBox
- QPlainTextEdit * pteMacroComment
- QPlainTextEdit * pteMacroText
- QWidget * layoutWidget
- QVBoxLayout * verticalLayout_2
- QGroupBox * groupBox_3
- QGroupBox * groupBox_4
- QWidget * layoutWidget_2
- QGridLayout * gridLayout_10
- QPushButton * btnStopMacro
- QPushButton * btnClearMacro
- QPushButton * btnStartMacro
- QLabel * label
- QLabel * lbl_macro
- QWidget * layoutWidget2
- QHBoxLayout * delayButtons
- QRadioButton * rb_delay_no
- QRadioButton * rb_delay_default
- QRadioButton * rb_delay_asTyped
- QLabel * txtBuffer
- QHBoxLayout * horizontalLayout
- QSpacerItem * horizontalSpacer_14
- QPushButton * unbindButton
- QPushButton * resetButton
- QPushButton * cancelButton
- QPushButton * applyButton

12.80.1 Detailed Description

Definition at line 1062 of file ui_rebindwidget.h.

12.80.2 Member Function Documentation

12.80.2.1 void Ui_RebindWidget::retranslateUi (QWidget * *RebindWidget*) [inline], [inherited]

Definition at line 901 of file ui_rebindwidget.h.

References Ui_RebindWidget::actionTab, Ui_RebindWidget::animBox, Ui_RebindWidget::animButton, Ui_RebindWidget::animKrBox, Ui_RebindWidget::animOnceBox, Ui_RebindWidget::animTab, Ui_RebindWidget::applyButton, Ui_RebindWidget::btnClearMacro, Ui_RebindWidget::btnStartMacro, Ui_RebindWidget::btnStopMacro, Ui_RebindWidget::cancelButton, Ui_RebindWidget::dpiBox, Ui_RebindWidget::dpiButton, Ui_RebindWidget::dpiCustLabel, Ui_RebindWidget::dpiWarning, Ui_RebindWidget::fnBox, Ui_RebindWidget::fnButton, Ui_RebindWidget::groupBox_2, Ui_RebindWidget::groupBox_3, Ui_RebindWidget::groupBox_4, Ui_RebindWidget::keyTab, Ui_RebindWidget::label, Ui_RebindWidget::lbl_macro, Ui_RebindWidget::lightBox, Ui_RebindWidget::lightButton,

Ui_RebindWidget::lightWrapBox, Ui_RebindWidget::lockBox, Ui_RebindWidget::lockButton, Ui_RebindWidget::macroTab, Ui_RebindWidget::mb2Box, Ui_RebindWidget::mb2Button, Ui_RebindWidget::mbBox, Ui_RebindWidget::mbButton, Ui_RebindWidget::mediaBox, Ui_RebindWidget::mediaButton, Ui_RebindWidget::modBox, Ui_RebindWidget::modButton, Ui_RebindWidget::modeBox, Ui_RebindWidget::modeButton, Ui_RebindWidget::modeWrapBox, Ui_RebindWidget::mouseTab, Ui_RebindWidget::numBox, Ui_RebindWidget::numButton, Ui_RebindWidget::programKpButton, Ui_RebindWidget::programKpModeBox, Ui_RebindWidget::programKpSIBox, Ui_RebindWidget::programKrButton, Ui_RebindWidget::programKrModeBox, Ui_RebindWidget::programKrSIBox, Ui_RebindWidget::programTab, Ui_RebindWidget::progTipLabel, Ui_RebindWidget::pteMacroComment, Ui_RebindWidget::pteMacroText, Ui_RebindWidget::rb_delay_asTyped, Ui_RebindWidget::rb_delay_default, Ui_RebindWidget::rb_delay_no, Ui_RebindWidget::resetButton, Ui_RebindWidget::tabWidget, Ui_RebindWidget::txtBuffer, Ui_RebindWidget::typingBox, Ui_RebindWidget::typingButton, Ui_RebindWidget::unbindButton, Ui_RebindWidget::wheelBox, and Ui_RebindWidget::wheelButton.

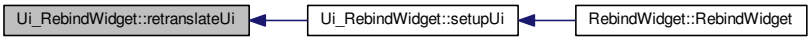
Referenced by Ui_RebindWidget::setupUi().

```
902     {
903         numBox->clear();
904         numBox->insertItems(0, QStringList()
905          << QString()
906         );
907         typingBox->clear();
908         typingBox->insertItems(0, QStringList()
909          << QString()
910         );
911         fnBox->clear();
912         fnBox->insertItems(0, QStringList()
913          << QString()
914         );
915         typingButton->setText(QApplication::translate("RebindWidget", "Typing:", Q_NULLPTR));
916         modBox->clear();
917         modBox->insertItems(0, QStringList()
918          << QString()
919         );
920         mediaButton->setText(QApplication::translate("RebindWidget", "Media:", Q_NULLPTR));
921         modButton->setText(QApplication::translate("RebindWidget", "Modifier:", Q_NULLPTR));
922         fnButton->setText(QApplication::translate("RebindWidget", "Miscellaneous:", Q_NULLPTR));
923         mediaBox->clear();
924         mediaBox->insertItems(0, QStringList()
925          << QString()
926         );
927         numButton->setText(QApplication::translate("RebindWidget", "Number pad:", Q_NULLPTR));
928         tabWidget->setTabText(tabWidget->indexOf(keyTab), QApplication::translate("
    RebindWidget", "Keyboard", Q_NULLPTR));
929         dpiButton->setText(QApplication::translate("RebindWidget", "Change DPI:", Q_NULLPTR));
930         mb2Box->clear();
931         mb2Box->insertItems(0, QStringList()
932          << QString()
933          << QApplication::translate("RebindWidget", "4 (Back)", Q_NULLPTR)
934          << QApplication::translate("RebindWidget", "5 (Forward)", Q_NULLPTR)
935          << QApplication::translate("RebindWidget", "6", Q_NULLPTR)
936          << QApplication::translate("RebindWidget", "7", Q_NULLPTR)
937          << QApplication::translate("RebindWidget", "8", Q_NULLPTR)
938         );
939         dpiBox->clear();
940         dpiBox->insertItems(0, QStringList()
941          << QString()
942          << QApplication::translate("RebindWidget", "(Cycle Up)", Q_NULLPTR)
943          << QApplication::translate("RebindWidget", "(Cycle Down)", Q_NULLPTR)
944          << QApplication::translate("RebindWidget", "(Up)", Q_NULLPTR)
945          << QApplication::translate("RebindWidget", "(Down)", Q_NULLPTR)
946          << QApplication::translate("RebindWidget", "Sniper", Q_NULLPTR)
947          << QApplication::translate("RebindWidget", "1:", Q_NULLPTR)
948          << QApplication::translate("RebindWidget", "2:", Q_NULLPTR)
949          << QApplication::translate("RebindWidget", "3:", Q_NULLPTR)
950          << QApplication::translate("RebindWidget", "4:", Q_NULLPTR)
951          << QApplication::translate("RebindWidget", "5:", Q_NULLPTR)
952          << QApplication::translate("RebindWidget", "Custom:", Q_NULLPTR)
953         );
954         mbBox->clear();
955         mbBox->insertItems(0, QStringList()
956          << QString()
957          << QApplication::translate("RebindWidget", "Left", Q_NULLPTR)
958          << QApplication::translate("RebindWidget", "Right", Q_NULLPTR)
959          << QApplication::translate("RebindWidget", "Middle", Q_NULLPTR)
960         );
961         wheelBox->clear();
962         wheelBox->insertItems(0, QStringList()
963          << QString()
964          << QApplication::translate("RebindWidget", "Up", Q_NULLPTR)
965          << QApplication::translate("RebindWidget", "Down", Q_NULLPTR)
966         );
```

```
967         mbButton->setText(QApplication::translate("RebindWidget", "Standard button:", Q_NULLPTR));
968         dpiWarning->setText(QApplication::translate("RebindWidget", "DPI is only available for
     mice.", Q_NULLPTR));
969         wheelButton->setText(QApplication::translate("RebindWidget", "Wheel:", Q_NULLPTR));
970         mb2Button->setText(QApplication::translate("RebindWidget", "Special button:", Q_NULLPTR));
971         dpiCustLabel->setText(QApplication::translate("RebindWidget", "x", Q_NULLPTR));
972         tabWidget->setTabText(tabWidget->indexOf(mouseTab),
    QApplication::translate("RebindWidget", "Mouse", Q_NULLPTR));
973         animBox->clear();
974         animBox->insertItems(0, QStringList()
975          << QString()
976         );
977         animKrBox->setText(QApplication::translate("RebindWidget", "Stop on key release",
    Q_NULLPTR));
978         animOnceBox->setText(QApplication::translate("RebindWidget", "Restart every time",
    Q_NULLPTR));
979         animButton->setText(QApplication::translate("RebindWidget", "Start animation on keypress:
    ", Q_NULLPTR));
980         tabWidget->setTabText(tabWidget->indexOf(animTab), QApplication::translate
    ("RebindWidget", "Animation", Q_NULLPTR));
981         modeButton->setText(QApplication::translate("RebindWidget", "Switch to mode:", Q_NULLPTR)
    );
982         modeWrapBox->setText(QApplication::translate("RebindWidget", "Wrap-around", Q_NULLPTR));
983         lightWrapBox->setText(QApplication::translate("RebindWidget", "Wrap-around", Q_NULLPTR)
    );
984         lockButton->setText(QApplication::translate("RebindWidget", "Windows lock:", Q_NULLPTR));
985         lightBox->clear();
986         lightBox->insertItems(0, QStringList()
987          << QString()
988          << QApplication::translate("RebindWidget", "Up", Q_NULLPTR)
989          << QApplication::translate("RebindWidget", "Down", Q_NULLPTR)
990         );
991         lockBox->clear();
992         lockBox->insertItems(0, QStringList()
993          << QString()
994          << QApplication::translate("RebindWidget", "Toggle", Q_NULLPTR)
995          << QApplication::translate("RebindWidget", "On", Q_NULLPTR)
996          << QApplication::translate("RebindWidget", "Off", Q_NULLPTR)
997         );
998         lightButton->setText(QApplication::translate("RebindWidget", "Brightness:", Q_NULLPTR));
999         modeBox->clear();
1000         modeBox->insertItems(0, QStringList()
1001          << QString()
1002          << QApplication::translate("RebindWidget", "(Previous)", Q_NULLPTR)
1003          << QApplication::translate("RebindWidget", "(Next)", Q_NULLPTR)
1004         );
1005         tabWidget->setTabText(tabWidget->indexOf(actionTab),
    QApplication::translate("RebindWidget", "Special", Q_NULLPTR));
1006         programKpButton->setText(QApplication::translate("RebindWidget", "Launch program on
     key press:", Q_NULLPTR));
1007         programKrButton->setText(QApplication::translate("RebindWidget", "Launch program on
     key release:", Q_NULLPTR));
1008         programKpSIBox->setText(QApplication::translate("RebindWidget", "Single instance",
    Q_NULLPTR));
1009         programKpModeBox->clear();
1010         programKpModeBox->insertItems(0, QStringList()
1011          << QApplication::translate("RebindWidget", "Run indefinitely", Q_NULLPTR)
1012          << QApplication::translate("RebindWidget", "Stop on release", Q_NULLPTR)
1013          << QApplication::translate("RebindWidget", "Stop on re-press", Q_NULLPTR)
1014         );
1015         programKrSIBox->setText(QApplication::translate("RebindWidget", "Single instance",
    Q_NULLPTR));
1016         programKrModeBox->clear();
1017         programKrModeBox->insertItems(0, QStringList()
1018          << QApplication::translate("RebindWidget", "Run indefinitely", Q_NULLPTR)
1019          << QApplication::translate("RebindWidget", "Stop on re-press", Q_NULLPTR)
1020         );
1021         progTipLabel->setText(QApplication::translate("RebindWidget", "Tip", Q_NULLPTR));
1022         tabWidget->setTabText(tabWidget->indexOf(programTab),
    QApplication::translate("RebindWidget", "Program", Q_NULLPTR));
1023         groupBox_2->setTitle(QApplication::translate("RebindWidget", "Macro Key Actions",
    Q_NULLPTR));
1024 #ifndef QT_NO_TOOLTIP
1025         pteMacroComment->setToolTip(QApplication::translate("RebindWidget", "What is this
     macro for?", Q_NULLPTR));
1026 #endif // QT_NO_TOOLTIP
1027         pteMacroComment->setDocumentTitle(QString());
1028         pteMacroComment->setPlainText(QString());
1029         pteMacroText->setDocumentTitle(QString());
1030         pteMacroText->setPlainText(QString());
1031         groupBox_3->setTitle(QApplication::translate("RebindWidget", "Macro Comment", Q_NULLPTR))
    ;
1032         groupBox_4->setTitle(QApplication::translate("RebindWidget", "Macro Text", Q_NULLPTR));
1033         btnStopMacro->setText(QApplication::translate("RebindWidget", "Stop", Q_NULLPTR));
1034         btnClearMacro->setText(QApplication::translate("RebindWidget", "Clear", Q_NULLPTR));
1035         btnStartMacro->setText(QApplication::translate("RebindWidget", "Start", Q_NULLPTR));
1036         label->setText(QApplication::translate("RebindWidget", "Record macro ", Q_NULLPTR));
```

```
1037         lbl_macro->setText(QApplication::translate("RebindWidget", "Comment label for help",
     Q_NULLPTR));
1038 #ifndef QT_NO_WHATSTHIS
1039         rb_delay_no->setWhatsThis(QApplication::translate("RebindWidget", "configure delays
     between keystrokes: Disable delay", Q_NULLPTR));
1040 #endif // QT_NO_WHATSTHIS
1041         rb_delay_no->setText(QApplication::translate("RebindWidget", "No delay", Q_NULLPTR));
1042 #ifndef QT_NO_WHATSTHIS
1043         rb_delay_default->setWhatsThis(QApplication::translate("RebindWidget", "configure
     delays between keystrokes: Set delay to default values: 20us up to 15 chars, 200us above", Q_NULLPTR));
1044 #endif // QT_NO_WHATSTHIS
1045         rb_delay_default->setText(QApplication::translate("RebindWidget", "default",
     Q_NULLPTR));
1046 #ifndef QT_NO_WHATSTHIS
1047         rb_delay_asTyped->setWhatsThis(QApplication::translate("RebindWidget", "configure
     delays between keystrokes: Remember delays as they had been typed while the macro definition process",
     Q_NULLPTR));
1048 #endif // QT_NO_WHATSTHIS
1049         rb_delay_asTyped->setText(QApplication::translate("RebindWidget", "as typed",
     Q_NULLPTR));
1050         txtBuffer->setText(QString());
1051         tabWidget->setTabText(tabWidget->indexOf(macroTab),
     QApplication::translate("RebindWidget", "Macro", Q_NULLPTR));
1052         unbindButton->setText(QApplication::translate("RebindWidget", "Unbind", Q_NULLPTR));
1053         resetButton->setText(QApplication::translate("RebindWidget", "Reset to Default",
     Q_NULLPTR));
1054         cancelButton->setText(QApplication::translate("RebindWidget", "Cancel", Q_NULLPTR));
1055         applyButton->setText(QApplication::translate("RebindWidget", "Apply", Q_NULLPTR));
1056         Q_UNUSED(RebindWidget);
1057     } // retranslateUi
```

Here is the caller graph for this function:



12.80.2.2 void Ui_RebindWidget::setupUi (QWidget * *RebindWidget*) [inline],[inherited]

Definition at line 163 of file ui_rebindwidget.h.

References Ui_RebindWidget::actionTab, Ui_RebindWidget::animBox, Ui_RebindWidget::animButton, Ui_RebindWidget::animKrBox, Ui_RebindWidget::animOnceBox, Ui_RebindWidget::animTab, Ui_RebindWidget::applyButton, Ui_RebindWidget::btnClearMacro, Ui_RebindWidget::btnStartMacro, Ui_RebindWidget::btnStopMacro, Ui_RebindWidget::cancelButton, Ui_RebindWidget::delayButtons, Ui_RebindWidget::dpiBox, Ui_RebindWidget::dpiButton, Ui_RebindWidget::dpiCustLabel, Ui_RebindWidget::dpiCustXBox, Ui_RebindWidget::dpiCustYBox, Ui_RebindWidget::dpiWarning, Ui_RebindWidget::fnBox, Ui_RebindWidget::fnButton, Ui_RebindWidget::gridLayout, Ui_RebindWidget::gridLayout_10, Ui_RebindWidget::gridLayout_2, Ui_RebindWidget::gridLayout_3, Ui_RebindWidget::gridLayout_4, Ui_RebindWidget::gridLayout_5, Ui_RebindWidget::gridLayout_8, Ui_RebindWidget::groupBox_2, Ui_RebindWidget::groupBox_3, Ui_RebindWidget::groupBox_4, Ui_RebindWidget::horizontalLayout, Ui_RebindWidget::horizontalLayout_2, Ui_RebindWidget::horizontalLayout_3, Ui_RebindWidget::horizontalSpacer, Ui_RebindWidget::horizontalSpacer_10, Ui_RebindWidget::horizontalSpacer_11, Ui_RebindWidget::horizontalSpacer_12, Ui_RebindWidget::horizontalSpacer_13, Ui_RebindWidget::horizontalSpacer_14, Ui_RebindWidget::horizontalSpacer_15, Ui_RebindWidget::horizontalSpacer_16, Ui_RebindWidget::horizontalSpacer_17, Ui_RebindWidget::horizontalSpacer_18, Ui_RebindWidget::horizontalSpacer_19, Ui_RebindWidget::horizontalSpacer_2, Ui_RebindWidget::horizontalSpacer_20, Ui_RebindWidget::horizontalSpacer_21, Ui_RebindWidget::horizontalSpacer_22, Ui_RebindWidget::horizontalSpacer_23, Ui_RebindWidget::horizontalSpacer_24, Ui_RebindWidget::horizontalSpacer_25, Ui_RebindWidget::horizontalSpacer_26, Ui_RebindWidget::horizontalSpacer_3, Ui_RebindWidget::horizontalSpacer_4, Ui_RebindWidget::horizontalSpacer_5, Ui_RebindWidget::horizontalSpacer_6, Ui_RebindWidget::horizontalSpacer_7, Ui_RebindWidget::horizontalSpacer_8, Ui_RebindWidget::horizontalSpacer_9, Ui_RebindWidget::keyTab, Ui_RebindWidget::label, Ui_RebindWidget::layoutWidget, Ui_RebindWidget::layoutWidget1, Ui_RebindWidget::layoutWidget2, Ui_RebindWidget::layoutWidget_2, Ui_RebindWidget::lbl_macro, Ui_RebindWidget::lightBox, Ui_RebindWidget::lightButton, Ui_RebindWidget::lightWrapBox, Ui_RebindWidget::lockBox, Ui_RebindWidget::lockButton, Ui_RebindWidget::macroTab, Ui_RebindWidget-

::mb2Box, Ui_RebindWidget::mb2Button, Ui_RebindWidget::mbBox, Ui_RebindWidget::mbButton, Ui_RebindWidget::mediaBox, Ui_RebindWidget::mediaButton, Ui_RebindWidget::modBox, Ui_RebindWidget::modButton, Ui_RebindWidget::modeBox, Ui_RebindWidget::modeButton, Ui_RebindWidget::modeWrapBox, Ui_RebindWidget::mouseTab, Ui_RebindWidget::numBox, Ui_RebindWidget::numButton, Ui_RebindWidget::programKpBox, Ui_RebindWidget::programKpButton, Ui_RebindWidget::programKpExtra, Ui_RebindWidget::programKpModeBox, Ui_RebindWidget::programKpSIBox, Ui_RebindWidget::programKrBox, Ui_RebindWidget::programKrButton, Ui_RebindWidget::programKrExtra, Ui_RebindWidget::programKrModeBox, Ui_RebindWidget::programKrSIBox, Ui_RebindWidget::programTab, Ui_RebindWidget::progTipLabel, Ui_RebindWidget::pteMacroBox, Ui_RebindWidget::pteMacroComment, Ui_RebindWidget::pteMacroText, Ui_RebindWidget::rb_delay_asTyped, Ui_RebindWidget::rb_delay_default, Ui_RebindWidget::rb_delay_no, Ui_RebindWidget::resetButton, Ui_RebindWidget::retranslateUi(), Ui_RebindWidget::tabWidget, Ui_RebindWidget::txtBuffer, Ui_RebindWidget::typingBox, Ui_RebindWidget::typingButton, Ui_RebindWidget::unbindButton, Ui_RebindWidget::verticalLayout, Ui_RebindWidget::verticalLayout_2, Ui_RebindWidget::verticalSpacer, Ui_RebindWidget::verticalSpacer_10, Ui_RebindWidget::verticalSpacer_11, Ui_RebindWidget::verticalSpacer_2, Ui_RebindWidget::verticalSpacer_3, Ui_RebindWidget::verticalSpacer_4, Ui_RebindWidget::verticalSpacer_5, Ui_RebindWidget::verticalSpacer_6, Ui_RebindWidget::verticalSpacer_7, Ui_RebindWidget::verticalSpacer_8, Ui_RebindWidget::wheelBox, and Ui_RebindWidget::wheelButton.

Referenced by RebindWidget::RebindWidget().

```
164     {
165         if (RebindWidget->objectName().isEmpty())
166             RebindWidget->setObjectName(QStringLiteral("RebindWidget"));
167         RebindWidget->resize(687, 342);
168         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
169         sizePolicy.setHorizontalStretch(0);
170         sizePolicy.setVerticalStretch(0);
171         sizePolicy.setHeightForWidth(RebindWidget->sizePolicy().hasHeightForWidth());
172         RebindWidget->setSizePolicy(sizePolicy);
173         RebindWidget->setStyleSheet(QStringLiteral("font-size: 12px ;"));
174         verticalLayout = new QVBoxLayout(RebindWidget);
175         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
176         tabWidget = new QTabWidget(RebindWidget);
177         tabWidget->setObjectName(QStringLiteral("tabWidget"));
178         tabWidget->setStyleSheet(QStringLiteral(""));
179         tabWidget->setUsesScrollButtons(false);
180         tabWidget->setDocumentMode(false);
181         keyTab = new QWidget();
182         keyTab->setObjectName(QStringLiteral("keyTab"));
183         sizePolicy.setHeightForWidth(keyTab->sizePolicy().hasHeightForWidth());
184         keyTab->setSizePolicy(sizePolicy);
185         gridLayout_2 = new QGridLayout(keyTab);
186         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
187         gridLayout_2->setContentsMargins(-1, -1, -1, 0);
188         numBox = new QComboBox(keyTab);
189         numBox->setObjectName(QStringLiteral("numBox"));
190         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Fixed);
191         sizePolicy1.setHorizontalStretch(0);
192         sizePolicy1.setVerticalStretch(0);
193         sizePolicy1.setHeightForWidth(numBox->sizePolicy().hasHeightForWidth());
194         numBox->setSizePolicy(sizePolicy1);
195 
196         gridLayout_2->addWidget(numBox, 6, 4, 1, 1);
197 
198         typingBox = new QComboBox(keyTab);
199         typingBox->setObjectName(QStringLiteral("typingBox"));
200         sizePolicy1.setHeightForWidth(typingBox->sizePolicy().hasHeightForWidth());
201         typingBox->setSizePolicy(sizePolicy1);
202 
203         gridLayout_2->addWidget(typingBox, 1, 1, 1, 1);
204 
205         fnBox = new QComboBox(keyTab);
206         fnBox->setObjectName(QStringLiteral("fnBox"));
207         sizePolicy1.setHeightForWidth(fnBox->sizePolicy().hasHeightForWidth());
208         fnBox->setSizePolicy(sizePolicy1);
209 
210         gridLayout_2->addWidget(fnBox, 6, 1, 1, 1);
211 
212         verticalSpacer_10 = new QSpacerItem(0, 5, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
213 
214         gridLayout_2->addItem(verticalSpacer_10, 9, 1, 1, 1);
215 
216         verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
217 
218         gridLayout_2->addItem(verticalSpacer_3, 15, 2, 1, 1);
219 
220         horizontalSpacer_2 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
```

```
221
222         gridLayout_2->addItem(horizontalSpacer_2, 1, 3, 1, 1);
223
224         typingButton = new QRadioButton(keyTab);
225         typingButton->setObjectName(QStringLiteral("typingButton"));
226         typingButton->setAutoExclusive(false);
227
228         gridLayout_2->addWidget(typingButton, 0, 0, 1, 2);
229
230         modBox = new QComboBox(keyTab);
231         modBox->setObjectName(QStringLiteral("modBox"));
232         sizePolicy1.setHeightForWidth(modBox->sizePolicy().hasHeightForWidth());
233         modBox->setSizePolicy(sizePolicy1);
234
235         gridLayout_2->addWidget(modBox, 1, 4, 1, 1);
236
237         horizontalSpacer_3 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
238
239         gridLayout_2->addItem(horizontalSpacer_3, 1, 0, 1, 1);
240
241         mediaButton = new QRadioButton(keyTab);
242         mediaButton->setObjectName(QStringLiteral("mediaButton"));
243         mediaButton->setAutoExclusive(false);
244
245         gridLayout_2->addWidget(mediaButton, 10, 0, 1, 2);
246
247         verticalSpacer_4 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
248
249         gridLayout_2->addItem(verticalSpacer_4, 2, 1, 1, 1);
250
251         horizontalSpacer_9 = new QSpacerItem(0, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
252
253         gridLayout_2->addItem(horizontalSpacer_9, 1, 5, 1, 2);
254
255         horizontalSpacer_4 = new QSpacerItem(0, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
256
257         gridLayout_2->addItem(horizontalSpacer_4, 1, 2, 1, 1);
258
259         modButton = new QRadioButton(keyTab);
260         modButton->setObjectName(QStringLiteral("modButton"));
261
262         gridLayout_2->addWidget(modButton, 0, 3, 1, 2);
263
264         fnButton = new QRadioButton(keyTab);
265         fnButton->setObjectName(QStringLiteral("fnButton"));
266         fnButton->setAutoExclusive(false);
267
268         gridLayout_2->addWidget(fnButton, 5, 0, 1, 2);
269
270         mediaBox = new QComboBox(keyTab);
271         mediaBox->setObjectName(QStringLiteral("mediaBox"));
272         sizePolicy1.setHeightForWidth(mediaBox->sizePolicy().hasHeightForWidth());
273         mediaBox->setSizePolicy(sizePolicy1);
274
275         gridLayout_2->addWidget(mediaBox, 11, 1, 1, 1);
276
277         numButton = new QRadioButton(keyTab);
278         numButton->setObjectName(QStringLiteral("numButton"));
279         numButton->setAutoExclusive(false);
280
281         gridLayout_2->addWidget(numButton, 5, 3, 1, 2);
282
283         horizontalSpacer_17 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
284
285         gridLayout_2->addItem(horizontalSpacer_17, 1, 7, 1, 1);
286
287         tabWidget->addTab(keyTab, QString());
288         mouseTab = new QWidget();
289         mouseTab->setObjectName(QStringLiteral("mouseTab"));
290         gridLayout = new QGridLayout(mouseTab);
291         gridLayout->setObjectName(QStringLiteral("gridLayout"));
292         dpiButton = new QRadioButton(mouseTab);
293         dpiButton->setObjectName(QStringLiteral("dpiButton"));
294         dpiButton->setAutoExclusive(false);
295
296         gridLayout->addWidget(dpiButton, 3, 3, 1, 5);
297
298         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
299
300         gridLayout->addItem(verticalSpacer_2, 5, 1, 1, 1);
301
```

```
302         verticalSpacer_8 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
303
304         gridLayout->addItem(verticalSpacer_8, 2, 1, 1, 1);
305
306         horizontalSpacer_18 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
307
308         gridLayout->addItem(horizontalSpacer_18, 1, 0, 1, 1);
309
310         horizontalSpacer_20 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
311
312         gridLayout->addItem(horizontalSpacer_20, 1, 3, 1, 1);
313
314         mb2Box = new QComboBox(mouseTab);
315         mb2Box->setObjectName(QStringLiteral("mb2Box"));
316         sizePolicy1.setHeightForWidth(mb2Box->sizePolicy().hasHeightForWidth());
317         mb2Box->setSizePolicy(sizePolicy1);
318
319         gridLayout->addWidget(mb2Box, 1, 4, 1, 4);
320
321         dpiBox = new QComboBox(mouseTab);
322         dpiBox->setObjectName(QStringLiteral("dpiBox"));
323
324         gridLayout->addWidget(dpiBox, 4, 4, 1, 1);
325
326         dpiCustYBox = new QSpinBox(mouseTab);
327         dpiCustYBox->setObjectName(QStringLiteral("dpiCustYBox"));
328         dpiCustYBox->setMinimum(400);
329         dpiCustYBox->setMaximum(8500);
330         dpiCustYBox->setSingleStep(50);
331
332         gridLayout->addWidget(dpiCustYBox, 4, 7, 1, 1);
333
334         mbBox = new QComboBox(mouseTab);
335         mbBox->setObjectName(QStringLiteral("mbBox"));
336         sizePolicy1.setHeightForWidth(mbBox->sizePolicy().hasHeightForWidth());
337         mbBox->setSizePolicy(sizePolicy1);
338
339         gridLayout->addWidget(mbBox, 1, 1, 1, 1);
340
341         wheelBox = new QComboBox(mouseTab);
342         wheelBox->setObjectName(QStringLiteral("wheelBox"));
343
344         gridLayout->addWidget(wheelBox, 4, 1, 1, 1);
345
346         mbButton = new QRadioButton(mouseTab);
347         mbButton->setObjectName(QStringLiteral("mbButton"));
348         mbButton->setAutoExclusive(false);
349
350         gridLayout->addWidget(mbButton, 0, 0, 1, 2);
351
352         dpiCustXBox = new QSpinBox(mouseTab);
353         dpiCustXBox->setObjectName(QStringLiteral("dpiCustXBox"));
354         dpiCustXBox->setMinimum(400);
355         dpiCustXBox->setMaximum(8500);
356         dpiCustXBox->setSingleStep(50);
357
358         gridLayout->addWidget(dpiCustXBox, 4, 5, 1, 1);
359
360         dpiWarning = new QLabel(mouseTab);
361         dpiWarning->setObjectName(QStringLiteral("dpiWarning"));
362         QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Fixed);
363         sizePolicy2.setHorizontalStretch(0);
364         sizePolicy2.setVerticalStretch(0);
365         sizePolicy2.setHeightForWidth(dpiWarning->sizePolicy().hasHeightForWidth());
366         dpiWarning->setSizePolicy(sizePolicy2);
367
368         gridLayout->addWidget(dpiWarning, 6, 0, 1, 10);
369
370         wheelButton = new QRadioButton(mouseTab);
371         wheelButton->setObjectName(QStringLiteral("wheelButton"));
372
373         gridLayout->addWidget(wheelButton, 3, 0, 1, 2);
374
375         horizontalSpacer_22 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
376
377         gridLayout->addItem(horizontalSpacer_22, 1, 8, 1, 1);
378
379         mb2Button = new QRadioButton(mouseTab);
380         mb2Button->setObjectName(QStringLiteral("mb2Button"));
381         mb2Button->setAutoExclusive(false);
382
383         gridLayout->addWidget(mb2Button, 0, 3, 1, 5);
384
```

```
385         dpiCustLabel = new QLabel(mouseTab);
386         dpiCustLabel->setObjectName(QStringLiteral("dpiCustLabel"));
387 
388         gridLayout->addWidget(dpiCustLabel, 4, 6, 1, 1);
389 
390         horizontalSpacer_19 = new QSpacerItem(0, 0, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
391 
392         gridLayout->addItem(horizontalSpacer_19, 1, 2, 1, 1);
393 
394         horizontalSpacer_21 = new QSpacerItem(0, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
395 
396         gridLayout->addItem(horizontalSpacer_21, 1, 9, 1, 1);
397 
398         tabWidget->addTab(mouseTab, QString());
399         animTab = new QWidget();
400         animTab->setObjectName(QStringLiteral("animTab"));
401         gridLayout_5 = new QGridLayout(animTab);
402         gridLayout_5->setObjectName(QStringLiteral("gridLayout_5"));
403         verticalSpacer_11 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
404 
405         gridLayout_5->addItem(verticalSpacer_11, 6, 2, 1, 1);
406 
407         animBox = new QComboBox(animTab);
408         animBox->setObjectName(QStringLiteral("animBox"));
409         sizePolicy1.setHeightForWidth(animBox->sizePolicy().hasHeightForWidth());
410         animBox->setSizePolicy(sizePolicy1);
411 
412         gridLayout_5->addWidget(animBox, 4, 2, 1, 1);
413 
414         horizontalSpacer_24 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
415 
416         gridLayout_5->addItem(horizontalSpacer_24, 4, 3, 1, 1);
417 
418         horizontalSpacer_25 = new QSpacerItem(438, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
419 
420         gridLayout_5->addItem(horizontalSpacer_25, 4, 7, 1, 1);
421 
422         animKrBox = new QCheckBox(animTab);
423         animKrBox->setObjectName(QStringLiteral("animKrBox"));
424 
425         gridLayout_5->addWidget(animKrBox, 4, 6, 1, 1);
426 
427         animOnceBox = new QCheckBox(animTab);
428         animOnceBox->setObjectName(QStringLiteral("animOnceBox"));
429 
430         gridLayout_5->addWidget(animOnceBox, 4, 4, 1, 1);
431 
432         horizontalSpacer_26 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
433 
434         gridLayout_5->addItem(horizontalSpacer_26, 4, 5, 1, 1);
435 
436         animButton = new QRadioButton(animTab);
437         animButton->setObjectName(QStringLiteral("animButton"));
438         sizePolicy1.setHeightForWidth(animButton->sizePolicy().hasHeightForWidth());
439         animButton->setSizePolicy(sizePolicy1);
440 
441         gridLayout_5->addWidget(animButton, 0, 1, 1, 2);
442 
443         horizontalSpacer_23 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
444 
445         gridLayout_5->addItem(horizontalSpacer_23, 4, 1, 1, 1);
446 
447         tabWidget->addTab(animTab, QString());
448         actionTab = new QWidget();
449         actionTab->setObjectName(QStringLiteral("actionTab"));
450         sizePolicy.setHeightForWidth(actionTab->sizePolicy().hasHeightForWidth());
451         actionTab->setSizePolicy(sizePolicy);
452         gridLayout_3 = new QGridLayout(actionTab);
453         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
454         gridLayout_3->setContentsMargins(-1, -1, -1, 0);
455         horizontalSpacer_5 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
456 
457         gridLayout_3->addItem(horizontalSpacer_5, 1, 0, 1, 1);
458 
459         modeButton = new QRadioButton(actionTab);
460         modeButton->setObjectName(QStringLiteral("modeButton"));
461         modeButton->setAutoExclusive(false);
462 
463         gridLayout_3->addWidget(modeButton, 0, 0, 1, 2);
```

```
464
465         modeWrapBox = new QCheckBox(actionTab);
466         modeWrapBox->setObjectName(QStringLiteral("modeWrapBox"));
467         modeWrapBox->setChecked(true);
468
469         gridLayout_3->addWidget(modeWrapBox, 1, 3, 1, 1);
470
471         lightWrapBox = new QCheckBox(actionTab);
472         lightWrapBox->setObjectName(QStringLiteral("lightWrapBox"));
473         lightWrapBox->setChecked(true);
474
475         gridLayout_3->addWidget(lightWrapBox, 4, 3, 1, 1);
476
477         verticalSpacer_7 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
478
479         gridLayout_3->addItem(verticalSpacer_7, 2, 1, 1, 1);
480
481         verticalSpacer_6 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
482
483         gridLayout_3->addItem(verticalSpacer_6, 5, 1, 1, 1);
484
485         lockButton = new QRadioButton(actionTab);
486         lockButton->setObjectName(QStringLiteral("lockButton"));
487         lockButton->setAutoExclusive(false);
488
489         gridLayout_3->addWidget(lockButton, 6, 0, 1, 2);
490
491         verticalSpacer_5 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
492
493         gridLayout_3->addItem(verticalSpacer_5, 8, 1, 1, 1);
494
495         lightBox = new QComboBox(actionTab);
496         lightBox->setObjectName(QStringLiteral("lightBox"));
497         sizePolicy1.setHeightForWidth(lightBox->sizePolicy().hasHeightForWidth());
498         lightBox->setSizePolicy(sizePolicy1);
499
500         gridLayout_3->addWidget(lightBox, 4, 1, 1, 1);
501
502         lockBox = new QComboBox(actionTab);
503         lockBox->setObjectName(QStringLiteral("lockBox"));
504         sizePolicy1.setHeightForWidth(lockBox->sizePolicy().hasHeightForWidth());
505         lockBox->setSizePolicy(sizePolicy1);
506
507         gridLayout_3->addWidget(lockBox, 7, 1, 1, 1);
508
509         lightButton = new QRadioButton(actionTab);
510         lightButton->setObjectName(QStringLiteral("lightButton"));
511         lightButton->setAutoExclusive(false);
512
513         gridLayout_3->addWidget(lightButton, 3, 0, 1, 2);
514
515         modeBox = new QComboBox(actionTab);
516         modeBox->setObjectName(QStringLiteral("modeBox"));
517         sizePolicy1.setHeightForWidth(modeBox->sizePolicy().hasHeightForWidth());
518         modeBox->setSizePolicy(sizePolicy1);
519
520         gridLayout_3->addWidget(modeBox, 1, 1, 1, 1);
521
522         horizontalSpacer_8 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
523
524         gridLayout_3->addItem(horizontalSpacer_8, 1, 2, 1, 1);
525
526         horizontalSpacer_10 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
527
528         gridLayout_3->addItem(horizontalSpacer_10, 7, 3, 1, 1);
529
530         horizontalSpacer_7 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
531
532         gridLayout_3->addItem(horizontalSpacer_7, 1, 4, 1, 1);
533
534         horizontalSpacer_6 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
535
536         gridLayout_3->addItem(horizontalSpacer_6, 1, 5, 1, 1);
537
538         tabWidget->addTab(actionTab, QString());
539         programTab = new QWidget();
540         programTab->setObjectName(QStringLiteral("programTab"));
541         gridLayout_4 = new QGridLayout(programTab);
542         gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
543         verticalSpacer = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed);
```

```
544
545         gridLayout_4->addItem(verticalSpacer, 2, 1, 1, 1);
546
547         horizontalSpacer = new QSpacerItem(5, 20, QSizePolicy::Fixed, QSizePolicy::Minimum)
    ;
548
549         gridLayout_4->addItem(horizontalSpacer, 1, 0, 1, 1);
550
551         programKpButton = new QCheckBox(programTab);
552         programKpButton->setObjectName(QStringLiteral("programKpButton"));
553
554         gridLayout_4->addWidget(programKpButton, 0, 0, 1, 2);
555
556         programKpBox = new QLineEdit(programTab);
557         programKpBox->setObjectName(QStringLiteral("programKpBox"));
558
559         gridLayout_4->addWidget(programKpBox, 1, 1, 1, 1);
560
561         horizontalSpacer_11 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
562
563         gridLayout_4->addItem(horizontalSpacer_11, 0, 2, 1, 2);
564
565         horizontalSpacer_13 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
566
567         gridLayout_4->addItem(horizontalSpacer_13, 1, 2, 1, 1);
568
569         programKrBox = new QLineEdit(programTab);
570         programKrBox->setObjectName(QStringLiteral("programKrBox"));
571
572         gridLayout_4->addWidget(programKrBox, 4, 1, 1, 1);
573
574         programKrButton = new QCheckBox(programTab);
575         programKrButton->setObjectName(QStringLiteral("programKrButton"));
576
577         gridLayout_4->addWidget(programKrButton, 3, 0, 1, 2);
578
579         horizontalSpacer_12 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
580
581         gridLayout_4->addItem(horizontalSpacer_12, 0, 4, 1, 1);
582
583         programKpExtra = new QWidget(programTab);
584         programKpExtra->setObjectName(QStringLiteral("programKpExtra"));
585         horizontalLayout_2 = new QHBoxLayout(programKpExtra);
586         horizontalLayout_2->setSpacing(12);
587         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
588         horizontalLayout_2->setContentsMargins(0, 0, 0, 0);
589         programKpSIBox = new QCheckBox(programKpExtra);
590         programKpSIBox->setObjectName(QStringLiteral("programKpSIBox"));
591
592         horizontalLayout_2->addWidget(programKpSIBox);
593
594         programKpModeBox = new QComboBox(programKpExtra);
595         programKpModeBox->setObjectName(QStringLiteral("programKpModeBox"));
596
597         horizontalLayout_2->addWidget(programKpModeBox);
598
599         horizontalSpacer_15 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
600
601         horizontalLayout_2->addItem(horizontalSpacer_15);
602
603
604         gridLayout_4->addWidget(programKpExtra, 1, 3, 1, 2);
605
606         programKrExtra = new QWidget(programTab);
607         programKrExtra->setObjectName(QStringLiteral("programKrExtra"));
608         horizontalLayout_3 = new QHBoxLayout(programKrExtra);
609         horizontalLayout_3->setSpacing(12);
610         horizontalLayout_3->setObjectName(QStringLiteral("horizontalLayout_3"));
611         horizontalLayout_3->setContentsMargins(0, 0, 0, 0);
612         programKrSIBox = new QCheckBox(programKrExtra);
613         programKrSIBox->setObjectName(QStringLiteral("programKrSIBox"));
614
615         horizontalLayout_3->addWidget(programKrSIBox);
616
617         programKrModeBox = new QComboBox(programKrExtra);
618         programKrModeBox->setObjectName(QStringLiteral("programKrModeBox"));
619
620         horizontalLayout_3->addWidget(programKrModeBox);
621
622         horizontalSpacer_16 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
623
624         horizontalLayout_3->addItem(horizontalSpacer_16);
```

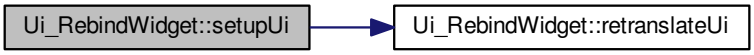
```
625
626
627         gridLayout_4->addWidget(programKrExtra, 4, 3, 1, 2);
628
629         progTipLabel = new QLabel(programTab);
630         progTipLabel->setObjectName(QStringLiteral("progTipLabel"));
631         QSizePolicy sizePolicy3(QSizePolicy::Preferred, QSizePolicy::Expanding);
632         sizePolicy3.setHorizontalStretch(0);
633         sizePolicy3.setVerticalStretch(0);
634         sizePolicy3.setHeightForWidth(progTipLabel->sizePolicy().hasHeightForWidth());
635         progTipLabel->setSizePolicy(sizePolicy3);
636         progTipLabel->setTextInteractionFlags(Qt::LinksAccessibleByMouse|
    Qt::TextSelectableByMouse);
637
638         gridLayout_4->addWidget(progTipLabel, 5, 1, 1, 4);
639
640         tabWidget->addTab(programTab, QString());
641         macroTab = new QWidget();
642         macroTab->setObjectName(QStringLiteral("macroTab"));
643         layoutWidget1 = new QWidget(macroTab);
644         layoutWidget1->setObjectName(QStringLiteral("layoutWidget1"));
645         layoutWidget1->setGeometry(QRect(40, 12, 241, 161));
646         gridLayout_8 = new QGridLayout(layoutWidget1);
647         gridLayout_8->setObjectName(QStringLiteral("gridLayout_8"));
648         gridLayout_8->setSizeConstraint(QLayout::SetNoConstraint);
649         gridLayout_8->setContentsMargins(0, 0, 0, 0);
650         groupBox_2 = new QGroupBox(layoutWidget1);
651         groupBox_2->setObjectName(QStringLiteral("groupBox_2"));
652         pteMacroBox = new QPlainTextEdit(groupBox_2);
653         pteMacroBox->setObjectName(QStringLiteral("pteMacroBox"));
654         pteMacroBox->setGeometry(QRect(0, 20, 241, 131));
655         pteMacroBox->setFocusPolicy(Qt::WheelFocus);
656         pteMacroBox->setTabChangesFocus(true);
657
658         gridLayout_8->addWidget(groupBox_2, 0, 0, 1, 1);
659
660         pteMacroComment = new QPlainTextEdit(macroTab);
661         pteMacroComment->setObjectName(QStringLiteral("pteMacroComment"));
662         pteMacroComment->setEnabled(true);
663         pteMacroComment->setGeometry(QRect(410, 10, 311, 31));
664         pteMacroComment->setFocusPolicy(Qt::WheelFocus);
665         pteMacroComment->setContextMenuPolicy(Qt::NoContextMenu);
666         pteMacroComment->setAcceptDrops(true);
667         pteMacroComment->setToolTipDuration(1);
668         pteMacroComment->setInputMethodHints(Qt::ImhNone);
669         pteMacroComment->setVerticalScrollBarPolicy(Qt::ScrollBarAlwaysOff);
670         pteMacroComment->setTabChangesFocus(true);
671         pteMacroComment->setUndoRedoEnabled(true);
672         pteMacroComment->setReadOnly(false);
673         pteMacroComment->setTabStopWidth(1);
674         pteMacroText = new QPlainTextEdit(macroTab);
675         pteMacroText->setObjectName(QStringLiteral("pteMacroText"));
676         pteMacroText->setGeometry(QRect(410, 50, 311, 66));
677         pteMacroText->setFocusPolicy(Qt::StrongFocus);
678         pteMacroText->setContextMenuPolicy(Qt::NoContextMenu);
679         pteMacroText->setAcceptDrops(false);
680         pteMacroText->setInputMethodHints(Qt::ImhNoAutoUppercase);
681         pteMacroText->setUndoRedoEnabled(false);
682         pteMacroText->setReadOnly(false);
683         pteMacroText->setTabStopWidth(1);
684         layoutWidget = new QWidget(macroTab);
685         layoutWidget->setObjectName(QStringLiteral("layoutWidget"));
686         layoutWidget->setGeometry(QRect(300, 10, 131, 74));
687         verticalLayout_2 = new QVBoxLayout(layoutWidget);
688         verticalLayout_2->setObjectName(QStringLiteral("verticalLayout_2"));
689         verticalLayout_2->setSizeConstraint(QLayout::SetNoConstraint);
690         verticalLayout_2->setContentsMargins(0, 0, 0, 0);
691         groupBox_3 = new QGroupBox(layoutWidget);
692         groupBox_3->setObjectName(QStringLiteral("groupBox_3"));
693
694         verticalLayout_2->addWidget(groupBox_3);
695
696         groupBox_4 = new QGroupBox(layoutWidget);
697         groupBox_4->setObjectName(QStringLiteral("groupBox_4"));
698
699         verticalLayout_2->addWidget(groupBox_4);
700
701         layoutWidget_2 = new QWidget(macroTab);
702         layoutWidget_2->setObjectName(QStringLiteral("layoutWidget_2"));
703         layoutWidget_2->setGeometry(QRect(300, 160, 407, 36));
704         gridLayout_10 = new QGridLayout(layoutWidget_2);
705         gridLayout_10->setObjectName(QStringLiteral("gridLayout_10"));
706         gridLayout_10->setContentsMargins(0, 0, 0, 0);
707         btnStopMacro = new QPushButton(layoutWidget_2);
708         btnStopMacro->setObjectName(QStringLiteral("btnStopMacro"));
709         btnStopMacro->setEnabled(false);
710
```

```
711         gridLayout_10->addWidget(btnStopMacro, 0, 2, 1, 1);
712 
713         btnClearMacro = new QPushButton(layoutWidget_2);
714         btnClearMacro->setObjectName(QStringLiteral("btnClearMacro"));
715 
716         gridLayout_10->addWidget(btnClearMacro, 0, 3, 1, 1);
717 
718         btnStartMacro = new QPushButton(layoutWidget_2);
719         btnStartMacro->setObjectName(QStringLiteral("btnStartMacro"));
720 
721         gridLayout_10->addWidget(btnStartMacro, 0, 1, 1, 1);
722 
723         label = new QLabel(layoutWidget_2);
724         label->setObjectName(QStringLiteral("label"));
725         label->setStyleSheet(QLatin1String("font: 75 10pt ; \n"
726 ""));
727 
728         gridLayout_10->addWidget(label, 0, 0, 1, 1);
729 
730         lbl_macro = new QLabel(macroTab);
731         lbl_macro->setObjectName(QStringLiteral("lbl_macro"));
732         lbl_macro->setGeometry(QRect(300, 130, 431, 27));
733         lbl_macro->setStyleSheet(QStringLiteral("font: 75 11pt ; color: darkblue;"));
734         layoutWidget2 = new QWidget(macroTab);
735         layoutWidget2->setObjectName(QStringLiteral("layoutWidget2"));
736         layoutWidget2->setGeometry(QRect(40, 170, 238, 28));
737         delayButtons = new QHBoxLayout(layoutWidget2);
738         delayButtons->setSpacing(0);
739         delayButtons->setObjectName(QStringLiteral("delayButtons"));
740         delayButtons->setSizeConstraint(QLayout::SetMinimumSize);
741         delayButtons->setContentsMargins(0, 0, 0, 0);
742         rb_delay_no = new QRadioButton(layoutWidget2);
743         rb_delay_no->setObjectName(QStringLiteral("rb_delay_no"));
744         rb_delay_no->setStyleSheet(QStringLiteral("font: 10pt ; color:  green;"));
745 
746         delayButtons->addWidget(rb_delay_no);
747 
748         rb_delay_default = new QRadioButton(layoutWidget2);
749         rb_delay_default->setObjectName(QStringLiteral("rb_delay_default"));
750         rb_delay_default->setStyleSheet(QStringLiteral("font: 10pt ; color:  green;"));
751         rb_delay_default->setChecked(false);
752         rb_delay_default->setAutoExclusive(true);
753 
754         delayButtons->addWidget(rb_delay_default);
755 
756         rb_delay_asTyped = new QRadioButton(layoutWidget2);
757         rb_delay_asTyped->setObjectName(QStringLiteral("rb_delay_asTyped"));
758         rb_delay_asTyped->setStyleSheet(QStringLiteral("font: 10pt ; color:  green;"));
759         rb_delay_asTyped->setChecked(true);
760 
761         delayButtons->addWidget(rb_delay_asTyped);
762 
763         txtBuffer = new QLabel(macroTab);
764         txtBuffer->setObjectName(QStringLiteral("txtBuffer"));
765         txtBuffer->setEnabled(false);
766         txtBuffer->setGeometry(QRect(300, 100, 0, 0));
767         QSizePolicy sizePolicy4(QSizePolicy::Fixed, QSizePolicy::Fixed);
768         sizePolicy4.setHorizontalStretch(0);
769         sizePolicy4.setVerticalStretch(0);
770         sizePolicy4.setHeightForWidth(txtBuffer->sizePolicy().hasHeightForWidth());
771         txtBuffer->setSizePolicy(sizePolicy4);
772         txtBuffer->setMaximumSize(QSize(0, 0));
773         txtBuffer->setContextMenuPolicy(Qt::NoContextMenu);
774 #ifndef QT_NO_TOOLTIP
775         txtBuffer->setToolTip(QStringLiteral(""));
776 #endif // QT_NO_TOOLTIP
777         txtBuffer->setToolTipDuration(0);
778 #ifndef QT_NO_STATUSTIP
779         txtBuffer->setStatusTip(QStringLiteral(""));
780 #endif // QT_NO_STATUSTIP
781 #ifndef QT_NO_WHATSTHIS
782         txtBuffer->setWhatsThis(QStringLiteral(""));
783 #endif // QT_NO_WHATSTHIS
784         txtBuffer->setFrameShadow(QFrame::Plain);
785         txtBuffer->setLineWidth(0);
786         txtBuffer->setTextFormat(Qt::PlainText);
787         txtBuffer->setIndent(0);
788         txtBuffer->setTextInteractionFlags(Qt::NoTextInteraction);
789         tabWidget->addTab(macroTab, QString());
790         pteMacroComment->raise();
791         layoutWidget->raise();
792         layoutWidget1->raise();
793         layoutWidget->raise();
794         pteMacroText->raise();
795         layoutWidget_2->raise();
796         lbl_macro->raise();
797         txtBuffer->raise();
```

```
798
799         verticalLayout->addWidget(tabWidget);
800
801         horizontalLayout = new QHBoxLayout();
802         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
803         horizontalSpacer_14 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
804
805         horizontalLayout->addItem(horizontalSpacer_14);
806
807         unbindButton = new QPushButton(RebindWidget);
808         unbindButton->setObjectName(QStringLiteral("unbindButton"));
809
810         horizontalLayout->addWidget(unbindButton);
811
812         resetButton = new QPushButton(RebindWidget);
813         resetButton->setObjectName(QStringLiteral("resetButton"));
814
815         horizontalLayout->addWidget(resetButton);
816
817         cancelButton = new QPushButton(RebindWidget);
818         cancelButton->setObjectName(QStringLiteral("cancelButton"));
819
820         horizontalLayout->addWidget(cancelButton);
821
822         applyButton = new QPushButton(RebindWidget);
823         applyButton->setObjectName(QStringLiteral("applyButton"));
824
825         horizontalLayout->addWidget(applyButton);
826
827
828         verticalLayout->addLayout(horizontalLayout);
829
830         QWidget::setTabOrder(tabWidget, typingButton);
831         QWidget::setTabOrder(typingButton, typingBox);
832         QWidget::setTabOrder(typingBox, modButton);
833         QWidget::setTabOrder(modButton, modBox);
834         QWidget::setTabOrder(modBox, fnButton);
835         QWidget::setTabOrder(fnButton, fnBox);
836         QWidget::setTabOrder(fnBox, numButton);
837         QWidget::setTabOrder(numButton, numBox);
838         QWidget::setTabOrder(numBox, mediaButton);
839         QWidget::setTabOrder(mediaButton, mediaBox);
840         QWidget::setTabOrder(mediaBox, mbButton);
841         QWidget::setTabOrder(mbButton, mbBox);
842         QWidget::setTabOrder(mbBox, mb2Button);
843         QWidget::setTabOrder(mb2Button, mb2Box);
844         QWidget::setTabOrder(mb2Box, wheelButton);
845         QWidget::setTabOrder(wheelButton, wheelBox);
846         QWidget::setTabOrder(wheelBox, dpiButton);
847         QWidget::setTabOrder(dpiButton, dpiBox);
848         QWidget::setTabOrder(dpiBox, dpiCustXBox);
849         QWidget::setTabOrder(dpiCustXBox, dpiCustYBox);
850         QWidget::setTabOrder(dpiCustYBox, animButton);
851         QWidget::setTabOrder(animButton, animBox);
852         QWidget::setTabOrder(animBox, animOnceBox);
853         QWidget::setTabOrder(animOnceBox, animKrBox);
854         QWidget::setTabOrder(animKrBox, modeButton);
855         QWidget::setTabOrder(modeButton, modeBox);
856         QWidget::setTabOrder(modeBox, modeWrapBox);
857         QWidget::setTabOrder(modeWrapBox, lightButton);
858         QWidget::setTabOrder(lightButton, lightBox);
859         QWidget::setTabOrder(lightBox, lightWrapBox);
860         QWidget::setTabOrder(lightWrapBox, lockButton);
861         QWidget::setTabOrder(lockButton, lockBox);
862         QWidget::setTabOrder(lockBox, programKpButton);
863         QWidget::setTabOrder(programKpButton, programKpBox);
864         QWidget::setTabOrder(programKpBox, programKpSIBox);
865         QWidget::setTabOrder(programKpSIBox, programKpModeBox);
866         QWidget::setTabOrder(programKpModeBox, programKrButton);
867         QWidget::setTabOrder(programKrButton, programKrBox);
868         QWidget::setTabOrder(programKrBox, programKrSIBox);
869         QWidget::setTabOrder(programKrSIBox, programKrModeBox);
870         QWidget::setTabOrder(programKrModeBox, pteMacroBox);
871         QWidget::setTabOrder(pteMacroBox, rb_delay_no);
872         QWidget::setTabOrder(rb_delay_no, rb_delay_default);
873         QWidget::setTabOrder(rb_delay_default, rb_delay_asTyped);
874         QWidget::setTabOrder(rb_delay_asTyped, pteMacroComment);
875         QWidget::setTabOrder(pteMacroComment, pteMacroText);
876         QWidget::setTabOrder(pteMacroText, btnStartMacro);
877         QWidget::setTabOrder(btnStartMacro, btnStopMacro);
878         QWidget::setTabOrder(btnStopMacro, btnClearMacro);
879         QWidget::setTabOrder(btnClearMacro, unbindButton);
880         QWidget::setTabOrder(unbindButton, resetButton);
881         QWidget::setTabOrder(resetButton, cancelButton);
882         QWidget::setTabOrder(cancelButton, applyButton);
883
```

```
884         retranslateUi(RebindWidget);
885         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    pteMacroText, SLOT(clear()));
886         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    pteMacroComment, SLOT(clear()));
887         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    btnStopMacro, SLOT(animateClick()));
888         QObject::connect(cancelButton, SIGNAL(clicked()), txtBuffer, SLOT(clear()));
889         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    txtBuffer, SLOT(clear()));
890         QObject::connect(btnStartMacro, SIGNAL(clicked()),
    txtBuffer, SLOT(clear()));
891         QObject::connect(unbindButton, SIGNAL(clicked()), txtBuffer, SLOT(clear()));
892         QObject::connect(resetButton, SIGNAL(clicked()), txtBuffer, SLOT(clear()));
893
894         tabWidget->setCurrentIndex(5);
895         applyButton->setDefault(true);
896
897
898         QMetaObject::connectSlotsByName(RebindWidget);
899     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.80.3 Field Documentation

12.80.3.1 QWidget* Ui_RebindWidget::actionTab [inherited]

Definition at line 92 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.2 QComboBox* Ui_RebindWidget::animBox [inherited]

Definition at line 84 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_animBox_currentIndexChanged(), RebindWidget::on_animButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.3 QRadioButton* Ui_RebindWidget::animButton [inherited]

Definition at line 90 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_animBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.4 QCheckBox* Ui_RebindWidget::animKrBox [inherited]

Definition at line 87 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.5 QCheckBox* Ui_RebindWidget::animOnceBox [inherited]

Definition at line 88 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.6 QWidget* Ui_RebindWidget::animTab [inherited]

Definition at line 81 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.7 QPushButton* Ui_RebindWidget::applyButton [inherited]

Definition at line 161 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.8 QPushButton* Ui_RebindWidget::btnClearMacro [inherited]

Definition at line 146 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.9 QPushButton* Ui_RebindWidget::btnStartMacro [inherited]

Definition at line 147 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.10 QPushButton* Ui_RebindWidget::btnStopMacro [inherited]

Definition at line 145 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.11 QPushButton* Ui_RebindWidget::cancelButton [inherited]

Definition at line 160 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.12 QHBoxLayout* Ui_RebindWidget::delayButtons [inherited]

Definition at line 151 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.13 QComboBox* Ui_RebindWidget::dpiBox [inherited]

Definition at line 68 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_dpiBox_currentIndexChanged(), RebindWidget::on_dpiButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBind(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.14 QRadioButton* Ui_RebindWidget::dpiButton [inherited]

Definition at line 62 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_dpiBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBind(), and Ui_RebindWidget::setupUi().

12.80.3.15 QLabel* Ui_RebindWidget::dpiCustLabel [inherited]

Definition at line 78 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_dpiBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBind(), and Ui_RebindWidget::setupUi().

12.80.3.16 QSpinBox* Ui_RebindWidget::dpiCustXBox [inherited]

Definition at line 73 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_dpiBox_currentIndexChanged(), RebindWidget::setBind(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.17 QSpinBox* Ui_RebindWidget::dpiCustYBox [inherited]

Definition at line 69 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_dpiBox_currentIndexChanged(), RebindWidget::setBind(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.18 QLabel* Ui_RebindWidget::dpiWarning [inherited]

Definition at line 74 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), RebindWidget::setBind(), and Ui_RebindWidget::setupUi().

12.80.3.19 QComboBox* Ui_RebindWidget::fnBox [inherited]

Definition at line 44 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_fnBox_currentIndexChanged(), RebindWidget::on_fnButton_clicked(), RebindWidget::RebindWidget(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.20 QRadioButton* Ui_RebindWidget::fnButton [inherited]

Definition at line 56 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_fnBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_-RebindWidget::setupUi().

12.80.3.21 QGridLayout* Ui_RebindWidget::gridLayout [inherited]

Definition at line 61 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.22 QGridLayout* Ui_RebindWidget::gridLayout_10 [inherited]

Definition at line 144 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.23 QGridLayout* Ui_RebindWidget::gridLayout_2 [inherited]

Definition at line 41 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.24 QGridLayout* Ui_RebindWidget::gridLayout_3 [inherited]

Definition at line 93 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.25 QGridLayout* Ui_RebindWidget::gridLayout_4 [inherited]

Definition at line 111 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.26 QGridLayout* Ui_RebindWidget::gridLayout_5 [inherited]

Definition at line 82 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.27 QGridLayout* Ui_RebindWidget::gridLayout_8 [inherited]

Definition at line 134 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.28 QGroupBox* Ui_RebindWidget::groupBox_2 [inherited]

Definition at line 135 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.29 QGroupBox* Ui_RebindWidget::groupBox_3 [inherited]

Definition at line 141 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.30 QGroupBox* Ui_RebindWidget::groupBox_4 [inherited]

Definition at line 142 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.31 QHBoxLayout* Ui_RebindWidget::horizontalLayout [inherited]

Definition at line 156 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.32 QHBoxLayout* Ui_RebindWidget::horizontalLayout_2 [inherited]

Definition at line 122 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.33 QHBoxLayout* Ui_RebindWidget::horizontalLayout_3 [inherited]

Definition at line 127 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.34 QSpacerItem* Ui_RebindWidget::horizontalSpacer [inherited]

Definition at line 113 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.35 QSpacerItem* Ui_RebindWidget::horizontalSpacer_10 [inherited]

Definition at line 107 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.36 QSpacerItem* Ui_RebindWidget::horizontalSpacer_11 [inherited]

Definition at line 116 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.37 QSpacerItem* Ui_RebindWidget::horizontalSpacer_12 [inherited]

Definition at line 120 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.38 QSpacerItem* Ui_RebindWidget::horizontalSpacer_13 [inherited]

Definition at line 117 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.39 QSpacerItem* Ui_RebindWidget::horizontalSpacer_14 [inherited]

Definition at line 157 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.40 QSpacerItem* Ui_RebindWidget::horizontalSpacer_15 [inherited]

Definition at line 125 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.41 QSpacerItem* Ui_RebindWidget::horizontalSpacer_16 [inherited]

Definition at line 130 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.42 QSpacerItem* Ui_RebindWidget::horizontalSpacer_17 [inherited]

Definition at line 59 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.43 QSpacerItem* Ui_RebindWidget::horizontalSpacer_18 [inherited]

Definition at line 65 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.44 QSpacerItem* Ui_RebindWidget::horizontalSpacer_19 [inherited]

Definition at line 79 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.45 QSpacerItem* Ui_RebindWidget::horizontalSpacer_2 [inherited]

Definition at line 47 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.46 QSpacerItem* Ui_RebindWidget::horizontalSpacer_20 [inherited]

Definition at line 66 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.47 QSpacerItem* Ui_RebindWidget::horizontalSpacer_21 [inherited]

Definition at line 80 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.48 QSpacerItem* Ui_RebindWidget::horizontalSpacer_22 [inherited]

Definition at line 76 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.49 QSpacerItem* Ui_RebindWidget::horizontalSpacer_23 [inherited]

Definition at line 91 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.50 QSpacerItem* Ui_RebindWidget::horizontalSpacer_24 [inherited]

Definition at line 85 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.51 QSpacerItem* Ui_RebindWidget::horizontalSpacer_25 [inherited]

Definition at line 86 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.52 QSpacerItem* Ui_RebindWidget::horizontalSpacer_26 [inherited]

Definition at line 89 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.53 QSpacerItem* Ui_RebindWidget::horizontalSpacer_3 [inherited]

Definition at line 50 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.54 QSpacerItem* Ui_RebindWidget::horizontalSpacer_4 [inherited]

Definition at line 54 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.55 QSpacerItem* Ui_RebindWidget::horizontalSpacer_5 [inherited]

Definition at line 94 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.56 QSpacerItem* Ui_RebindWidget::horizontalSpacer_6 [inherited]

Definition at line 109 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.57 QSpacerItem* Ui_RebindWidget::horizontalSpacer_7 [inherited]

Definition at line 108 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.58 QSpacerItem* Ui_RebindWidget::horizontalSpacer_8 [inherited]

Definition at line 106 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.59 QSpacerItem* Ui_RebindWidget::horizontalSpacer_9 [inherited]

Definition at line 53 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.60 QWidget* Ui_RebindWidget::keyTab [inherited]

Definition at line 40 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.61 QLabel* Ui_RebindWidget::label [inherited]

Definition at line 148 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.62 QWidget* Ui_RebindWidget::layoutWidget [inherited]

Definition at line 139 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.63 QWidget* Ui_RebindWidget::layoutWidget1 [inherited]

Definition at line 133 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.64 QWidget* Ui_RebindWidget::layoutWidget2 [inherited]

Definition at line 150 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.65 QWidget* Ui_RebindWidget::layoutWidget_2 [inherited]

Definition at line 143 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.66 QLabel* Ui_RebindWidget::lbl_macro [inherited]

Definition at line 149 of file ui_rebindwidget.h.

Referenced by RebindWidget::helpStatus(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.67 QComboBox* Ui_RebindWidget::lightBox [inherited]

Definition at line 102 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_lightBox_currentIndexChanged(), RebindWidget::on_lightButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.68 QRadioButton* Ui_RebindWidget::lightButton [inherited]

Definition at line 104 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_lightBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.69 QCheckBox* Ui_RebindWidget::lightWrapBox [inherited]

Definition at line 97 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_lightBox_currentIndexChanged(), RebindWidget::RebindWidget(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.70 QComboBox* Ui_RebindWidget::lockBox [inherited]

Definition at line 103 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_lockBox_currentIndexChanged(), RebindWidget::on_lockButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.71 QRadioButton* Ui_RebindWidget::lockButton [inherited]

Definition at line 100 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_lockBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.72 QWidget* Ui_RebindWidget::macroTab [inherited]

Definition at line 132 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.73 QComboBox* Ui_RebindWidget::mb2Box [inherited]

Definition at line 67 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_mb2Box_currentIndexChanged(), RebindWidget::on_mb2Button_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.74 QRadioButton* Ui_RebindWidget::mb2Button [inherited]

Definition at line 77 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_mb2Box_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.75 QComboBox* Ui_RebindWidget::mbBox [inherited]

Definition at line 70 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_mbBox_currentIndexChanged(), RebindWidget::on_mbButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.76 QRadioButton* Ui_RebindWidget::mbButton [inherited]

Definition at line 72 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_mbBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.77 QComboBox* Ui_RebindWidget::mediaBox [inherited]

Definition at line 57 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_mediaBox_currentIndexChanged(), RebindWidget::on_mediaButton_clicked(), RebindWidget::RebindWidget(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.78 QRadioButton* Ui_RebindWidget::mediaButton [inherited]

Definition at line 51 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_mediaBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.79 QComboBox* Ui_RebindWidget::modBox [inherited]

Definition at line 49 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_modBox_currentIndexChanged(), RebindWidget::on_modButton_clicked(), RebindWidget::RebindWidget(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.80 QRadioButton* Ui_RebindWidget::modButton [inherited]

Definition at line 55 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_modBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_-RebindWidget::setupUi().

12.80.3.81 QComboBox* Ui_RebindWidget::modeBox [inherited]

Definition at line 105 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_modeBox_currentIndexChanged(), Rebind-Widget::on_modeButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBind(), RebindWidget::set-Box(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.82 QRadioButton* Ui_RebindWidget::modeButton [inherited]

Definition at line 95 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_modeBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_-RebindWidget::setupUi().

12.80.3.83 QCheckBox* Ui_RebindWidget::modeWrapBox [inherited]

Definition at line 96 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_modeBox_currentIndexChanged(), Rebind-Widget::RebindWidget(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget-::setupUi().

12.80.3.84 QWidget* Ui_RebindWidget::mouseTab [inherited]

Definition at line 60 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.85 QComboBox* Ui_RebindWidget::numBox [inherited]

Definition at line 42 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_numBox_currentIndexChanged(), Rebind-Widget::on_numButton_clicked(), RebindWidget::RebindWidget(), Ui_RebindWidget::retranslateUi(), Rebind-Widget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.86 QRadioButton* Ui_RebindWidget::numButton [inherited]

Definition at line 58 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_numBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_-RebindWidget::setupUi().

12.80.3.87 QLineEdit* Ui_RebindWidget::programKpBox [inherited]

Definition at line 115 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKpBox_textChanged(), RebindWidget-::on_programKpButton_clicked(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget-::setupUi().

12.80.3.88 QCheckBox* Ui_RebindWidget::programKpButton [inherited]

Definition at line 114 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKpBox_textChanged(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), and Ui_RebindWidget::setupUi().

12.80.3.89 QWidget* Ui_RebindWidget::programKpExtra [inherited]

Definition at line 121 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKpBox_textChanged(), RebindWidget::RebindWidget(), and Ui_RebindWidget::setupUi().

12.80.3.90 QComboBox* Ui_RebindWidget::programKpModeBox [inherited]

Definition at line 124 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKpSIBox_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.91 QCheckBox* Ui_RebindWidget::programKpSIBox [inherited]

Definition at line 123 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.92 QLineEdit* Ui_RebindWidget::programKrBox [inherited]

Definition at line 118 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKrBox_textChanged(), RebindWidget::on_programKrButton_clicked(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.93 QCheckBox* Ui_RebindWidget::programKrButton [inherited]

Definition at line 119 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKrBox_textChanged(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), and Ui_RebindWidget::setupUi().

12.80.3.94 QWidget* Ui_RebindWidget::programKrExtra [inherited]

Definition at line 126 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKrBox_textChanged(), RebindWidget::RebindWidget(), and Ui_RebindWidget::setupUi().

12.80.3.95 QComboBox* Ui_RebindWidget::programKrModeBox [inherited]

Definition at line 129 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKrSIBox_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.96 QCheckBox* Ui_RebindWidget::programKrSlBox [inherited]

Definition at line 128 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.97 QWidget* Ui_RebindWidget::programTab [inherited]

Definition at line 110 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.98 QLabel* Ui_RebindWidget::progTipLabel [inherited]

Definition at line 131 of file ui_rebindwidget.h.

Referenced by RebindWidget::RebindWidget(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.99 QPlainTextEdit* Ui_RebindWidget::pteMacroBox [inherited]

Definition at line 136 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::convertMacroBox(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_pteMacroBox_textChanged(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.100 QPlainTextEdit* Ui_RebindWidget::pteMacroComment [inherited]

Definition at line 137 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.101 QPlainTextEdit* Ui_RebindWidget::pteMacroText [inherited]

Definition at line 138 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_btnStartMacro_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.102 QRadioButton* Ui_RebindWidget::rb_delay_asTyped [inherited]

Definition at line 154 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setCorrectRadioButton(), and Ui_RebindWidget::setupUi().

12.80.3.103 QRadioButton* Ui_RebindWidget::rb_delay_default [inherited]

Definition at line 153 of file ui_rebindwidget.h.

Referenced by RebindWidget::convertMacroBox(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setCorrectRadioButton(), and Ui_RebindWidget::setupUi().

12.80.3.104 QRadioButton* Ui_RebindWidget::rb_delay_no [inherited]

Definition at line 152 of file ui_rebindwidget.h.

Referenced by RebindWidget::convertMacroBox(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setCorrectRadioButton(), and Ui_RebindWidget::setupUi().

12.80.3.105 QPushButton* Ui_RebindWidget::resetButton [inherited]

Definition at line 159 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.106 QTabWidget* Ui_RebindWidget::tabWidget [inherited]

Definition at line 39 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::retranslateUi(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.107 QLabel* Ui_RebindWidget::txtBuffer [inherited]

Definition at line 155 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::convertMacroBox(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.108 QComboBox* Ui_RebindWidget::typingBox [inherited]

Definition at line 43 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_typingBox_currentIndexChanged(), RebindWidget::on_typingButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBind(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.109 QRadioButton* Ui_RebindWidget::typingButton [inherited]

Definition at line 48 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_typingBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.110 QPushButton* Ui_RebindWidget::unbindButton [inherited]

Definition at line 158 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

12.80.3.111 QVBoxLayout* Ui_RebindWidget::verticalLayout [inherited]

Definition at line 38 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.112 QVBoxLayout* Ui_RebindWidget::verticalLayout_2 [inherited]

Definition at line 140 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.113 QSpacerItem* Ui_RebindWidget::verticalSpacer [inherited]

Definition at line 112 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.114 QSpacerItem* Ui_RebindWidget::verticalSpacer_10 [inherited]

Definition at line 45 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.115 QSpacerItem* Ui_RebindWidget::verticalSpacer_11 [inherited]

Definition at line 83 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.116 QSpacerItem* Ui_RebindWidget::verticalSpacer_2 [inherited]

Definition at line 63 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.117 QSpacerItem* Ui_RebindWidget::verticalSpacer_3 [inherited]

Definition at line 46 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.118 QSpacerItem* Ui_RebindWidget::verticalSpacer_4 [inherited]

Definition at line 52 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.119 QSpacerItem* Ui_RebindWidget::verticalSpacer_5 [inherited]

Definition at line 101 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.120 QSpacerItem* Ui_RebindWidget::verticalSpacer_6 [inherited]

Definition at line 99 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.121 QSpacerItem* Ui_RebindWidget::verticalSpacer_7 [inherited]

Definition at line 98 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.122 QSpacerItem* Ui_RebindWidget::verticalSpacer_8 [inherited]

Definition at line 64 of file ui_rebindwidget.h.

Referenced by Ui_RebindWidget::setupUi().

12.80.3.123 QComboBox* Ui_RebindWidget::wheelBox [inherited]

Definition at line 71 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_wheelBox_currentIndexChanged(), RebindWidget::on_wheelButton_clicked(), Ui_RebindWidget::retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and Ui_RebindWidget::setupUi().

12.80.3.124 QRadioButton* Ui_RebindWidget::wheelButton [inherited]

Definition at line 75 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_wheelBox_currentIndexChanged(), Ui_RebindWidget::retranslateUi(), and Ui_RebindWidget::setupUi().

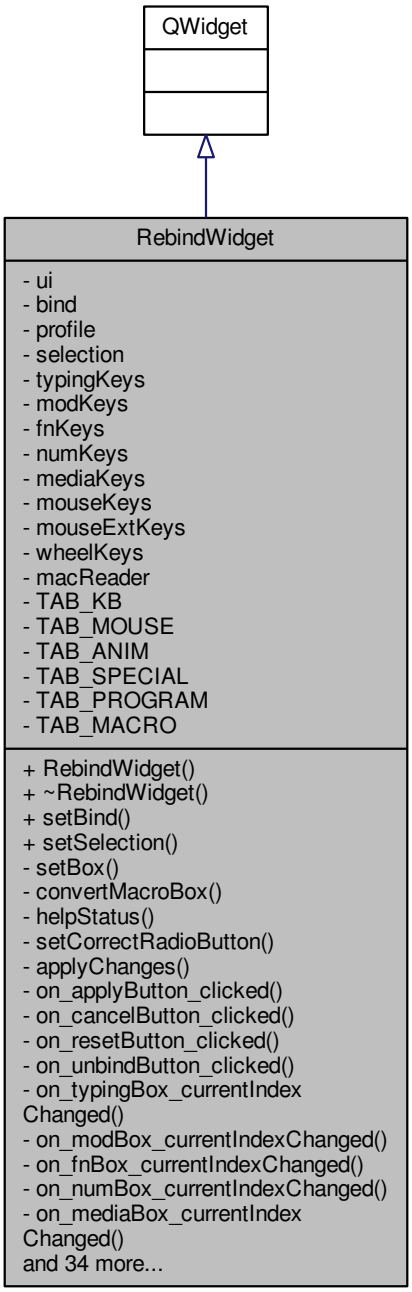
The documentation for this class was generated from the following file:

- src/ckb/ui_rebindwidget.h

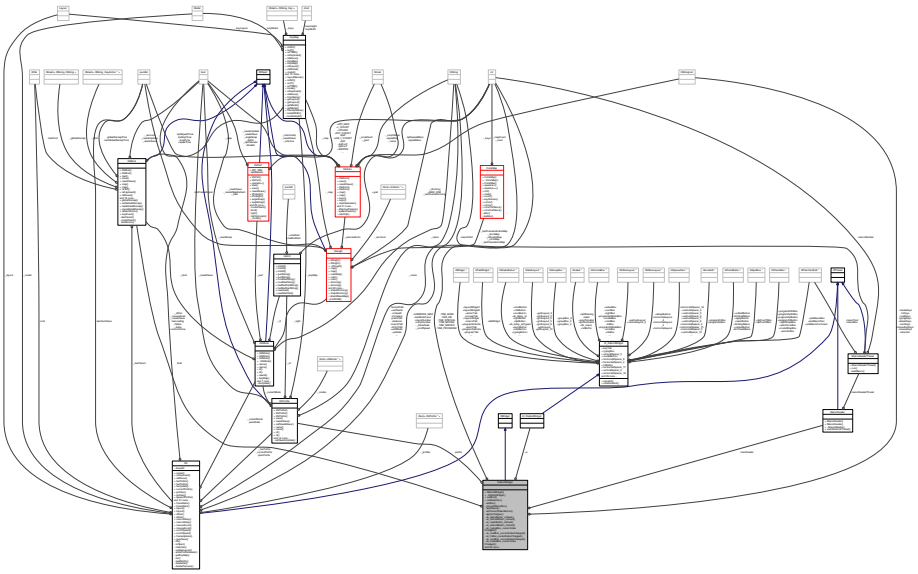
12.81 RebindWidget Class Reference

`#include <src/ckb/rebindwidget.h>`

Inheritance diagram for RebindWidget:



Collaboration diagram for RebindWidget:



Public Member Functions

- RebindWidget (QWidget *parent)
- ~RebindWidget ()
- void setBind (KbBind *newBind, KbProfile *newProfile)
- void setSelection (const QStringList &newSelection, bool applyPrevious=false)

Private Slots

- void applyChanges (const QStringList &keys, bool doUnbind)
- void on_applyButton_clicked ()
- void on_cancelButton_clicked ()
- void on_resetButton_clicked ()
- void on_unbindButton_clicked ()
- void on_typingBox_currentIndexChanged (int index)
- void on_modBox_currentIndexChanged (int index)
- void on_fnBox_currentIndexChanged (int index)
- void on_numBox_currentIndexChanged (int index)
- void on_mediaBox_currentIndexChanged (int index)
- void on_mbBox_currentIndexChanged (int index)
- void on_mb2Box_currentIndexChanged (int index)
- void on_wheelBox_currentIndexChanged (int index)
- void on_dpiBox_currentIndexChanged (int index)
- void on_modeBox_currentIndexChanged (int index)
- void on_lightBox_currentIndexChanged (int index)
- void on_lockBox_currentIndexChanged (int index)
- void on_programKpBox_textChanged (const QString &arg1)
- void on_programKrBox_textChanged (const QString &arg1)
- void on_animBox_currentIndexChanged (int index)
- void on_typingButton_clicked (bool checked)
- void on_modButton_clicked (bool checked)
- void on_fnButton_clicked (bool checked)

- void on_numButton_clicked (bool checked)
- void on_mediaButton_clicked (bool checked)
- void on_mbButton_clicked (bool checked)
- void on_mb2Button_clicked (bool checked)
- void on_wheelButton_clicked (bool checked)
- void on_dpiButton_clicked (bool checked)
- void on_modeButton_clicked (bool checked)
- void on_lightButton_clicked (bool checked)
- void on_lockButton_clicked (bool checked)
- void on_programKpButton_clicked (bool checked)
- void on_programKrButton_clicked (bool checked)
- void on_programKpSIBox_clicked (bool checked)
- void on_programKrSIBox_clicked (bool checked)
- void on_animButton_clicked (bool checked)
- void on_pteMacroBox_textChanged ()
- void on_btnStartMacro_clicked ()

  RebindWidget::on_btnStartMacro_clicked starts macro recording. A new notification channel and MacroReader are created to do the job.
- void on_btnStopMacro_clicked ()

  RebindWidget::on_btnStopMacro_clicked ends the macro recording. Notify channel ist closed, the ReaderThread is deleted when the notification is really down.
- void on_btnClearMacro_clicked ()

  RebindWidget::on_btnClearMacro_clicked changes the help info an the panel. The job of clearing the input panels is triggerd with signal/slot via the RebindWidget.ui file.
- void on_rb_delay_no_toggled (bool checked)

  RebindWidget::on_rb_delay_no_toggled.
- void on_rb_delay_asTyped_toggled (bool checked)

  RebindWidget::on_rb_delay_asTyped_toggled.
- void on_rb_delay_default_toggled (bool checked)

  RebindWidget::on_rb_delay_default_toggled.

Private Member Functions

- void setBox (QWidget *box)
- void convertMacroBox ()

  RebindWidget::convertMacroBox converts the macroBox content. The KB sends each keypress as "key [+|-]<keyname><newline>" This is followed by timing information (delays between keystrokes).
- void helpStatus (int status)

  RebindWidget::helpStatus shows a help line in the ui.
- void setCorrectRadioButton (QString macdef)

  RebindWidget::setCorrectRadioButton.

Private Attributes

- Ui::RebindWidget * ui
- KbBind * bind
- KbProfile * profile
- QStringList selection
- QStringList typingKeys
- QStringList modKeys
- QStringList fnKeys
- QStringList numKeys
- QStringList mediaKeys

- QStringList mouseKeys
- QStringList mouseExtKeys
- QStringList wheelKeys
- MacroReader * macReader

  macReader holds the MacroReader when macro recording starts.

Static Private Attributes

- static const int TAB_KB = 0
- static const int TAB_MOUSE = 1
- static const int TAB_ANIM = 2
- static const int TAB_SPECIAL = 3
- static const int TAB_PROGRAM = 4
- static const int TAB_MACRO = 5

12.81.1 Detailed Description

Definition at line 15 of file rebindwidget.h.

12.81.2 Constructor & Destructor Documentation

12.81.2.1 RebindWidget::RebindWidget (QWidget * *parent*) [explicit]

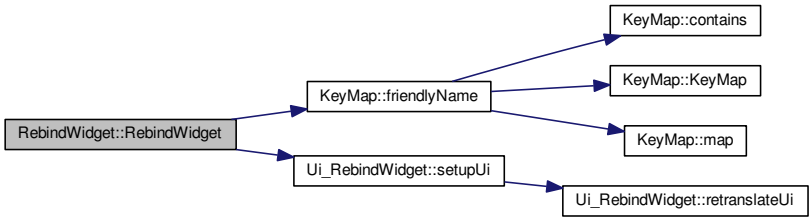
Definition at line 9 of file rebindwidget.cpp.

References Ui_RebindWidget::fnBox, fnKeys, KeyMap::friendlyName(), Ui_RebindWidget::lightWrapBox, Ui_RebindWidget::mediaBox, mediaKeys, Ui_RebindWidget::modBox, Ui_RebindWidget::modeWrapBox, modKeys, mouseExtKeys, mouseKeys, Ui_RebindWidget::numBox, numKeys, Ui_RebindWidget::programKpExtra, Ui_RebindWidget::programKrExtra, Ui_RebindWidget::progTipLabel, Ui_RebindWidget::setupUi(), ui, and wheelKeys.

```
9                                           :
10     QWidget(parent),
11     ui(new Ui::RebindWidget),
12     bind(0), profile(0), macReader(0)
13 {
14     ui->setupUi(this);
15     ui->lightWrapBox->hide();
16     ui->modeWrapBox->hide();
17     ui->programKpExtra->hide();
18     ui->programKrExtra->hide();
19
20     // Populate key lists
21     modKeys << "caps" << "lshift" << "rshift" << "lctrl" << "rctrl" << "lwin" << "rwin" << "lalt" <<
       "ralt" << "rmenu" << "fn";
22     fnKeys << "esc" << "f1" << "f2" << "f3" << "f4" << "f5" << "f6" << "f7" << "f8" << "f9" << "f10"
      << "f11" << "f12" << "prtscn" << "scroll" << "pause" << "f13" << "f14" << "f15" << "f16" << "f17" << "f18" <
      < "f19" << "f20" << "ins" << "del" << "home" << "end" << "pgup" << "pgdn" << "power" << "lightup" << "
      lightdn" << "up" << "down" << "left" << "right";
23     numKeys << "numlock" << "num0" << "num1" << "num2" << "num3" << "num4" << "num5" << "num6" << "
      num7" << "num8" << "num9" << "numslash" << "numstar" << "numminus" << "numplus" << "numdot" << "numenter";
24     mediaKeys << "stop" << "prev" << "play" << "next" << "volup" << "voldn" << "mute" << "eject";
25     mouseKeys << "mouse1" << "mouse2" << "mouse3";
26     mouseExtKeys << "mouse4" << "mouse5" << "mouse6" << "mouse7" << "mouse8";
27     wheelKeys << "wheelup" << "wheeldn";
28     // Populate boxes
29     foreach(const QString& key, modKeys)
30         ui->modBox->addItem(KeyMap::friendlyName(key));
31     foreach(const QString& key, fnKeys)
32         ui->fnBox->addItem(KeyMap::friendlyName(key).replace("\n", " / "));
33     foreach(const QString& key, numKeys)
34         ui->numBox->addItem(KeyMap::friendlyName(key).remove("NumPad "));
35     foreach(const QString& key, mediaKeys)
36         ui->mediaBox->addItem(KeyMap::friendlyName(key));
37 #ifdef Q_OS_MACX
38     // Replace some OSX keys with their actual meanings
39     ui->modBox->setItemText(modKeys.indexOf("lwin") + 1, "Left Cmd");
40     ui->modBox->setItemText(modKeys.indexOf("rwin") + 1, "Right Cmd");
41     ui->modBox->setItemText(modKeys.indexOf("lalt") + 1, "Left Option");
```

```
42     ui->modBox->setItemText(modKeys.indexOf("ralt") + 1, "Right Option");
43     ui->fnBox->setItemText(fnKeys.indexOf("prtscn") + 1, "F13");
44     ui->fnBox->setItemText(fnKeys.indexOf("scroll") + 1, "F14");
45     ui->fnBox->setItemText(fnKeys.indexOf("pause") + 1, "F15");
46     ui->fnBox->setItemText(fnKeys.indexOf("f13") + 1, "F13 (Windows/Linux)");
47     ui->fnBox->setItemText(fnKeys.indexOf("f14") + 1, "F14 (Windows/Linux)");
48     ui->fnBox->setItemText(fnKeys.indexOf("f15") + 1, "F15 (Windows/Linux)");
49     ui->fnBox->setItemText(fnKeys.indexOf("ins") + 1, "Help");
50     ui->numBox->setItemText(numKeys.indexOf("numlock") + 1, "Clear");
51
52     // Add tip label
53     ui->progTipLabel->setText("<p style=\"line-height:150%\">Tip: use the <font face=\"
       monospace\">open</font> command to launch a file, directory, or app. For instance, to start Safari:<br /><font
        face=\"monospace\">  open /Applications/Safari.app</font></p>");
54 #else
55     ui->progTipLabel->setText("<p style=\"line-height:150%\">Tip: use <font face=\"monospace
       \">xdg-open</font> to launch a file or directory. For instance, to open your home folder:<br /><font face=\"
       monospace\">  xdg-open " + QStandardPaths::writableLocation(QStandardPaths::HomeLocation) + "
       </font></p>");
56 #endif
57 }
```

Here is the call graph for this function:



12.81.2.2 RebindWidget::~RebindWidget ( )

Definition at line 59 of file rebindwidget.cpp.

References ui.

```
59                                {
60     delete ui;
61 }
```

12.81.3 Member Function Documentation

12.81.3.1 void RebindWidget::applyChanges (const QStringList & *keys*, bool *doUnbind*) [private], [slot]

G-key macro handling: Set the macro definiton for all keys selected (indeed, it may be multiple keys). First, concat the Macro Key Definion and the Macro plain text after escaping possible colos in the parts for Macro Text and Macro Comment.

But first, there is a special condition to handle: You have recorded a macro with timing infos. Afterwards you changed manually the timing infos in the pteMacroBox and press Apply. In that case we must overwrite the txtBuffer to remember your changes.

Todo There is still a bug in the state machine: If you record a macro in asTyped-mode, switch to another mode and change the vontent of the pteMacroBox manually, then the changes are not saved in the timing buffer. But anyhow, let's do more relevant things...

Definition at line 334 of file rebindwidget.cpp.

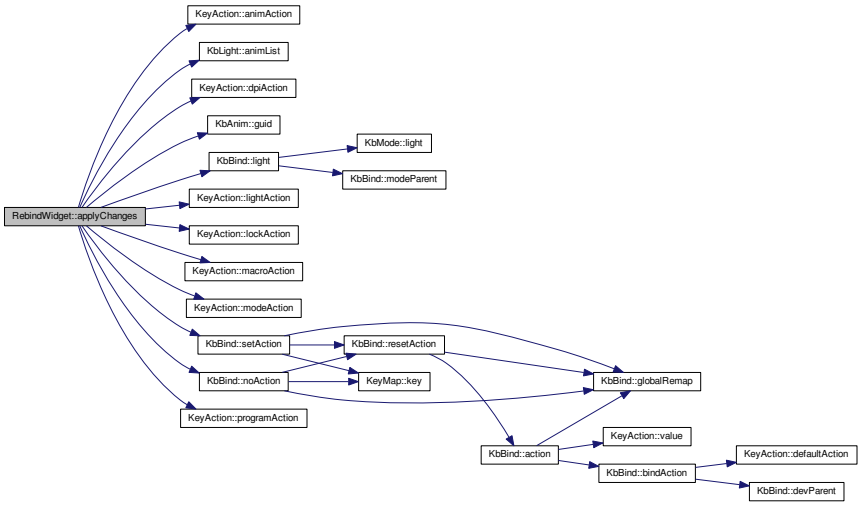
References anim, KeyAction::animAction(), Ui_RebindWidget::animBox, Ui_RebindWidget::animKrBox, KbLight::animList(), Ui_RebindWidget::animOnceBox, bind, DPI_OFFSET, KeyAction::dpiAction(), Ui_RebindWidget::dpiBox, Ui_RebindWidget::dpiCustXBox, Ui_RebindWidget::dpiCustYBox, Ui_RebindWidget::fnBox, fnKeys, KbAnim::guid(), KbBind::light(), KeyAction::lightAction(), Ui_RebindWidget::lightBox, Ui_RebindWidget::lightWrapBox, KeyAction::lockAction(), Ui_RebindWidget::lockBox, KeyAction::macroAction(), Ui_RebindWidget::mb2Box, Ui_RebindWidget::mbBox, Ui_RebindWidget::mediaBox, mediaKeys, Ui_RebindWidget::modBox, KeyAction::modeAction(), Ui_RebindWidget::modeBox, Ui_RebindWidget::modeWrapBox, modKeys, mouseExtKeys, mouseKeys, KbBind::noAction(), Ui_RebindWidget::numBox, numKeys, KeyAction::PROGRAM_PR_MULTI, KeyAction::PROGRAM_RE_INDEF, KeyAction::PROGRAM_RE_KPSTOP, KeyAction::PROGRAM_RE_MULTI, KeyAction::programAction(), Ui_RebindWidget::programKpBox, Ui_RebindWidget::programKpModeBox, Ui_RebindWidget::programKpSIBox, Ui_RebindWidget::programKrBox, Ui_RebindWidget::programKrModeBox, Ui_RebindWidget::programKrSIBox, Ui_RebindWidget::pteMacroBox, Ui_RebindWidget::pteMacroComment, Ui_RebindWidget::pteMacroText, Ui_RebindWidget::rb_delay_asTyped, KbBind::setAction(), Ui_RebindWidget::txtBuffer, Ui_RebindWidget::typingBox, typingKeys, ui, Ui_RebindWidget::wheelBox, and wheelKeys.

Referenced by on_applyButton_clicked(), and setSelection().

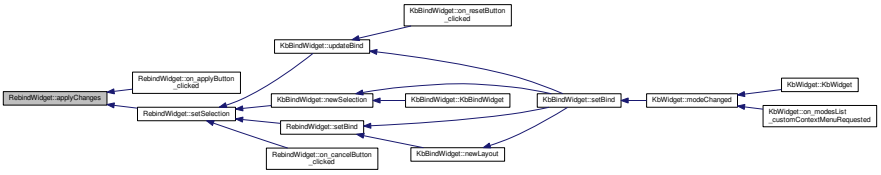
```
334                                                                     {
335     if(ui->typingBox->currentIndex() > 0)
336         bind->setAction(keys, typingKeys[ui->typingBox->currentIndex() -
    1]);
337     else if(ui->modBox->currentIndex() > 0)
338         bind->setAction(keys, modKeys[ui->modBox->currentIndex() - 1]);
339     else if(ui->fnBox->currentIndex() > 0)
340         bind->setAction(keys, fnKeys[ui->fnBox->currentIndex() - 1]);
341     else if(ui->numBox->currentIndex() > 0)
342         bind->setAction(keys, numKeys[ui->numBox->currentIndex() - 1]);
343     else if(ui->mediaBox->currentIndex() > 0)
344         bind->setAction(keys, mediaKeys[ui->mediaBox->currentIndex() - 1]);
345     else if(ui->mbBox->currentIndex() > 0)
346         bind->setAction(keys, mouseKeys[ui->mbBox->currentIndex() - 1]);
347     else if(ui->mb2Box->currentIndex() > 0)
348         bind->setAction(keys, mouseExtKeys[ui->mb2Box->currentIndex() - 1]
    );
349     else if(ui->wheelBox->currentIndex() > 0)
350         bind->setAction(keys, wheelKeys[ui->wheelBox->currentIndex() - 1]);
351     else if(ui->dpiBox->currentIndex() > 0)
352         bind->setAction(keys, KeyAction::dpiAction(
    ui->dpiBox->currentIndex() - DPI_OFFSET, ui->dpiCustXBox->value(),
    ui->dpiCustYBox->value()));
353     else if(ui->modeBox->currentIndex() > 0)
354         bind->setAction(keys, KeyAction::modeAction(
    ui->modeBox->currentIndex() - 3 - (ui->modeWrapBox->isChecked() &&
    ui->modeBox->currentIndex() < 3 ? 2 : 0)));
355     else if(ui->lightBox->currentIndex() > 0)
356         bind->setAction(keys, KeyAction::lightAction(
    ui->lightBox->currentIndex() - 1 + (ui->lightWrapBox->isChecked() ? 2 : 0)));
357     else if(ui->lockBox->currentIndex() > 0)
358         bind->setAction(keys, KeyAction::lockAction(
    ui->lockBox->currentIndex() - 1));
359     else if(ui->animBox->currentIndex() > 0){
360         KbAnim* anim = bind->light()->animList().at(
    ui->animBox->currentIndex() - 1);
361         bind->setAction(keys, KeyAction::animAction(anim->
    guid(), !ui->animOnceBox->isChecked(), ui->animKrBox->isChecked()));
362     } else if(!ui->programKpBox->text().isEmpty() || !ui->
    programKrBox->text().isEmpty()){
363         int kpStop = 0, krStop = 0;
364         if(!ui->programKpBox->text().isEmpty()){
365             if(!ui->programKpSIBox->isChecked())
366                 kpStop = KeyAction::PROGRAM_PR_MULTI;
367             else
368                 kpStop = ui->programKpModeBox->currentIndex();
369         }
370         if(!ui->programKrBox->text().isEmpty()){
371             if(!ui->programKrSIBox->isChecked())
372                 krStop = KeyAction::PROGRAM_RE_MULTI;
373             else if(ui->programKrModeBox->currentIndex() == 0)
374                 krStop = KeyAction::PROGRAM_RE_INDEF;
375             else
376                 krStop = KeyAction::PROGRAM_RE_KPSTOP;
377         }
378         bind->setAction(keys, KeyAction::programAction(
    ui->programKpBox->text(), ui->programKrBox->text(), kpStop | krStop));
379     } else if (ui->pteMacroBox->toPlainText().length() > 0) {
384
389         if (ui->rb_delay_asTyped->isChecked()) ui->txtBuffer->setText(
    ui->pteMacroBox->toPlainText());
390
396         QString mac;
```

```
397         mac = ui->txtBuffer->text();
398         mac = ui->pteMacroComment->toPlainText().replace(":", "&das_IST_31N_col0n;") + ":"
      + mac;
399         mac = ui->pteMacroText->toPlainText().replace(":", "&das_IST_31N_col0n;") + ":" + mac
    ;
400         mac = ui->pteMacroBox->toPlainText() + ":" + mac;
401         bind->setAction(keys, KeyAction::macroAction(mac));
402     } else if(doUnbind)
403         bind->noAction(keys);
404 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.81.3.2 void RebindWidget::convertMacroBox ( ) [private]

The ckb-daemon needs a shorter format, only " [+|-]<keyname>=<delay>", multiple entries are separated by comma.

That function does the conversion.

Definition at line 808 of file rebindwidget.cpp.

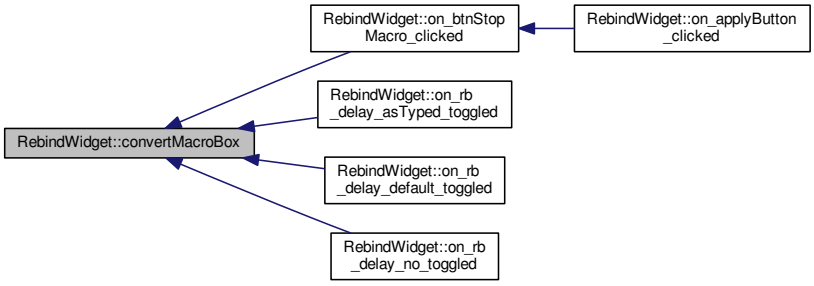
References Ui_RebindWidget::pteMacroBox, Ui_RebindWidget::rb_delay_default, Ui_RebindWidget::rb_delay_no, Ui_RebindWidget::txtBuffer, and ui.

Referenced by on_btnStopMacro_clicked(), on_rb_delay_asTyped_toggled(), on_rb_delay_default_toggled(), and on_rb_delay_no_toggled().

```
808                                 {
809     QString in;
810
```

```
811     // Remember the original input stream before it is converted.
812     // In case of new choice of delay mode we have to restore it.
813     if (ui->txtBuffer->text() == "") {
814         ui->txtBuffer->setText(ui->pteMacroBox->toPlainText());
815         in = ui->pteMacroBox->toPlainText();
816     } else in = ui->txtBuffer->text();
817
818     in.replace (QRegExp("\n"), ",");    // first join all in one line
819     in.replace (QRegExp("key "), "");   // then delete keyword "key" followed by space
820     in.replace (QRegExp(",="), "=");    // at last join each keystroke with its delay parameter
821
822     // How to deal with the delay params?
823     // Because the three radio buttons are mututally exclusive,
824     // we can run through the if-chain w/o conflicts.
825     // If rb_delay_asTyped is checked, do nothing, because that's the standard.
826
827     if (ui->rb_delay_default->isChecked()) {
828         in.replace(QRegExp("=\\d+,"), ",");  // Delete the timing infos, use default value
829         in.replace(QRegExp("=\\d+$"), "");   // The last entry is without comma
830     }
831     if (ui->rb_delay_no->isChecked()) {
832         in.replace(QRegExp("=\\d+,"), "=0,");  // Set timing infos to zero for no delay
833         in.replace(QRegExp("=\\d+$"), "=0");   // Again the last entry w/o comma
834         in.replace(QRegExp("([\\+\\-]\\w+),"), "\\1=0,");  // If no delay is given, force it to zero
835         in.replace(QRegExp("([\\+\\-]\\w+)$"), "\\1=0");
836     }
837
838     // Show the new format by replacing the older one.
839     ui->pteMacroBox->setPlainText(in);
840 }
```

Here is the caller graph for this function:



12.81.3.3 void RebindWidget::helpStatus (int *status*) [private]

Parameters

status	determines what to display.

Definition at line 782 of file rebindwidget.cpp.

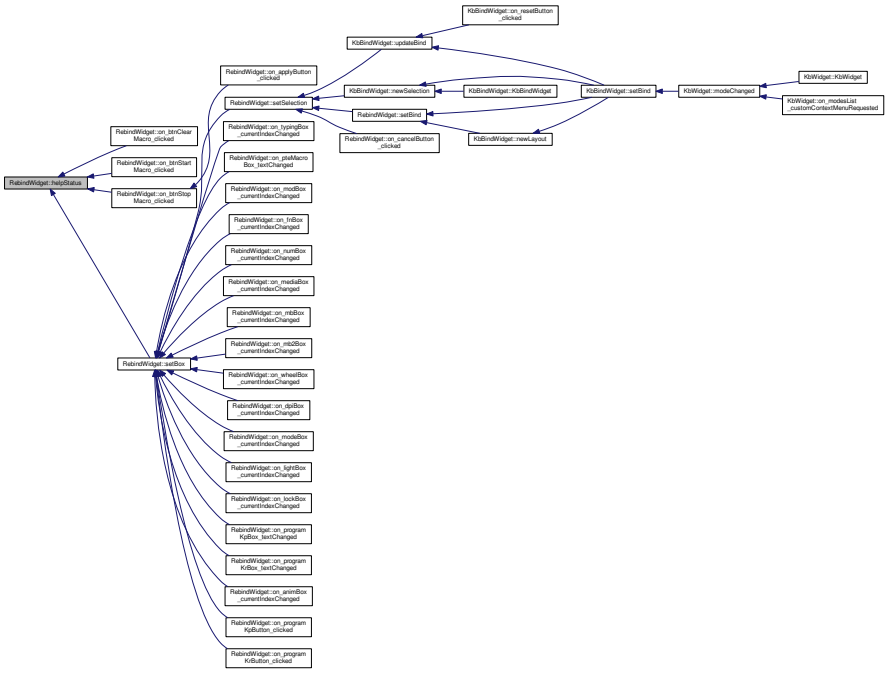
References Ui_RebindWidget::lbl_macro, and ui.

Referenced by on_btnClearMacro_clicked(), on_btnStartMacro_clicked(), on_btnStopMacro_clicked(), and setBox().

```
782                                     {
783     switch (status) {
784     case 1:
785         ui->lbl_macro->setText("Type in a macro name in the comment box and click start.");
786         break;
787     case 2:
788         ui->lbl_macro->setText("Type your macro and click stop when finished.");
789         break;
790     case 3:
791         ui->lbl_macro->setText("Click Apply or change values in Macro Key Actions in advance.");
792         break;
```

```
793     default:
794         ui->lbl_macro->setText(QString("Oops: Some magic in RebindWidget::helpStatus (%1)").arg(
    status));
795     }
796 }
```

Here is the caller graph for this function:



12.81.3.4 void RebindWidget::on_animBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 613 of file rebindwidget.cpp.

References Ui_RebindWidget::animBox, Ui_RebindWidget::animButton, setBox(), and ui.

```
613                                                     {
614     if(index == 0)
615         ui->animButton->setChecked(false);
616     else {
617         ui->animButton->setChecked(true);
618         setBox(ui->animBox);
619     }
620 }
```

Here is the call graph for this function:

12.81.3.5 void RebindWidget::on_animButton_clicked (bool *checked*) [private], [slot]

Definition at line 710 of file rebindwidget.cpp.

References Ui_RebindWidget::animBox, and ui.

```
710                                                 {
711     if(checked && ui->animBox->currentIndex() == 0)
712         ui->animBox->setCurrentIndex(1);
713 }
```

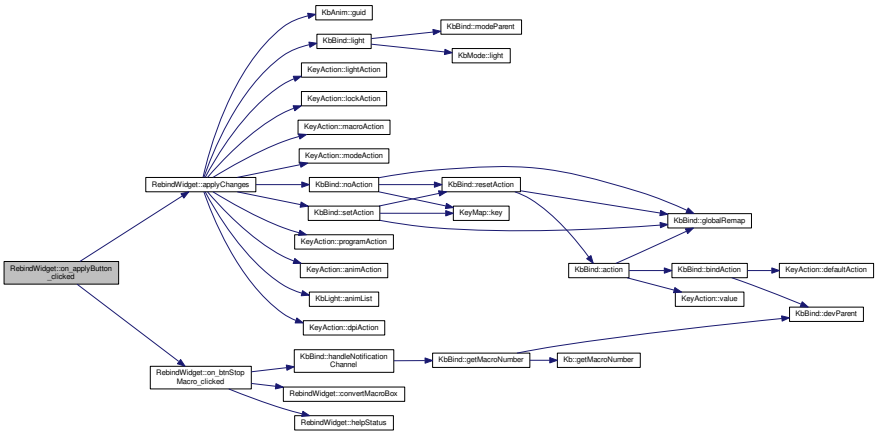
12.81.3.6 void RebindWidget::on_applyButton_clicked ( ) [private], [slot]

Definition at line 406 of file rebindwidget.cpp.

References applyChanges(), on_btnStopMacro_clicked(), and selection.

```
406                                         {
407     // Normally, this should be done via signalling.
408     // Because there is no serarate thread, we have to call it directly
409     // (otherwise we could do Key char conversion step by step,
410     // but so it is more easy to change the key definition):
411     on_btnStopMacro_clicked();
412     applyChanges(selection, true);
413 }
```

Here is the call graph for this function:



12.81.3.7 void RebindWidget::on_btnClearMacro_clicked ( ) [private], [slot]

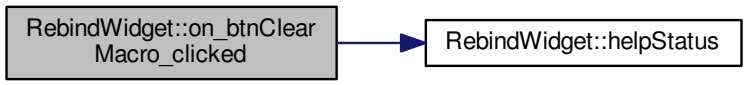
Todo I do not know what is the better solution with the delay-buttons in case of clicking clear: Reset the button to the default value or do not touch it? Not clear is ignored.

Definition at line 774 of file rebindwidget.cpp.

References helpStatus().

```
774                                             {
775     helpStatus(1);
776 }
```

Here is the call graph for this function:



12.81.3.8 void RebindWidget::on_btnStartMacro_clicked ( ) [private],[slot]

The UI is protected against false clicking (e.g. if you type start and than Apply, the channel is closed in wrong order).

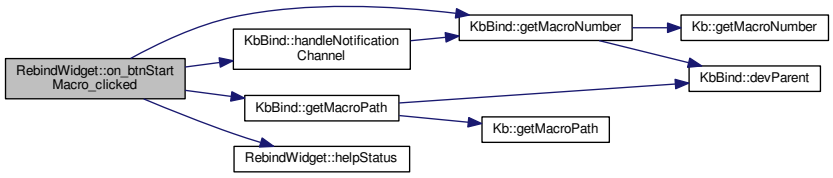
At this time, all neccessary params like macroNumber, macroPath, cmdFile etc. had been cached.

Definition at line 724 of file rebindwidget.cpp.

References Ui_RebindWidget::applyButton, bind, Ui_RebindWidget::btnStartMacro, Ui_RebindWidget::btnStopMacro, KbBind::getMacroNumber(), KbBind::getMacroPath(), KbBind::handleNotificationChannel(), helpStatus(), macReader, Ui_RebindWidget::pteMacroBox, Ui_RebindWidget::pteMacroText, Ui_RebindWidget::rb_delay_asTyped, Ui_RebindWidget::rb_delay_default, Ui_RebindWidget::rb_delay_no, Ui_RebindWidget::resetButton, ui, and Ui_RebindWidget::unbindButton.

```
724                                             {
725     if (!macReader) {
726         bind->handleNotificationChannel(true);
727         macReader = new MacroReader(bind->getMacroNumber(),
      bind->getMacroPath(), ui->pteMacroBox, ui->
      pteMacroText);
728         // because of the second thread we need to disable three of the four bottom buttons.
729         // Clicking "Stop" will enable them again.
730         ui->applyButton->setEnabled(false);
731         ui->resetButton->setEnabled(false);
732         ui->unbindButton->setEnabled(false);
733         ui->btnStartMacro->setEnabled(false);
734         ui->btnStopMacro->setEnabled(true);
735         ui->rb_delay_asTyped->setEnabled(false);
736         ui->rb_delay_no->setEnabled(false);
737         ui->rb_delay_default->setEnabled(false);
738         helpStatus(2);
739     }
740 }
```

Here is the call graph for this function:



12.81.3.9 void RebindWidget::on_btnStopMacro_clicked ( ) [private],[slot]

Afterwards, the characters in the MacroBox are changed from KB-out format to cmd-in format. At last the UI changes to the new state.

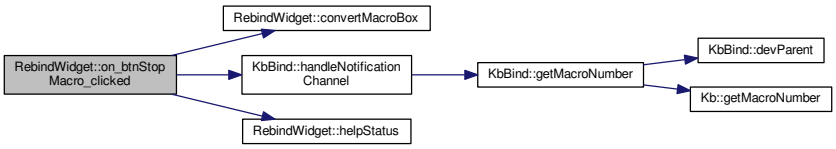
Definition at line 749 of file rebindwidget.cpp.

References Ui_RebindWidget::applyButton, bind, Ui_RebindWidget::btnStartMacro, Ui_RebindWidget::btnStopMacro, convertMacroBox(), KbBind::handleNotificationChannel(), helpStatus(), macReader, Ui_RebindWidget::rb_delay_asTyped, Ui_RebindWidget::rb_delay_default, Ui_RebindWidget::rb_delay_no, Ui_RebindWidget::resetButton, ui, and Ui_RebindWidget::unbindButton.

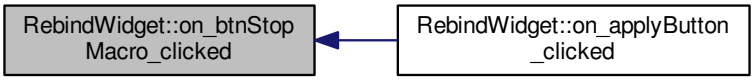
Referenced by on_applyButton_clicked().

```
749                                         {
750     if (macReader) {
751         bind->handleNotificationChannel(false);
752         delete macReader;
753         macReader = 0;
754         convertMacroBox();
755         ui->applyButton->setEnabled(true);
756         ui->resetButton->setEnabled(true);
757         ui->unbindButton->setEnabled(true);
758         ui->btnStartMacro->setEnabled(true);
759         ui->btnStopMacro->setEnabled(false);
760         ui->rb_delay_asTyped->setEnabled(true);
761         ui->rb_delay_no->setEnabled(true);
762         ui->rb_delay_default->setEnabled(true);
763         helpStatus(3);
764     }
765 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



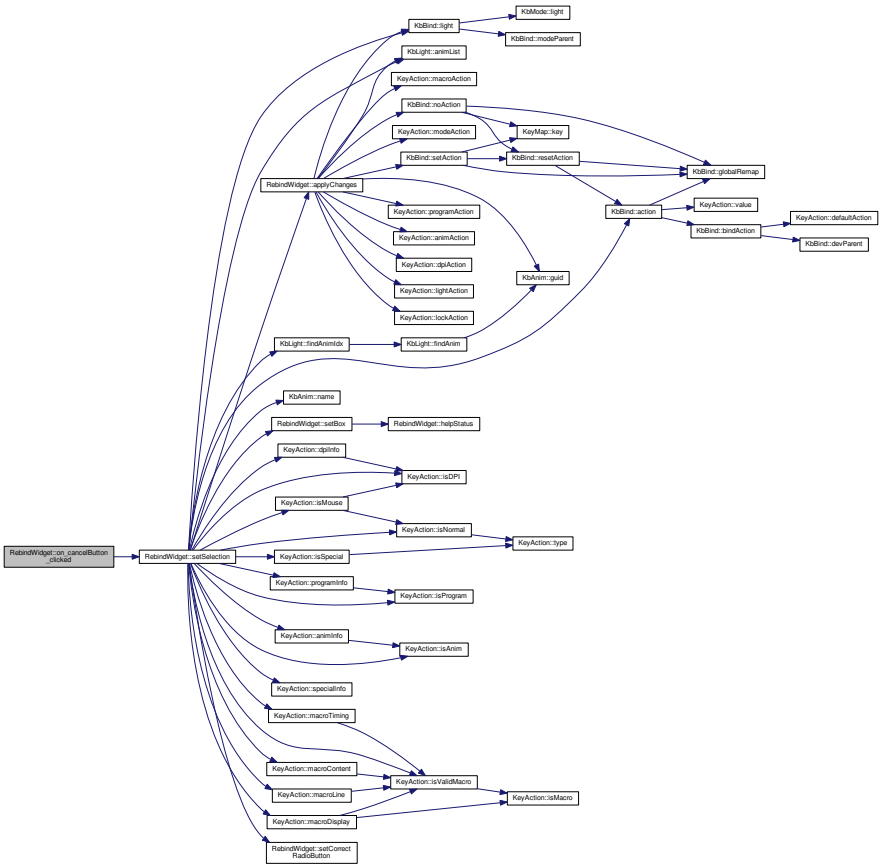
12.81.3.10 void RebindWidget::on_cancelButton_clicked ( ) [private], [slot]

Definition at line 415 of file rebindwidget.cpp.

References selection, and setSelection().

```
415                                         {
416     // Re-load selection
417     setSelection(selection);
418 }
```

Here is the call graph for this function:



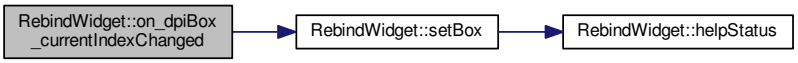
12.81.3.11 void RebindWidget::on_dpiBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 539 of file rebindwidget.cpp.

References DPI_CUST_IDX, Ui_RebindWidget::dpiBox, Ui_RebindWidget::dpiButton, Ui_RebindWidget::dpiCustLabel, Ui_RebindWidget::dpiCustXBox, Ui_RebindWidget::dpiCustYBox, setBox(), and ui.

```
539                                                     {
540     if(index != DPI_CUST_IDX){
541         // Clear custom DPI
542         ui->dpiCustXBox->setEnabled(false);
543         ui->dpiCustYBox->setEnabled(false);
544         ui->dpiCustLabel->setEnabled(false);
545     } else {
546         ui->dpiCustXBox->setEnabled(true);
547         ui->dpiCustYBox->setEnabled(true);
548         ui->dpiCustLabel->setEnabled(true);
549     }
550     if(index == 0)
551         ui->dpiButton->setChecked(false);
552     else {
553         ui->dpiButton->setChecked(true);
554         setBox(ui->dpiBox);
555     }
556 }
```

Here is the call graph for this function:



12.81.3.12 void RebindWidget::on_dpiButton_clicked (bool *checked*) [private], [slot]

Definition at line 695 of file rebindwidget.cpp.

References Ui_RebindWidget::dpiBox, and ui.

```
695                                                 {
696     if(checked && ui->dpiBox->currentIndex() == 0)
697         ui->dpiBox->setCurrentIndex(1);
698 }
```

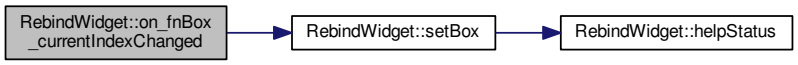
12.81.3.13 void RebindWidget::on_fnBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 485 of file rebindwidget.cpp.

References Ui_RebindWidget::fnBox, Ui_RebindWidget::fnButton, setBox(), and ui.

```
485                                                     {
486     if(index == 0)
487         ui->fnButton->setChecked(false);
488     else {
489         ui->fnButton->setChecked(true);
490         setBox(ui->fnBox);
491     }
492 }
```

Here is the call graph for this function:



12.81.3.14 void RebindWidget::on_fnButton_clicked (bool *checked*) [private], [slot]

Definition at line 632 of file rebindwidget.cpp.

References Ui_RebindWidget::fnBox, and ui.

```
632                                                 {
633     if(checked && ui->fnBox->currentIndex() == 0)
634         ui->fnBox->setCurrentIndex(1);
635 }
```

12.81.3.15 void RebindWidget::on_lightBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 571 of file rebindwidget.cpp.

References Ui_RebindWidget::lightBox, Ui_RebindWidget::lightButton, Ui_RebindWidget::lightWrapBox, setBox(), and ui.

```
571                                                     {
572     if(index == 0){
573         ui->lightButton->setChecked(false);
574         ui->lightWrapBox->hide();
575     } else {
576         ui->lightWrapBox->show();
577         ui->lightButton->setChecked(true);
578         setBox(ui->lightBox);
579     }
580 }
```

Here is the call graph for this function:

12.81.3.16 void RebindWidget::on_lightButton_clicked (bool *checked*) [private], [slot]

Definition at line 652 of file rebindwidget.cpp.

References Ui_RebindWidget::lightBox, and ui.

```
652                                                 {
653     if(checked && ui->lightBox->currentIndex() == 0)
654         ui->lightBox->setCurrentIndex(1);
655 }
```

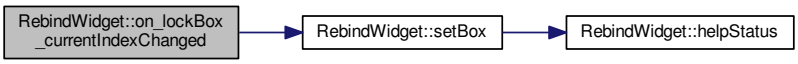
12.81.3.17 void RebindWidget::on_lockBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 582 of file rebindwidget.cpp.

References Ui_RebindWidget::lockBox, Ui_RebindWidget::lockButton, setBox(), and ui.

```
582                                                     {
583     if(index == 0)
584         ui->lockButton->setChecked(false);
585     else {
586         ui->lockButton->setChecked(true);
587         setBox(ui->lockBox);
588     }
589 }
```

Here is the call graph for this function:



12.81.3.18 void RebindWidget::on_lockButton_clicked (bool *checked*) [private], [slot]

Definition at line 657 of file rebindwidget.cpp.

References Ui_RebindWidget::lockBox, and ui.

```
657                                                 {
658     if(checked && ui->lockBox->currentIndex() == 0)
659         ui->lockBox->setCurrentIndex(1);
660 }
```

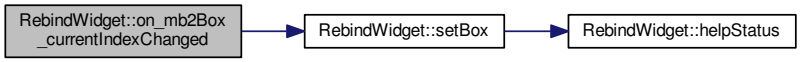
12.81.3.19 void RebindWidget::on_mb2Box_currentIndexChanged (int *index*) [private], [slot]

Definition at line 521 of file rebindwidget.cpp.

References Ui_RebindWidget::mb2Box, Ui_RebindWidget::mb2Button, setBox(), and ui.

```
521                                                     {
522     if(index == 0)
523         ui->mb2Button->setChecked(false);
524     else {
525         ui->mb2Button->setChecked(true);
526         setBox(ui->mb2Box);
527     }
528 }
```

Here is the call graph for this function:



12.81.3.20 void RebindWidget::on_mb2Button_clicked (bool *checked*) [private], [slot]

Definition at line 685 of file rebindwidget.cpp.

References Ui_RebindWidget::mb2Box, and ui.

```
685                                                 {
686     if(checked && ui->mb2Box->currentIndex() == 0)
687         ui->mb2Box->setCurrentIndex(1);
688 }
```

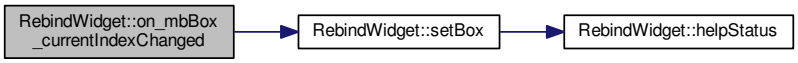
12.81.3.21 void RebindWidget::on_mbBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 512 of file rebindwidget.cpp.

References Ui_RebindWidget::mbBox, Ui_RebindWidget::mbButton, setBox(), and ui.

```
512                                                     {
513     if(index == 0)
514         ui->mbButton->setChecked(false);
515     else {
516         ui->mbButton->setChecked(true);
517         setBox(ui->mbBox);
518     }
519 }
```

Here is the call graph for this function:



12.81.3.22 void RebindWidget::on_mbButton_clicked (bool *checked*) [private], [slot]

Definition at line 680 of file rebindwidget.cpp.

References Ui_RebindWidget::mbBox, and ui.

```
680                                                 {
681     if(checked && ui->mbBox->currentIndex() == 0)
682         ui->mbBox->setCurrentIndex(1);
683 }
```

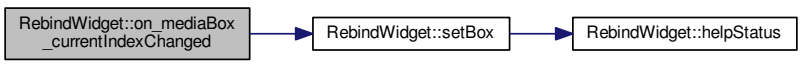
12.81.3.23 void RebindWidget::on_mediaBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 503 of file rebindwidget.cpp.

References Ui_RebindWidget::mediaBox, Ui_RebindWidget::mediaButton, setBox(), and ui.

```
503                                                     {
504     if(index == 0)
505         ui->mediaButton->setChecked(false);
506     else {
507         ui->mediaButton->setChecked(true);
508         setBox(ui->mediaBox);
509     }
510 }
```

Here is the call graph for this function:



12.81.3.24 void RebindWidget::on_mediaButton_clicked (bool *checked*) [private], [slot]

Definition at line 642 of file rebindwidget.cpp.

References Ui_RebindWidget::mediaBox, and ui.

```
642                                                 {
643     if(checked && ui->mediaBox->currentIndex() == 0)
644         ui->mediaBox->setCurrentIndex(1);
645 }
```

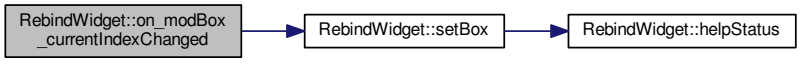
12.81.3.25 void RebindWidget::on_modBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 476 of file rebindwidget.cpp.

References Ui_RebindWidget::modBox, Ui_RebindWidget::modButton, setBox(), and ui.

```
476                                                     {
477     if(index == 0)
478         ui->modButton->setChecked(false);
479     else {
480         ui->modButton->setChecked(true);
481         setBox(ui->modBox);
482     }
483 }
```

Here is the call graph for this function:



12.81.3.26 void RebindWidget::on_modButton_clicked (bool *checked*) [private], [slot]

Definition at line 627 of file rebindwidget.cpp.

References Ui_RebindWidget::modBox, and ui.

```
627                                                 {
628     if(checked && ui->modBox->currentIndex() == 0)
629         ui->modBox->setCurrentIndex(1);
630 }
```

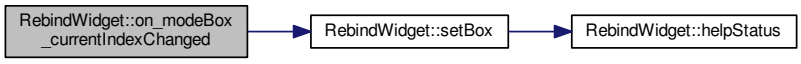
12.81.3.27 void RebindWidget::on_modeBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 558 of file rebindwidget.cpp.

References Ui_RebindWidget::modeBox, Ui_RebindWidget::modeButton, Ui_RebindWidget::modeWrapBox, setBox(), and ui.

```
558                                                     {
559     if(index == 1 || index == 2)
560         ui->modeWrapBox->show();
561     else
562         ui->modeWrapBox->hide();
563     if(index == 0)
564         ui->modeButton->setChecked(false);
565     else {
566         ui->modeButton->setChecked(true);
567         setBox(ui->modeBox);
568     }
569 }
```

Here is the call graph for this function:



12.81.3.28 void RebindWidget::on_modeButton_clicked (bool *checked*) [private], [slot]

Definition at line 647 of file rebindwidget.cpp.

References Ui_RebindWidget::modeBox, and ui.

```
647                                                 {
648     if(checked && ui->modeBox->currentIndex() == 0)
649         ui->modeBox->setCurrentIndex(1);
650 }
```

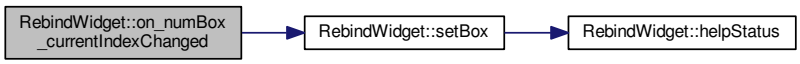
12.81.3.29 void RebindWidget::on_numBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 494 of file rebindwidget.cpp.

References Ui_RebindWidget::numBox, Ui_RebindWidget::numButton, setBox(), and ui.

```
494                                                     {
495     if(index == 0)
496         ui->numButton->setChecked(false);
497     else {
498         ui->numButton->setChecked(true);
499         setBox(ui->numBox);
500     }
501 }
```

Here is the call graph for this function:



12.81.3.30 void RebindWidget::on_numButton_clicked (bool *checked*) [private], [slot]

Definition at line 637 of file rebindwidget.cpp.

References Ui_RebindWidget::numBox, and ui.

```
637                                                 {
638     if(checked && ui->numBox->currentIndex() == 0)
639         ui->numBox->setCurrentIndex(1);
640 }
```

12.81.3.31 void RebindWidget::on_programKpBox_textChanged (const QString & *arg1*) [private], [slot]

Definition at line 591 of file rebindwidget.cpp.

References Ui_RebindWidget::programKpBox, Ui_RebindWidget::programKpButton, Ui_RebindWidget::programKpExtra, setBox(), and ui.

```
591                                                         {
592     if(arg1.isEmpty()){
593         ui->programKpButton->setChecked(false);
594         ui->programKpExtra->hide();
595     } else {
596         ui->programKpButton->setChecked(true);
597         setBox(ui->programKpBox);
598         ui->programKpExtra->show();
599     }
600 }
```

Here is the call graph for this function:

12.81.3.32 void RebindWidget::on_programKpButton_clicked (bool *checked*) [private], [slot]

Definition at line 662 of file rebindwidget.cpp.

References Ui_RebindWidget::programKpBox, setBox(), and ui.

```
662                                                     {
663     if(!checked){
664         ui->programKpBox->setText("");
665     } else if(ui->programKpBox->text().isEmpty()){
666         ui->programKpBox->setFocus();
667         setBox(ui->programKpBox);
668     }
669 }
```

Here is the call graph for this function:

12.81.3.33 void RebindWidget::on_programKpSlBox_clicked (bool *checked*) [private], [slot]

Definition at line 700 of file rebindwidget.cpp.

References Ui_RebindWidget::programKpModeBox, and ui.

```
700                                                     {
701     ui->programKpModeBox->setCurrentIndex(0);
702     ui->programKpModeBox->setEnabled(checked);
703 }
```

12.81.3.34 void RebindWidget::on_programKrBox_textChanged (const QString & *arg1*) [private], [slot]

Definition at line 602 of file rebindwidget.cpp.

References Ui_RebindWidget::programKrBox, Ui_RebindWidget::programKrButton, Ui_RebindWidget::programKrExtra, setBox(), and ui.

```
602                                                     {
603     if(arg1.isEmpty()){
604         ui->programKrButton->setChecked(false);
605         ui->programKrExtra->hide();
606     } else {
607         ui->programKrButton->setChecked(true);
608         setBox(ui->programKrBox);
609         ui->programKrExtra->show();
610     }
611 }
```

Here is the call graph for this function:

12.81.3.35 void RebindWidget::on_programKrButton_clicked (bool *checked*) [private], [slot]

Definition at line 671 of file rebindwidget.cpp.

References Ui_RebindWidget::programKrBox, setBox(), and ui.

```
671                                                     {
672     if(!checked){
673         ui->programKrBox->setText("");
674     } else if(ui->programKrBox->text().isEmpty()){
675         ui->programKrBox->setFocus();
676         setBox(ui->programKrBox);
677     }
678 }
```

Here is the call graph for this function:

12.81.3.36 void RebindWidget::on_programKrSlBox_clicked (bool *checked*) [private], [slot]

Definition at line 705 of file rebindwidget.cpp.

References Ui_RebindWidget::programKrModeBox, and ui.

```
705                                                     {
706     ui->programKrModeBox->setCurrentIndex(0);
707     ui->programKrModeBox->setEnabled(checked);
708 }
```

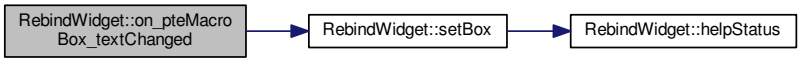
12.81.3.37 void RebindWidget::on_pteMacroBox_textChanged ( ) [private], [slot]

Definition at line 470 of file rebindwidget.cpp.

References Ui_RebindWidget::pteMacroBox, setBox(), and ui.

```
470                                             {
471     if (ui->pteMacroBox->toPlainText().length() > 0) {
472         setBox(ui->pteMacroBox);
473     }
474 }
```

Here is the call graph for this function:



12.81.3.38 void RebindWidget::on_rb_delay_asTyped_toggled (bool *checked*) [private], [slot]

Parameters

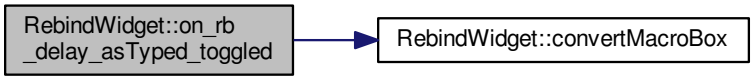
checked	This button ist clicked to use the delay times, as they are recorded. Returs a warning message, if we are not in the recording phase, because then we don't have the delay times any more.

Definition at line 862 of file rebindwidget.cpp.

References convertMacroBox().

```
863 {
864     convertMacroBox();
865 }
```

Here is the call graph for this function:



12.81.3.39 void RebindWidget::on_rb_delay_default_toggled (bool *checked*) [private], [slot]

Parameters

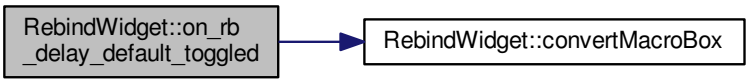
checked	This is as easy as the no-delay-button, because this means take the default values.

Definition at line 873 of file rebindwidget.cpp.

References convertMacroBox().

```
874 {
875     convertMacroBox();
876 }
```

Here is the call graph for this function:



12.81.3.40 void RebindWidget::on_rb_delay_no_toggled (bool *checked*) [private], [slot]

Parameters

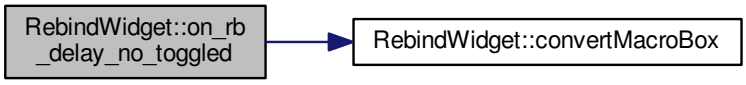
checked	The following slots are triggerd by changing the mutual exclusive radio buttons when choosing the delay. They are called, if the button ist enabled. This first one should disable all delay.

Definition at line 850 of file rebindwidget.cpp.

References convertMacroBox().

```
851 {
852     convertMacroBox();
853 }
```

Here is the call graph for this function:



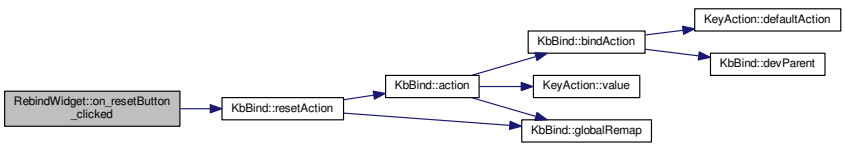
12.81.3.41 void RebindWidget::on_resetButton_clicked ( ) [private], [slot]

Definition at line 420 of file rebindwidget.cpp.

References bind, KbBind::resetAction(), and selection.

```
420                                     {
421     bind->resetAction(selection);
422 }
```

Here is the call graph for this function:



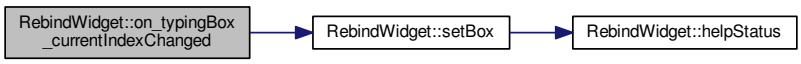
12.81.3.42 void RebindWidget::on_typingBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 461 of file rebindwidget.cpp.

References setBox(), Ui_RebindWidget::typingBox, Ui_RebindWidget::typingButton, and ui.

```
461                                                         {
462     if(index == 0)
463         ui->typingButton->setChecked(false);
464     else {
465         ui->typingButton->setChecked(true);
466         setBox(ui->typingBox);
467     }
468 }
```

Here is the call graph for this function:



12.81.3.43 void RebindWidget::on_typingButton_clicked (bool *checked*) [private], [slot]

Definition at line 622 of file rebindwidget.cpp.

References Ui_RebindWidget::typingBox, and ui.

```
622                                                 {
623     if(checked && ui->typingBox->currentIndex() == 0)
624         ui->typingBox->setCurrentIndex(1);
625 }
```

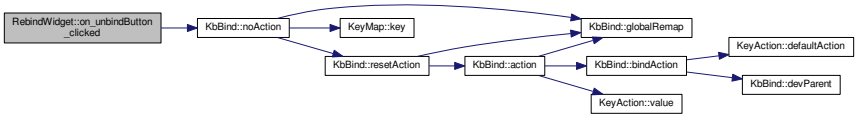
12.81.3.44 void RebindWidget::on_unbindButton_clicked ( ) [private], [slot]

Definition at line 424 of file rebindwidget.cpp.

References bind, KbBind::noAction(), and selection.

```
424                                             {
425     bind->noAction(selection);
426 }
```

Here is the call graph for this function:



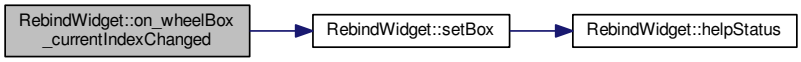
12.81.3.45 void RebindWidget::on_wheelBox_currentIndexChanged (int *index*) [private], [slot]

Definition at line 530 of file rebindwidget.cpp.

References setBox(), ui, Ui_RebindWidget::wheelBox, and Ui_RebindWidget::wheelButton.

```
530                                                          {
531     if(index == 0)
532         ui->wheelButton->setChecked(false);
533     else {
534         ui->wheelButton->setChecked(true);
535         setBox(ui->wheelBox);
536     }
537 }
```

Here is the call graph for this function:



12.81.3.46 void RebindWidget::on_wheelButton_clicked (bool *checked*) [private], [slot]

Definition at line 690 of file rebindwidget.cpp.

References ui, and Ui_RebindWidget::wheelBox.

```
690                                                      {
691     if(checked && ui->wheelBox->currentIndex() == 0)
692         ui->wheelBox->setCurrentIndex(1);
693 }
```

12.81.3.47 void RebindWidget::setBind (KbBind * *newBind*, KbProfile * *newProfile*)

Definition at line 63 of file rebindwidget.cpp.

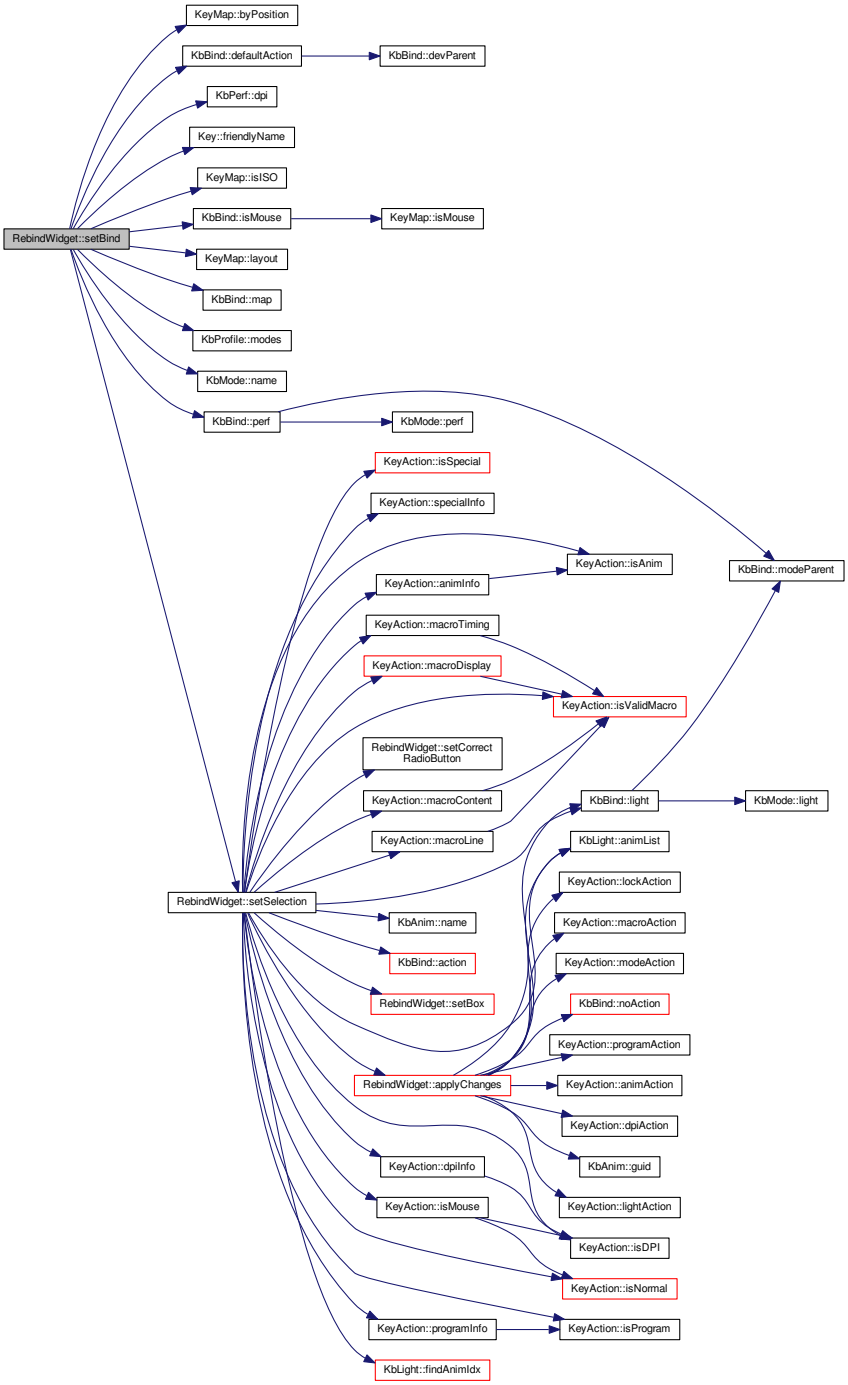
References bind, KeyMap::byPosition(), KbBind::defaultAction(), KbPerf::dpi(), KbPerf::DPI_COUNT, DPI_OFFSET, Ui_RebindWidget::dpiBox, Ui_RebindWidget::dpiButton, Ui_RebindWidget::dpiCustLabel, Ui_RebindWidget::dpiCustXBox, Ui_RebindWidget::dpiCustYBox, Ui_RebindWidget::dpiWarning, fnKeys, Key::friendlyName(), KeyMap::isISO(), KbBind::isMouse(), KeyMap::K95, KeyMap::layout(), KbBind::map(), mediaKeys, Ui_RebindWidget::modeBox, KbProfile::modes(), modKeys, KbMode::name(), numKeys, KbBind::perf(), profile, setSelection(), Ui_RebindWidget::typingBox, typingKeys, and ui.

Referenced by KbBindWidget::newLayout(), and KbBindWidget::setBind().

```
63                                                                {
64     bind = newBind;
65     profile = newProfile;
66     setSelection(QStringList());
67
68     // Populate typing keys by position
69     ui->typingBox->clear();
70     ui->typingBox->addItem(" ");
71     typingKeys.clear();
72     // Use the K95 map as it has all keys
73     const KeyMap& map = KeyMap(KeyMap::K95, bind->map().
       layout());
74     foreach(const QString& name, map.byPosition()){
75         KeyAction action(bind->defaultAction(name));
76         if(action.isNormal() && !modKeys.contains(name) && !fnKeys.contains(name) && !
       numKeys.contains(name) && !mediaKeys.contains(name) && name != "enter" && name != "tab" &&
       name != "bspace"){
```

```
77              const Key& pos = map[name];
78              QString friendly = pos.friendlyName();
79              ui->typingBox->addItem(friendly);
80              typingKeys.append(name);
81          }
82      }
83      typingKeys << "enter" << "tab" << "bspace";
84      ui->typingBox->addItem("Enter");
85      ui->typingBox->addItem("Tab");
86      ui->typingBox->addItem("Backspace");
87      if(!map.isISO()){
88          // Add ISO backslash (aka KEY_102ND) to ANSI options
89          typingKeys << "bslash_iso";
90          ui->typingBox->addItem("Backslash (ISO layout)");
91      }
92
93      // Populate mode list
94      ui->modeBox->clear();
95      ui->modeBox->addItem(" ");
96      ui->modeBox->addItem("(Previous)");
97      ui->modeBox->addItem("(Next)");
98      int idx = 1;
99      foreach(KbMode* mode, newProfile->modes())
100         ui->modeBox->addItem(QString("%1: %2").arg(idx++).arg(mode->
    name()));
101
102     // Enable/disable DPI based on device
103     if(bind->isMouse()){
104         ui->dpiButton->setEnabled(true);
105         ui->dpiBox->setEnabled(true);
106         ui->dpiWarning->hide();
107         // Fill DPI slots
108         const KbPerf* perf = bind->perf();
109         for(int i = 0; i < KbPerf::DPI_COUNT; i++){
110             bool sniper = (i == 0);
111             int boxIdx = i + DPI_OFFSET;
112             QPoint dpi = perf->dpi(i);
113             QString text = tr(sniper ? "Sniper:\t%1 x %2" : "%3:\t%1 x %2").arg(dpi.x()).arg(dpi.y());
114             if(!sniper) text = text.arg(i);
115             ui->dpiBox->setItemText(boxIdx, text);
116         }
117     } else {
118         ui->dpiButton->setEnabled(false);
119         ui->dpiBox->setEnabled(false);
120         ui->dpiWarning->show();
121     }
122     // Always disable custom DPI boxes until selected
123     ui->dpiCustXBox->setEnabled(false);
124     ui->dpiCustYBox->setEnabled(false);
125     ui->dpiCustLabel->setEnabled(false);
126 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.81.3.48 void RebindWidget::setBox (QWidget * *box*) [private]

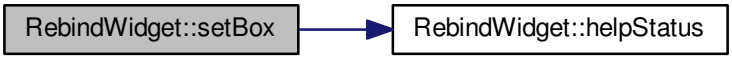
Definition at line 428 of file rebindwidget.cpp.

References Ui_RebindWidget::animBox, Ui_RebindWidget::dpiBox, Ui_RebindWidget::fnBox, helpStatus(), Ui_RebindWidget::lightBox, Ui_RebindWidget::lockBox, Ui_RebindWidget::mb2Box, Ui_RebindWidget::mbBox, Ui_RebindWidget::mediaBox, Ui_RebindWidget::modBox, Ui_RebindWidget::modeBox, Ui_RebindWidget::numBox, Ui_RebindWidget::programKpBox, Ui_RebindWidget::programKpButton, Ui_RebindWidget::programKrBox, Ui_RebindWidget::programKrButton, Ui_RebindWidget::pteMacroBox, Ui_RebindWidget::txtBuffer, Ui_RebindWidget::typingBox, ui, and Ui_RebindWidget::wheelBox.

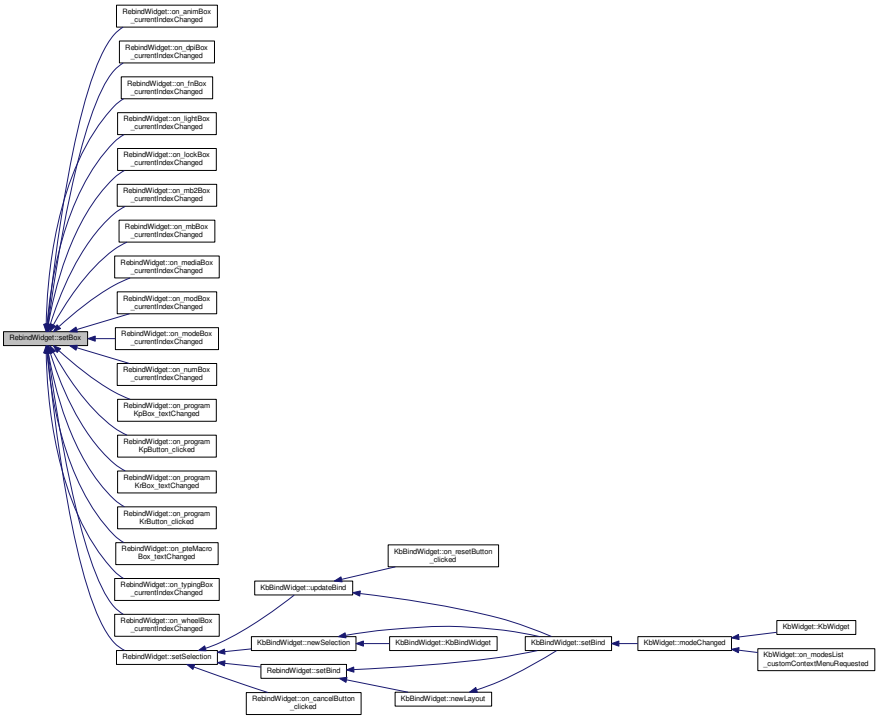
Referenced by on_animBox_currentIndexChanged(), on_dpiBox_currentIndexChanged(), on_fnBox_currentIndexChanged(), on_lightBox_currentIndexChanged(), on_lockBox_currentIndexChanged(), on_mb2Box_currentIndexChanged(), on_mbBox_currentIndexChanged(), on_mediaBox_currentIndexChanged(), on_modBox_currentIndexChanged(), on_modeBox_currentIndexChanged(), on_numBox_currentIndexChanged(), on_programKpBox_textChanged(), on_programKpButton_clicked(), on_programKrBox_textChanged(), on_programKrButton_clicked(), on_pteMacroBox_textChanged(), on_typingBox_currentIndexChanged(), on_wheelBox_currentIndexChanged(), and setSelection().

```
428                                     {
429     // Un-select every item except for the current one.
430     // on_*_currentIndexChanged will take care of deselecting the checkbox.
431     // Key
432     if(box != ui->typingBox) ui->typingBox->setCurrentIndex(0);
433     if(box != ui->modBox) ui->modBox->setCurrentIndex(0);
434     if(box != ui->fnBox) ui->fnBox->setCurrentIndex(0);
435     if(box != ui->numBox) ui->numBox->setCurrentIndex(0);
436     if(box != ui->mediaBox) ui->mediaBox->setCurrentIndex(0);
437     // Mouse
438     if(box != ui->mbBox) ui->mbBox->setCurrentIndex(0);
439     if(box != ui->mb2Box) ui->mb2Box->setCurrentIndex(0);
440     if(box != ui->wheelBox) ui->wheelBox->setCurrentIndex(0);
441     if(box != ui->dpiBox) ui->dpiBox->setCurrentIndex(0);
442     // Anim
443     if(box != ui->animBox) ui->animBox->setCurrentIndex(0);
444     // Special
445     if(box != ui->modeBox) ui->modeBox->setCurrentIndex(0);
446     if(box != ui->lightBox) ui->lightBox->setCurrentIndex(0);
447     if(box != ui->lockBox) ui->lockBox->setCurrentIndex(0);
448     // Program
449     if(box != ui->programKpBox && box != ui->programKrBox){
450         ui->programKpButton->setChecked(false);
451         ui->programKrButton->setChecked(false);
452     }
453     // Clear macro panel
454     if (box != ui->pteMacroBox) {
455         ui->pteMacroBox->setPlainText("");
456         ui->txtBuffer->setText("");
457         helpStatus(1);
458     }
459 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.81.3.49 void RebindWidget::setCorrectRadioButton (QString *macdef*) [private]

Parameters

macdef	Set the radiobutton for timing paramters according to the context. If no "=" followed by a number and comma can be found, it is the default button. If "=" can be found and numbers with more than one digit (means: $>$ 9), it is the "asTyped" button Otherwise it is the "no" button.

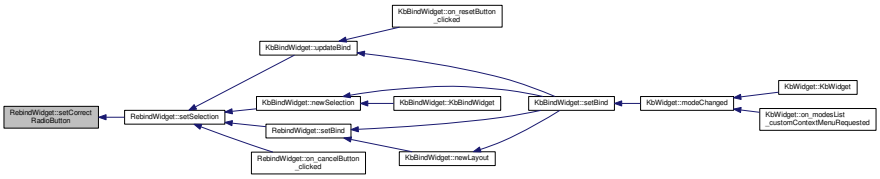
Definition at line 887 of file rebindwidget.cpp.

References Ui_RebindWidget::rb_delay_asTyped, Ui_RebindWidget::rb_delay_default, Ui_RebindWidget::rb_delay_no, and ui.

Referenced by setSelection().

```
887                                                     {
888     if (!macdef.contains(QRegExp("=\\d+,"))) {
889         ui->rb_delay_default->setChecked(true);
890         return;
891     }
892     if (macdef.contains(QRegExp("=\\d\\d+,"))) {
893         ui->rb_delay_asTyped->setChecked(true);
894         return;
895     }
896     ui->rb_delay_no->setChecked(true);
897 }
```

Here is the caller graph for this function:



12.81.3.50 void RebindWidget::setSelection (const QStringList & *newSelection,* bool *applyPrevious* = `false`)

Definition at line 128 of file rebindwidget.cpp.

References KbBind::action(), anim, Ui_RebindWidget::animBox, KeyAction::animInfo(), Ui_RebindWidget::animKrBox, KbLight::animList(), Ui_RebindWidget::animOnceBox, applyChanges(), bind, KeyAction::DPI_CUSTOM, DPI_OFFSET, Ui_RebindWidget::dpiBox, Ui_RebindWidget::dpiCustXBox, Ui_RebindWidget::dpiCustYBox, KeyAction::dpiInfo(), KbLight::findAnimIdx(), Ui_RebindWidget::fnBox, fnKeys, KeyAction::isAnim(), KeyAction::isDPI(), KeyAction::isMouse(), KeyAction::isNormal(), KeyAction::isProgram(), KeyAction::isSpecial(), KeyAction::isValidMacro(), KbBind::light(), KeyAction::LIGHT_DOWN, KeyAction::LIGHT_DOWN_WRAP, KeyAction::LIGHT_UP, KeyAction::LIGHT_UP_WRAP, Ui_RebindWidget::lightBox, Ui_RebindWidget::lightWrapBox, Ui_RebindWidget::lockBox, KeyAction::macroContent(), KeyAction::macroDisplay(), KeyAction::macroLine(), KeyAction::macroTiming(), Ui_RebindWidget::mb2Box, Ui_RebindWidget::mbBox, Ui_RebindWidget::mediaBox, mediaKeys, Ui_RebindWidget::modBox, KeyAction::MODE_NEXT, KeyAction::MODE_NEXT_WRAP, KeyAction::MODE_PREV, KeyAction::MODE_PREV_WRAP, Ui_RebindWidget::modeBox, Ui_RebindWidget::modeWrapBox, modKeys, mouseExtKeys, mouseKeys, KbAnim::name(), Ui_RebindWidget::numBox, numKeys, KeyAction::PROGRAM_PR_MULTI, KeyAction::PROGRAM_RE_INDEF, KeyAction::PROGRAM_RE_KPSTOP, KeyAction::PROGRAM_RE_MULTI, KeyAction::programInfo(), Ui_RebindWidget::programKpBox, Ui_RebindWidget::programKpModeBox, Ui_RebindWidget::programKpSIBox, Ui_RebindWidget::programKrBox, Ui_RebindWidget::programKrModeBox, Ui_RebindWidget::programKrSIBox, Ui_RebindWidget::pteMacroBox, Ui_RebindWidget::pteMacroComment, Ui_RebindWidget::pteMacroText, selection, setBox(), setCorrectRadioButton(), KeyAction::specialInfo(), TAB_ANIM, TAB_KB, TAB_MACRO, TAB_MOUSE, TAB_PROGRAM, TAB_SPECIAL, Ui_RebindWidget::tabWidget, Ui_RebindWidget::txtBuffer, Ui_RebindWidget::typingBox, typingKeys, ui, Ui_RebindWidget::wheelBox, and wheelKeys.

Referenced by KbBindWidget::newSelection(), on_cancelButton_clicked(), setBind(), and KbBindWidget::updateBind().

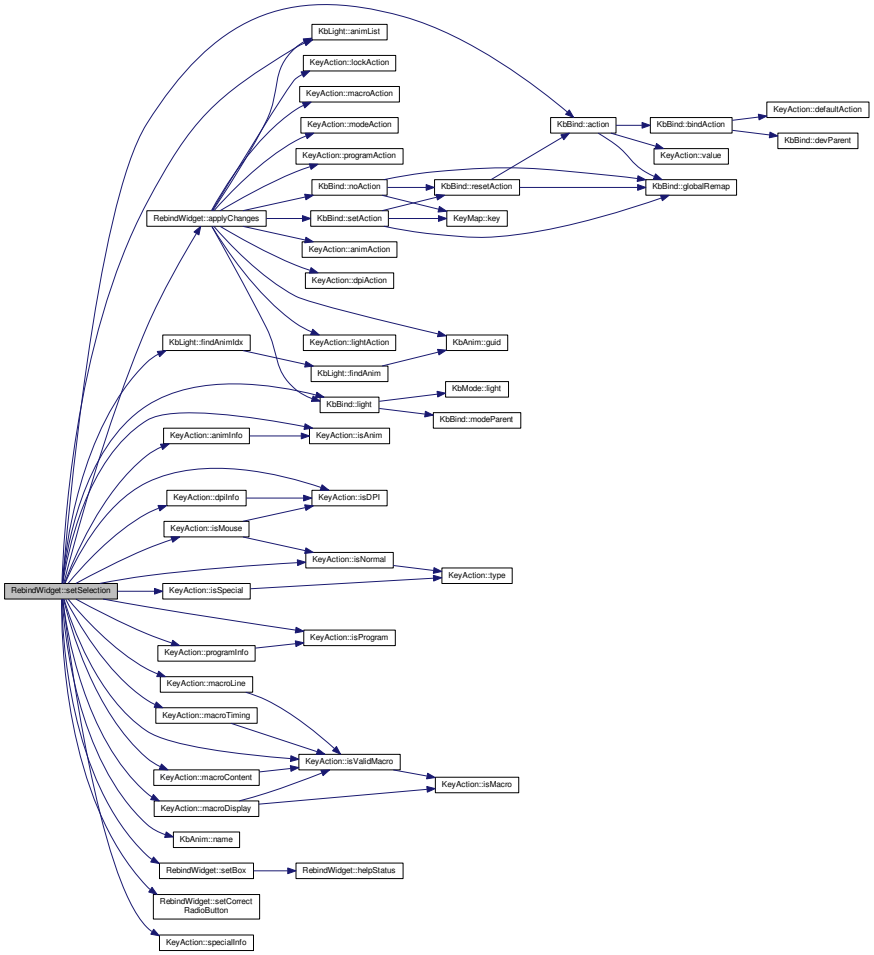
```
128                                                                                 {
129     if(bind){
130         // Populate animation list
131         // FIXME: There should be a trigger to update this whenever an animation is added/removed, even if
        the key selection hasn't changed
132         ui->animBox->clear();
133         ui->animBox->addItem(" ");
134         foreach(KbAnim* anim, bind->light()->animList())
135             ui->animBox->addItem(anim->name());
136     }
137
138     // Apply changes to previous selection (if any)
139     if(!selection.isEmpty() && applyPrevious){
140         QStringList previous = selection;
141         selection = QStringList();
142         applyChanges(previous, false);
143     }
144
145     selection = newSelection;
146     if(newSelection.isEmpty()){
147         hide();
148         return;
149     } else
150         show();
151
152     bool hasAction = false;
153     QString action;
154     if(selection.count() == 1){
```

```
155         // Select action from the UI
156         action = bind->action(selection.first());
157         if(!action.isEmpty())
158             hasAction = true;
159     } else {
160         // Select an action from the UI if all keys are the same.
161         // Select nothing if any are unbound or different
162         foreach(const QString& key, selection){
163             QString keyAction = bind->action(key);
164             if(keyAction.isEmpty() || (hasAction && keyAction != action)){
165                 hasAction = false;
166                 break;
167             } else if(action.isEmpty()){
168                 action = keyAction;
169                 hasAction = true;
170             }
171         }
172     }
173
174     if(!hasAction)
175         action = "";
176     KeyAction act(action);
177     // Clear everything
178     setBox(0);
179     ui->dpiCustXBox->setValue(400);
180     ui->dpiCustYBox->setValue(400);
181     ui->animOnceBox->setChecked(false);
182     ui->animKrBox->setChecked(false);
183     ui->programKpBox->setText("");
184     ui->programKrBox->setText("");
185     ui->programKpSIBox->setChecked(false);
186     ui->programKrSIBox->setChecked(false);
187     ui->programKpModeBox->setCurrentIndex(0);
188     ui->programKrModeBox->setCurrentIndex(0);
189     ui->programKpModeBox->setEnabled(false);
190     ui->programKrModeBox->setEnabled(false);
191     // Clear neu UI elements in MacroTab
192     ui->pteMacroBox->setPlainText("");
193     ui->pteMacroText->setPlainText("");
194     ui->pteMacroComment->setPlainText("");
195     ui->txtBuffer->setText("");
196     // Fill in field and select tab according to action type
197     bool mouse = act.isMouse();
198     if(mouse){
199         // Mouse buttons
200         ui->tabWidget->setCurrentIndex(TAB_MOUSE);
201         // Set mouse buttons (indexOf returns -1 if not found, index zero is blank)
202         ui->mbBox->setCurrentIndex(mouseKeys.indexOf(action) + 1);
203         ui->mb2Box->setCurrentIndex(mouseExtKeys.indexOf(action) + 1);
204         ui->wheelBox->setCurrentIndex(wheelKeys.indexOf(action) + 1);
205         if(act.isDPI()){
206             QPoint custom;
207             int value = act.dpiInfo(custom);
208             if(value <= -DPI_OFFSET || value > KeyAction::DPI_CUSTOM)
209                 return;
210             ui->dpiBox->setCurrentIndex(value + DPI_OFFSET);
211             if(value == KeyAction::DPI_CUSTOM){
212                 ui->dpiCustXBox->setValue(custom.x());
213                 ui->dpiCustYBox->setValue(custom.y());
214             }
215         }
216     } else if(act.isNormal()){
217         // Standard key
218         ui->tabWidget->setCurrentIndex(TAB_KB);
219         ui->typingBox->setCurrentIndex(typingKeys.indexOf(action) + 1);
220         ui->modBox->setCurrentIndex(modKeys.indexOf(action) + 1);
221         ui->fnBox->setCurrentIndex(fnKeys.indexOf(action) + 1);
222         ui->numBox->setCurrentIndex(numKeys.indexOf(action) + 1);
223         ui->mediaBox->setCurrentIndex(mediaKeys.indexOf(action) + 1);
224     } else if(act.isProgram()){
225         // Program key
226         ui->tabWidget->setCurrentIndex(TAB_PROGRAM);
227         QString onPress, onRelease;
228         int stop = act.programInfo(onPress, onRelease);
229         ui->programKpBox->setText(onPress);
230         ui->programKrBox->setText(onRelease);
231         switch(stop & 0x0F){
232         case KeyAction::PROGRAM_PR_MULTI:
233             ui->programKpSIBox->setChecked(false);
234             ui->programKpModeBox->setCurrentIndex(0);
235             ui->programKpModeBox->setEnabled(false);
236             break;
237         default:
238             ui->programKpSIBox->setChecked(true);
239             ui->programKpModeBox->setCurrentIndex(stop & 0x0F);
240             ui->programKpModeBox->setEnabled(true);
241             break;
```

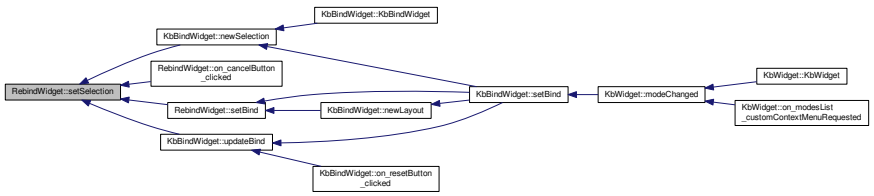
```
242         }
243         switch(stop & 0xF0){
244         case KeyAction::PROGRAM_RE_MULTI:
245             ui->programKrSIBox->setChecked(false);
246             ui->programKrModeBox->setCurrentIndex(0);
247             ui->programKrModeBox->setEnabled(false);
248             break;
249         case KeyAction::PROGRAM_RE_INDEF:
250             ui->programKrSIBox->setChecked(true);
251             ui->programKrModeBox->setCurrentIndex(0);
252             ui->programKrModeBox->setEnabled(true);
253             break;
254         case KeyAction::PROGRAM_RE_KPSTOP:
255             ui->programKrSIBox->setChecked(true);
256             ui->programKrModeBox->setCurrentIndex(1);
257             ui->programKrModeBox->setEnabled(true);
258             break;
259         }
260     } else if(act.isAnim()){
261         // Animation key
262         ui->tabWidget->setCurrentIndex(TAB_ANIM);
263         bool onlyOnce = false, stopOnRelease = false;
264         QUuid id = act.animInfo(onlyOnce, stopOnRelease);
265         // Find this animation in the list. If not found, -1 will be returned, resulting in the blank space
    being selected
266         int index = bind->light()->findAnimIdx(id);
267         ui->animBox->setCurrentIndex(index + 1);
268         ui->animOnceBox->setChecked(!onlyOnce);
269         ui->animKrBox->setChecked(stopOnRelease);
270     } else if(act.isSpecial()){
271         // Other special keys
272         ui->tabWidget->setCurrentIndex(TAB_SPECIAL);
273         int param;
274         QString sAction = act.specialInfo(param);
275         // Mode selection. Check wrap-around flag
276         ui->modeWrapBox->setChecked(true);
277         if(sAction == "mode"){
278             ui->modeWrapBox->setChecked(true);
279             if(param == KeyAction::MODE_PREV_WRAP)
280                 param = KeyAction::MODE_PREV;
281             else if(param == KeyAction::MODE_NEXT_WRAP)
282                 param = KeyAction::MODE_NEXT;
283             else if(param < 0)
284                 ui->modeWrapBox->setChecked(false);
285             // Set mode box to current selection, or to mode 1 if invalid
286             param += 3;
287             if(param >= 0 && param < ui->modeBox->count())
288                 ui->modeBox->setCurrentIndex(param);
289             else
290                 // 0 -> "", 1 -> Prev, 2 -> Next, 3 -> Mode 1
291                 ui->modeBox->setCurrentIndex(3);
292         } else if (sAction == "macro") {
293             ui->tabWidget->setCurrentIndex(TAB_MACRO);
294             if (act.isValidMacro()) {
295                 ui->pteMacroBox->setPlainText(act.macroContent());
296                 ui->pteMacroText->setPlainText(act.macroLine()[1].replace("
    &das_IST_31N_col0n;", ":"));
297                 ui->pteMacroComment->setPlainText(act.macroLine()[2].replace("
    &das_IST_31N_col0n;", ":"));
298                 // Set the invisible Buffer to the original timing information.
299                 // For convenience / Migration from older versions:
300                 // If the timing information is only "x", then ignore it by setting it to an empty QString.
301                 ui->txtBuffer->setText("");
302                 if (act.macroTiming() != "x") ui->txtBuffer->setText(act.macroTiming());
303                 setCorrectRadioButton(act.macroContent());
304             } else {
305                 qDebug("RebindWidget::setSelection found invalid macro definition.");
306                 act.macroDisplay();
307             }
308         } else
309             ui->modeBox->setCurrentIndex(0);
310         // Brightness control. Also check wrap
311         if(sAction == "light"){
312             ui->lightWrapBox->setChecked(true);
313             if(param == KeyAction::LIGHT_DOWN_WRAP)
314                 param = KeyAction::LIGHT_DOWN;
315             else if(param == KeyAction::LIGHT_UP_WRAP)
316                 param = KeyAction::LIGHT_UP;
317             else
318                 ui->lightWrapBox->setChecked(false);
319             if(param < 0 || param > 1)
320                 param = -1;
321             ui->lightBox->setCurrentIndex(param + 1);
322         } else
323             ui->lightBox->setCurrentIndex(0);
324         // Win lock
325         if(sAction == "lock"){
```

```
326             if(param < 0 || param > 2)
327                 param = -1;
328             ui->lockBox->setCurrentIndex(param + 1);
329         } else
330             ui->lockBox->setCurrentIndex(0);
331     }
332 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.81.4 Field Documentation

12.81.4.1 KbBind* RebindWidget::bind [private]

Definition at line 94 of file rebindwidget.h.

Referenced by applyChanges(), on_btnStartMacro_clicked(), on_btnStopMacro_clicked(), on_resetButton_clicked(), on_unbindButton_clicked(), setBind(), and setSelection().

12.81.4.2 QStringList RebindWidget::fnKeys [private]

Definition at line 100 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

12.81.4.3 MacroReader* RebindWidget::macReader [private]

Definition at line 106 of file rebindwidget.h.

Referenced by on_btnStartMacro_clicked(), and on_btnStopMacro_clicked().

12.81.4.4 QStringList RebindWidget::mediaKeys [private]

Definition at line 102 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

12.81.4.5 QStringList RebindWidget::modKeys [private]

Definition at line 99 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

12.81.4.6 QStringList RebindWidget::mouseExtKeys [private]

Definition at line 104 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), and setSelection().

12.81.4.7 QStringList RebindWidget::mouseKeys [private]

Definition at line 103 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), and setSelection().

12.81.4.8 QStringList RebindWidget::numKeys [private]

Definition at line 101 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

12.81.4.9 KbProfile* RebindWidget::profile [private]

Definition at line 95 of file rebindwidget.h.

Referenced by setBind().

12.81.4.10 QStringList RebindWidget::selection [private]

Definition at line 96 of file rebindwidget.h.

Referenced by on_applyButton_clicked(), on_cancelButton_clicked(), on_resetButton_clicked(), on_unbindButton_clicked(), and setSelection().

12.81.4.11 const int RebindWidget::TAB_ANIM = 2 [static], [private]

Definition at line 83 of file rebindwidget.h.

Referenced by setSelection().

12.81.4.12 const int RebindWidget::TAB_KB = 0 [static], [private]

Definition at line 83 of file rebindwidget.h.

Referenced by setSelection().

12.81.4.13 const int RebindWidget::TAB_MACRO = 5 [static], [private]

Definition at line 83 of file rebindwidget.h.

Referenced by setSelection().

12.81.4.14 const int RebindWidget::TAB_MOUSE = 1 [static], [private]

Definition at line 83 of file rebindwidget.h.

Referenced by setSelection().

12.81.4.15 const int RebindWidget::TAB_PROGRAM = 4 [static], [private]

Definition at line 83 of file rebindwidget.h.

Referenced by setSelection().

12.81.4.16 const int RebindWidget::TAB_SPECIAL = 3 [static], [private]

Definition at line 83 of file rebindwidget.h.

Referenced by setSelection().

12.81.4.17 QStringList RebindWidget::typingKeys [private]

Definition at line 98 of file rebindwidget.h.

Referenced by applyChanges(), setBind(), and setSelection().

12.81.4.18 Ui::RebindWidget* RebindWidget::ui [private]

Definition at line 80 of file rebindwidget.h.

Referenced by applyChanges(), convertMacroBox(), helpStatus(), on_animBox_currentIndexChanged(), on_animButton_clicked(), on_btnStartMacro_clicked(), on_btnStopMacro_clicked(), on_dpiBox_currentIndexChanged(),

on_dpiButton_clicked(), on_fnBox_currentIndexChanged(), on_fnButton_clicked(), on_lightBox_currentIndexChanged(), on_lightButton_clicked(), on_lockBox_currentIndexChanged(), on_lockButton_clicked(), on_mb2Box_currentIndexChanged(), on_mb2Button_clicked(), on_mbBox_currentIndexChanged(), on_mbButton_clicked(), on_mediaBox_currentIndexChanged(), on_mediaButton_clicked(), on_modBox_currentIndexChanged(), on_modButton_clicked(), on_modeBox_currentIndexChanged(), on_modeButton_clicked(), on_numBox_currentIndexChanged(), on_numButton_clicked(), on_programKpBox_textChanged(), on_programKpButton_clicked(), on_programKpSIBox_clicked(), on_programKrBox_textChanged(), on_programKrButton_clicked(), on_programKrSIBox_clicked(), on_pteMacroBox_textChanged(), on_typingBox_currentIndexChanged(), on_typingButton_clicked(), on_wheelBox_currentIndexChanged(), on_wheelButton_clicked(), RebindWidget(), setBind(), setBox(), setCorrectRadioButton(), setSelection(), and ~RebindWidget().

12.81.4.19 QStringList RebindWidget::wheelKeys [private]

Definition at line 105 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), and setSelection().

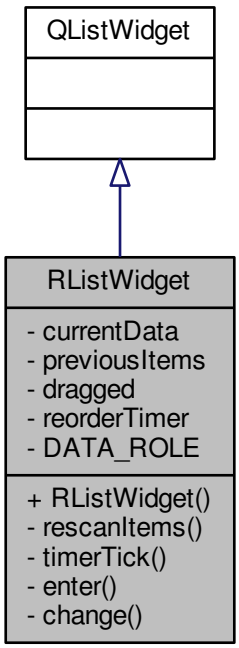
The documentation for this class was generated from the following files:

- src/ckb/rebindwidget.h
- src/ckb/rebindwidget.cpp

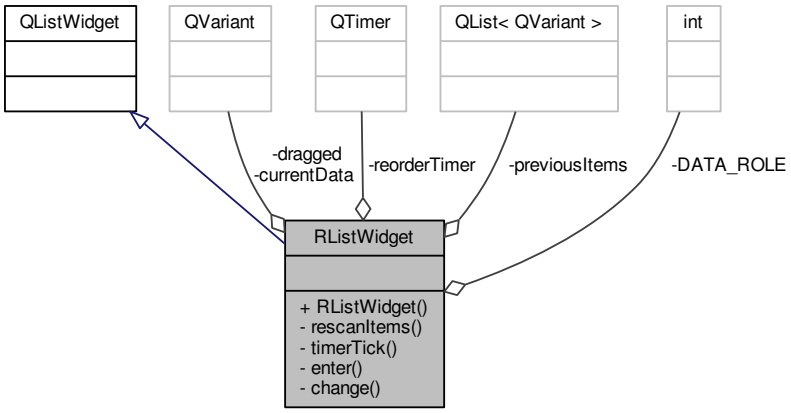
12.82 RListWidget Class Reference

`#include <src/ckb/rlistwidget.h>`

Inheritance diagram for RListWidget:



Collaboration diagram for RListWidget:



Signals

- void orderChanged ()

Public Member Functions

- RListWidget (QWidget ∗parent=0)

Private Slots

- void timerTick ()
- void enter (QListWidgetItem ∗item)
- void change (QListWidgetItem ∗item)

Private Member Functions

- void rescanItems ()

Private Attributes

- QVariant currentData
- QList< QVariant > previousItems
- QVariant dragged
- QTimer reorderTimer

Static Private Attributes

- static const int DATA_ROLE = Qt::UserRole + 100

12.82.1 Detailed Description

Definition at line 9 of file rlistwidget.h.

12.82.2 Constructor & Destructor Documentation

12.82.2.1 RListWidget::RListWidget (QWidget * *parent* = 0) [explicit]

Definition at line 4 of file rlistwidget.cpp.

References change(), enter(), reorderTimer, and timerTick().

```
4                                          :
5     QListWidget(parent)
6 {
7     setDragDropMode(QAbstractItemView::InternalMove);
8     setMovement(QListView::Snap);
9
10     reorderTimer.setSingleShot(true);
11     reorderTimer.setInterval(100);
12     connect(&reorderTimer, SIGNAL(timeout()), this, SLOT(timerTick()));
13
14     connect(this, SIGNAL(itemEntered(QListWidgetItem*)), this, SLOT(enter(QListWidgetItem*)));
15     connect(this, SIGNAL(itemChanged(QListWidgetItem*)), this, SLOT(change(QListWidgetItem*)));
16 }
```

Here is the call graph for this function:

12.82.3 Member Function Documentation

12.82.3.1 void RListWidget::change (QListWidgetItem * *item*) [private], [slot]

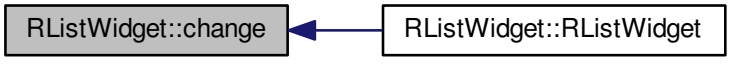
Definition at line 54 of file rlistwidget.cpp.

References reorderTimer.

Referenced by RListWidget().

```
54                                                {
55     reorderTimer.stop();
56     reorderTimer.start();
57 }
```

Here is the caller graph for this function:



12.82.3.2 void RListWidget::enter (QListWidgetItem * *item*) [private],[slot]

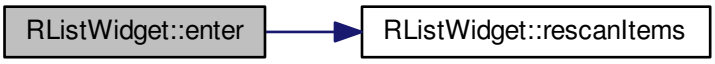
Definition at line 47 of file rlistwidget.cpp.

References DATA_ROLE, dragged, and rescanItems().

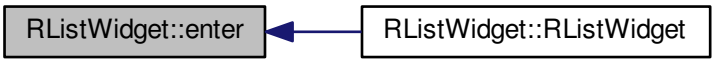
Referenced by RListWidget().

```
47                                              {
48     rescanItems();
49     // Check for drag+drop setup
50     if(item)
51         dragged = item->data(DATA_ROLE);
52 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



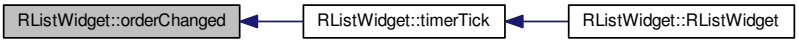
12.82.3.3 void RListWidget::orderChanged ( ) [signal]

Definition at line 141 of file moc_rlistwidget.cpp.

Referenced by timerTick().

```
142 {
143     QMetaObject::activate(this, &staticMetaObject, 0, Q_NULLPTR);
144 }
```

Here is the caller graph for this function:



12.82.3.4 void RListWidget::rescanItems ( ) [private]

Definition at line 59 of file rlistwidget.cpp.

References DATA_ROLE, and previousItems.

Referenced by enter().

```
59                              {
60      QList<QVariant> newItems;
61      int c = count();
62      for(int i = 0; i < c; i++){
63          QVariant data = this->item(i)->data(DATA_ROLE);
64          if(data.isNull()){
65              // Generate the ID for this item if it doesn't already exist
66              data = QUuid::createUuid();
67              this->item(i)->setData(DATA_ROLE, data);
68          }
69          newItems.append(data);
70      }
71      previousItems = newItems;
72  }
```

Here is the caller graph for this function:

12.82.3.5 void RListWidget::timerTick ( ) [private], [slot]

Definition at line 18 of file rlistwidget.cpp.

References DATA_ROLE, dragged, orderChanged(), and previousItems.

Referenced by RListWidget().

```
18                            {
19      bool reordered = false;
20      QListWidgetItem* reselect = 0;
21      QList<QVariant> newItems;
22      // Scan the item list to see if they changed
23      int c = count();
24      for(int i = 0; i < c; i++){
25          QListWidgetItem* itm = item(i);
26          QVariant data = itm->data(DATA_ROLE);
27          newItems.append(data);
28          if(i >= previousItems.count() || data != previousItems[i])
29              reordered = true;
30          // Re-select the dragged item (if any)
31          if(data == dragged)
32              reselect = itm;
33      }
34      if(previousItems.length() != newItems.length())
35          return;
36      if(reordered){
37          previousItems = newItems;
38          emit orderChanged();
39          if(reselect){
40              reselect->setSelected(true);
41              setCurrentItem(reselect);
42              dragged = QVariant();
43          }
44      }
45  }
```

Here is the caller graph for this function:

12.82.4 Field Documentation

12.82.4.1 QVariant RListWidget::currentData [private]

Definition at line 24 of file rlistwidget.h.

12.82.4.2 const int RListWidget::DATA_ROLE = Qt::UserRole + 100 [static], [private]

Definition at line 29 of file rlistwidget.h.

Referenced by enter(), rescanItems(), and timerTick().

12.82.4.3 QVariant RListWidget::dragged [private]

Definition at line 26 of file rlistwidget.h.

Referenced by enter(), and timerTick().

12.82.4.4 QList<QVariant> RListWidget::previousItems [private]

Definition at line 25 of file rlistwidget.h.

Referenced by rescanItems(), and timerTick().

12.82.4.5 QTimer RListWidget::reorderTimer [private]

Definition at line 27 of file rlistwidget.h.

Referenced by change(), and RListWidget().

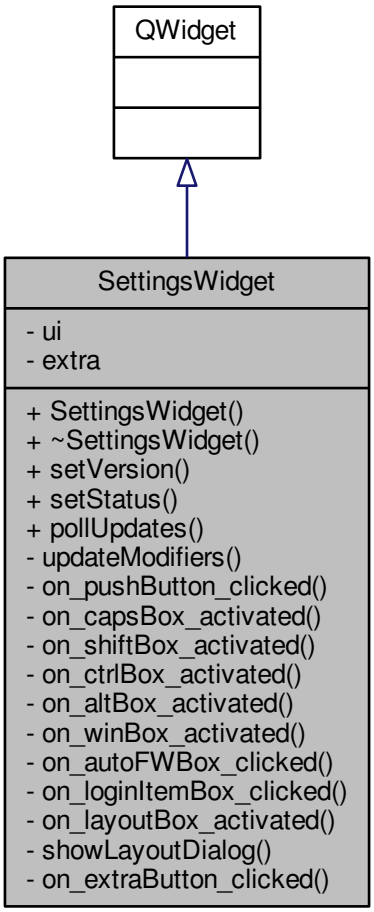
The documentation for this class was generated from the following files:

- src/ckb/rlistwidget.h
- src/ckb/release/moc_rlistwidget.cpp
- src/ckb/rlistwidget.cpp

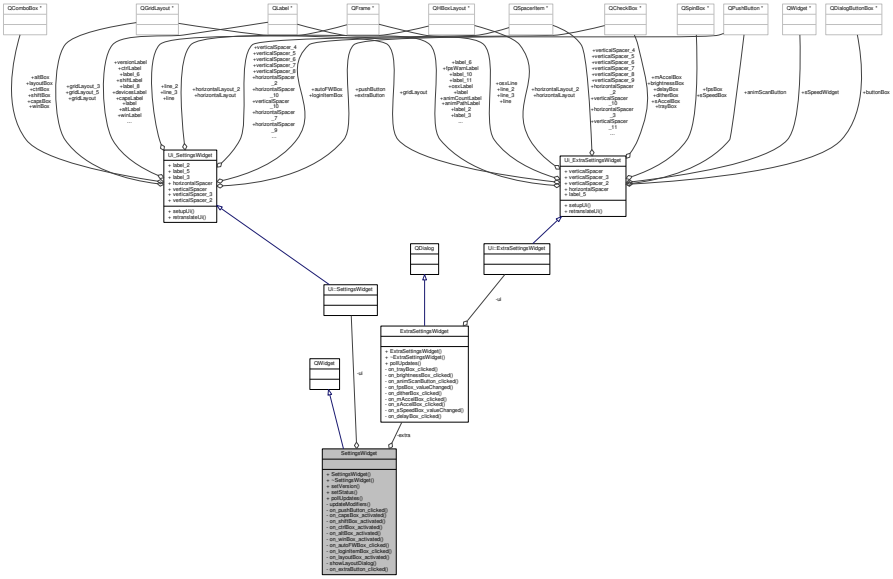
12.83 SettingsWidget Class Reference

`#include <src/ckb/settingswidget.h>`

Inheritance diagram for SettingsWidget:



Collaboration diagram for SettingsWidget:



Public Member Functions

- SettingsWidget (QWidget ∗parent=0)
- ∼SettingsWidget ()
- void setVersion (const QString &version)
- void setStatus (const QString &text)
- void pollUpdates ()

Private Slots

- void on_pushButton_clicked ()
- void on_capsBox_activated (int index)
- void on_shiftBox_activated (int index)
- void on_ctrlBox_activated (int index)
- void on_altBox_activated (int index)
- void on_winBox_activated (int index)
- void on_autoFWBox_clicked (bool checked)
- void on_loginItemBox_clicked (bool checked)
- void on_layoutBox_activated (int index)
- void showLayoutDialog ()
- void on_extraButton_clicked ()

Private Member Functions

- void updateModifiers ()

Private Attributes

- Ui::SettingsWidget ∗ ui
- ExtraSettingsWidget ∗ extra

Friends

- class MainWindow

12.83.1 Detailed Description

Definition at line 11 of file settingswidget.h.

12.83.2 Constructor & Destructor Documentation

12.83.2.1 SettingsWidget::SettingsWidget (QWidget * *parent* = 0) [explicit]

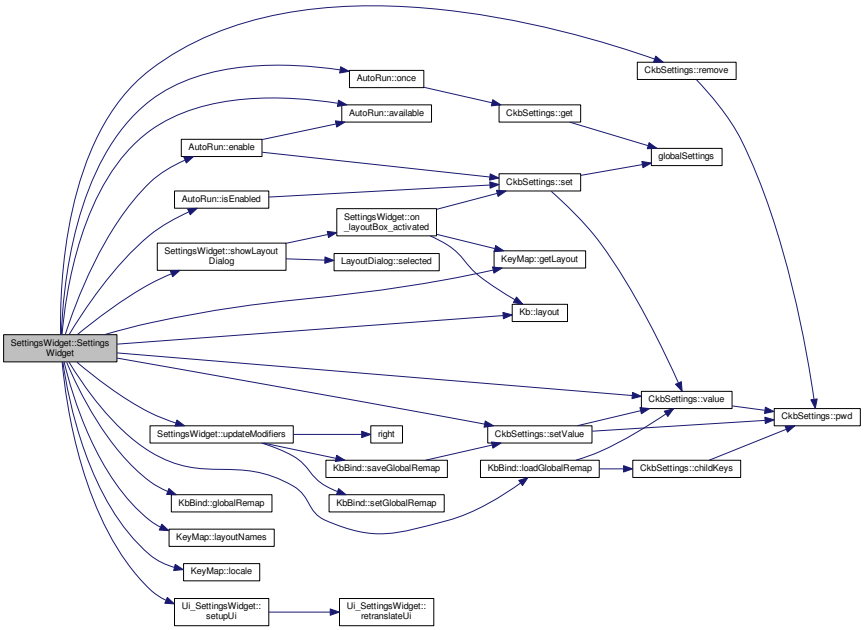
Definition at line 15 of file settingswidget.cpp.

References Ui_SettingsWidget::altBox, Ui_SettingsWidget::autoFWBox, AutoRun::available(), Ui_SettingsWidget::capsBox, Ui_SettingsWidget::ctrlBox, AutoRun::enable(), extra, KeyMap::getLayout(), KbBind::globalRemap(), AutoRun::isEnabled(), Kb::layout(), Ui_SettingsWidget::layoutBox, KeyMap::layoutNames(), KbBind::loadGlobalRemap(), KeyMap::locale(), Ui_SettingsWidget::loginItemBox, modKeys, modNames, KeyMap::NO_LAYOUT, AutoRun::once(), CkbSettings::remove(), Ui_SettingsWidget::setupUi(), CkbSettings::setValue(), Ui_SettingsWidget::shiftBox, showLayoutDialog(), ui, updateModifiers(), CkbSettings::value(), and Ui_SettingsWidget::winBox.

```
15                                                  :
16     QWidget(parent),
17     ui(new Ui::SettingsWidget)
18 {
19     ui->setupUi(this);
20     CkbSettings settings("Program");
21
22     // Read keyboard layout
23     ui->layoutBox->addItems(KeyMap::layoutNames());
24     KeyMap::Layout layout = KeyMap::getLayout(settings.value("KbdLayout").
       toString());
25     if(layout == KeyMap::NO_LAYOUT){
26         // If the layout hasn't been set yet, show a dialog to let the user choose it
27         layout = KeyMap::locale();
28         QTimer::singleShot(1000, this, SLOT(showLayoutDialog()));   // Run the function
       after a delay as the dialog may not appear correctly otherwise
29         settings.setValue("KbdLayout", KeyMap::getLayout(layout));
30     }
31     Kb::layout(layout);
32     ui->layoutBox->setCurrentIndex((int)layout);
33
34     // Load modifier remap
35     KbBind::loadGlobalRemap();
36     if(modKeys.isEmpty()){
37         modKeys << "caps" << "lshift" << "lctrl" << "lalt" << "lwin";
38 #ifdef Q_OS_MACX
39         modNames << "Caps Lock" << "Shift" << "Control ()" << "Option ()" << "Command ()";
40 #else
41         modNames << "Caps Lock" << "Shift" << "Control" << "Alt" << "Super";
42 #endif
43     }
44     foreach(const QString& name, modNames){
45         ui->capsBox->addItem(name);
46         ui->shiftBox->addItem(name);
47         ui->ctrlBox->addItem(name);
48         ui->altBox->addItem(name);
49         ui->winBox->addItem(name);
50     }
51
52     int idx = modKeys.indexOf(KbBind::globalRemap("caps"));
53     if(idx >= 0)
54         ui->capsBox->setCurrentIndex(idx);
55     idx = modKeys.indexOf(KbBind::globalRemap("lshift"));
56     if(idx >= 0)
57         ui->shiftBox->setCurrentIndex(idx);
58     idx = modKeys.indexOf(KbBind::globalRemap("lctrl"));
59     if(idx >= 0)
60         ui->ctrlBox->setCurrentIndex(idx);
61     idx = modKeys.indexOf(KbBind::globalRemap("lwin"));
62     if(idx >= 0)
63         ui->winBox->setCurrentIndex(idx);
64     idx = modKeys.indexOf(KbBind::globalRemap("lalt"));
65     if(idx >= 0)
66         ui->altBox->setCurrentIndex(idx);
```

```
67 #ifdef Q_OS_MACX
68     if(settings.value("osxCmdSwap").toBool()){
69         // ckb <0.0.41
70         settings.remove("osxCmdSwap");
71         ui->ctrlBox->setCurrentIndex(modKeys.indexOf("lwin"));
72         ui->winBox->setCurrentIndex(modKeys.indexOf("lctrl"));
73         updateModifiers();
74     }
75 #endif
76
77     // Read auto update settings
78     ui->autoFWBox->setChecked(!settings.value("DisableAutoFWCheck").toBool());
79
80     // Read auto run settings
81     if(!AutoRun::available())
82         ui->loginItemBox->hide();
83     else {
84         if(!AutoRun::once())
85             // If this is the first time running the app, enable auto run by default
86             AutoRun::enable();
87         ui->loginItemBox->setChecked(AutoRun::isEnabled());
88     }
89
90     // Prepare extra settings
91     extra = new ExtraSettingsWidget(this);
92 }
```

Here is the call graph for this function:



12.83.2.2 SettingsWidget::~SettingsWidget ( )

Definition at line 94 of file settingswidget.cpp.

References ui.

```
94                                {
95     delete ui;
96 }
```

12.83.3 Member Function Documentation

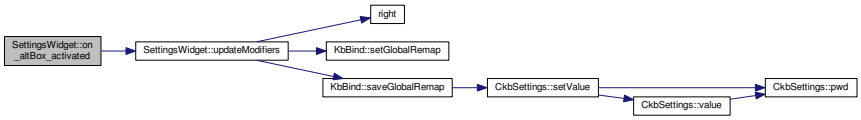
12.83.3.1 void SettingsWidget::on_altBox_activated (int *index*) [private], [slot]

Definition at line 155 of file settingswidget.cpp.

References updateModifiers().

```
155                                                 {
156     updateModifiers();
157 }
```

Here is the call graph for this function:



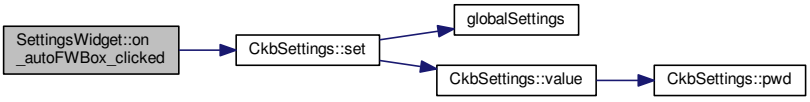
12.83.3.2 void SettingsWidget::on_autoFWBox_clicked (bool *checked*) [private], [slot]

Definition at line 163 of file settingswidget.cpp.

References CkbSettings::set().

```
163                                                     {
164     CkbSettings::set("Program/DisableAutoFWCheck", !checked);
165 }
```

Here is the call graph for this function:



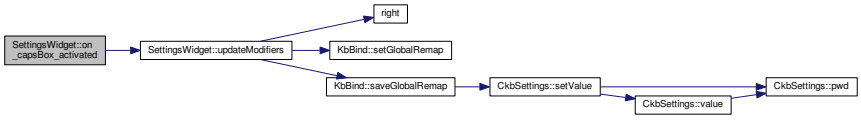
12.83.3.3 void SettingsWidget::on_capsBox_activated (int *index*) [private], [slot]

Definition at line 143 of file settingswidget.cpp.

References updateModifiers().

```
143                                                 {
144     updateModifiers();
145 }
```

Here is the call graph for this function:



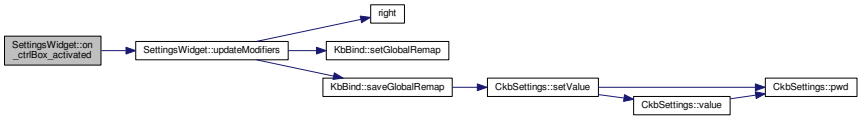
12.83.3.4 void SettingsWidget::on_ctrlBox_activated (int *index*) [private], [slot]

Definition at line 151 of file settingswidget.cpp.

References updateModifiers().

```
151                                                 {
152     updateModifiers();
153 }
```

Here is the call graph for this function:



12.83.3.5 void SettingsWidget::on_extraButton_clicked ( ) [private], [slot]

Definition at line 180 of file settingswidget.cpp.

References extra.

```
180                                             {
181     extra->exec();
182 }
```

12.83.3.6 void SettingsWidget::on_layoutBox_activated (int *index*) [private], [slot]

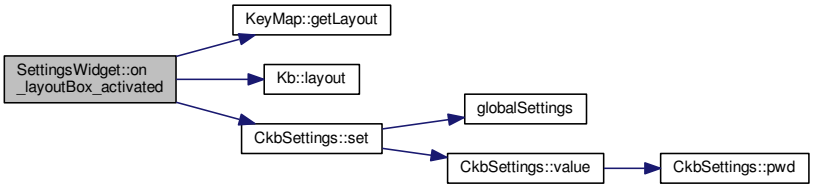
Definition at line 174 of file settingswidget.cpp.

References KeyMap::getLayout(), Kb::layout(), and CkbSettings::set().

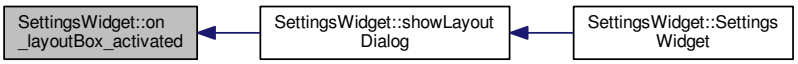
Referenced by showLayoutDialog().

```
174                                                 {
175     KeyMap::Layout layout = (KeyMap::Layout)index;
176     CkbSettings::set("Program/KbdLayout", KeyMap::getLayout(layout));
177     Kb::layout(layout);
178 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.83.3.7 void SettingsWidget::on_loginItemBox_clicked (bool *checked*) [private], [slot]

Definition at line 167 of file settingswidget.cpp.

References AutoRun::disable(), and AutoRun::enable().

```
167                                                     {
168     if(checked)
169         AutoRun::enable();
170     else
171         AutoRun::disable();
172 }
```

Here is the call graph for this function:

12.83.3.8 void SettingsWidget::on_pushButton_clicked ( ) [private], [slot]

Definition at line 139 of file settingswidget.cpp.

```
139                                         {
140     qApp->quit();
141 }
```

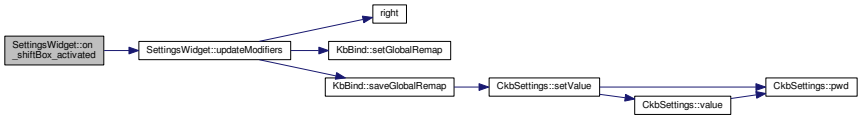
12.83.3.9 void SettingsWidget::on_shiftBox_activated (int *index*) [private], [slot]

Definition at line 147 of file settingswidget.cpp.

References updateModifiers().

```
147                                                 {
148     updateModifiers();
149 }
```

Here is the call graph for this function:



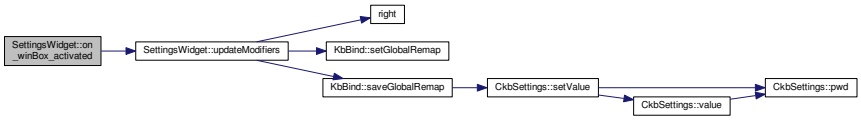
12.83.3.10 void SettingsWidget::on_winBox_activated (int *index*) [private], [slot]

Definition at line 159 of file settingswidget.cpp.

References updateModifiers().

```
159                                                 {
160     updateModifiers();
161 }
```

Here is the call graph for this function:



12.83.3.11 void SettingsWidget::pollUpdates ( )

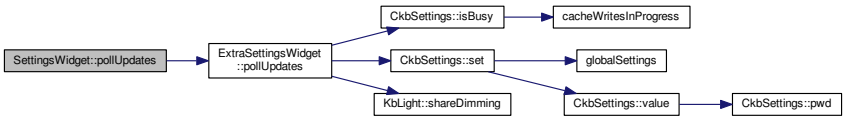
Definition at line 106 of file settingswidget.cpp.

References extra, and ExtraSettingsWidget::pollUpdates().

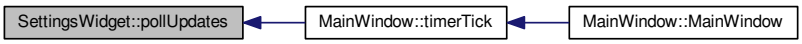
Referenced by MainWindow::timerTick().

```
106                                    {
107     extra->pollUpdates();
108 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.83.3.12 void SettingsWidget::setStatus (const QString & *text*)

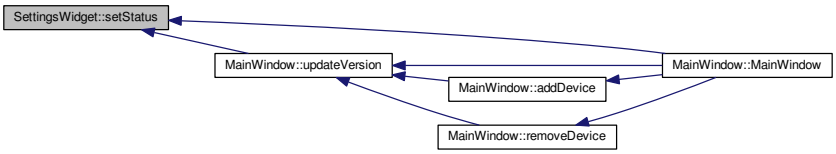
Definition at line 110 of file settingswidget.cpp.

References Ui_SettingsWidget::devicesLabel, and ui.

Referenced by MainWindow::MainWindow(), and MainWindow::updateVersion().

```
110                                                 {
111     ui->devicesLabel->setText(text);
112 }
```

Here is the caller graph for this function:



12.83.3.13 void SettingsWidget::setVersion (const QString & *version*)

Definition at line 114 of file settingswidget.cpp.

References ui, and Ui_SettingsWidget::versionLabel.

Referenced by MainWindow::MainWindow().

```
114                                                     {
115     ui->versionLabel->setText(version);
116 }
```

Here is the caller graph for this function:

12.83.3.14 void SettingsWidget::showLayoutDialog ( ) [private], [slot]

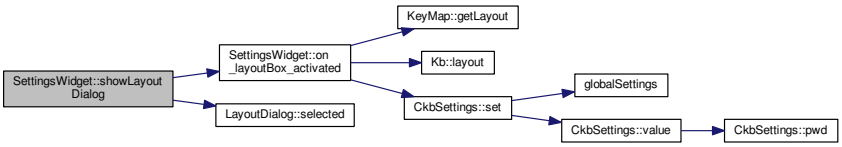
Definition at line 98 of file settingswidget.cpp.

References Ui_SettingsWidget::layoutBox, on_layoutBox_activated(), LayoutDialog::selected(), and ui.

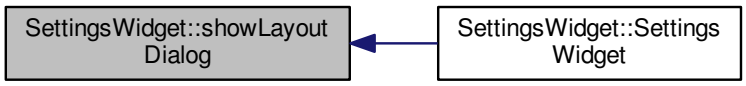
Referenced by SettingsWidget().

```
98                                      {
99     LayoutDialog dialog(this);
100     dialog.exec();
101     // Set selected layout
102     ui->layoutBox->setCurrentIndex((int)dialog.selected());
103     on_layoutBox_activated((int)dialog.selected());         // Call activated()
        signal manually to trigger save
104 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.83.3.15 void SettingsWidget::updateModifiers ( ) [private]

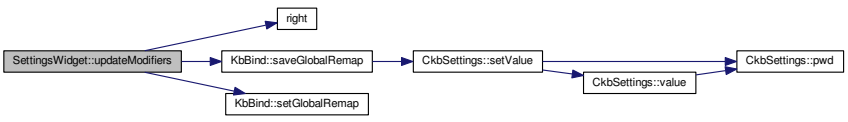
Definition at line 124 of file settingswidget.cpp.

References Ui_SettingsWidget::altBox, Ui_SettingsWidget::capsBox, Ui_SettingsWidget::ctrlBox, modKeys, right(), KbBind::saveGlobalRemap(), KbBind::setGlobalRemap(), Ui_SettingsWidget::shiftBox, ui, and Ui_SettingsWidget-::winBox.

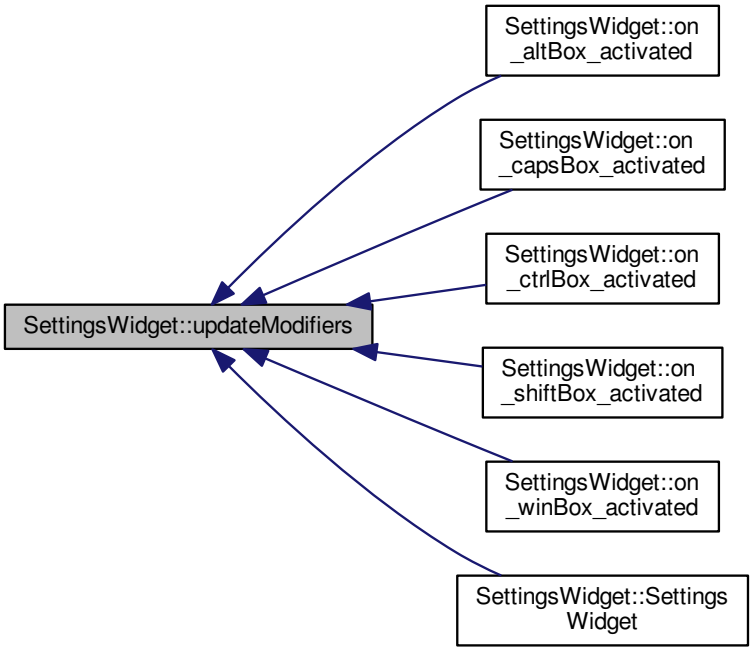
Referenced by on_altBox_activated(), on_capsBox_activated(), on_ctrlBox_activated(), on_shiftBox_activated(), on_winBox_activated(), and SettingsWidget().

```
124                                  {
125     QHash<QString, QString> newMods;
126     newMods["caps"] = modKeys[ui->capsBox->currentIndex()];
127     newMods["lshift"] = modKeys[ui->shiftBox->currentIndex()];
128     newMods["rshift"] = right(modKeys[ui->shiftBox->currentIndex()]);
129     newMods["lctrl"] = modKeys[ui->ctrlBox->currentIndex()];
130     newMods["rctrl"] = right(modKeys[ui->ctrlBox->currentIndex()]);
131     newMods["lalt"] = modKeys[ui->altBox->currentIndex()];
132     newMods["ralt"] = right(modKeys[ui->altBox->currentIndex()]);
133     newMods["lwin"] = modKeys[ui->winBox->currentIndex()];
134     newMods["rwin"] = right(modKeys[ui->winBox->currentIndex()]);
135     KbBind::setGlobalRemap(newMods);
136     KbBind::saveGlobalRemap();
137 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.83.4 Friends And Related Function Documentation

12.83.4.1 friend class MainWindow [friend]

Definition at line 44 of file settingswidget.h.

12.83.5 Field Documentation

12.83.5.1 ExtraSettingsWidget* SettingsWidget::extra [private]

Definition at line 46 of file settingswidget.h.

Referenced by on_extraButton_clicked(), pollUpdates(), and SettingsWidget().

12.83.5.2 Ui::SettingsWidget* SettingsWidget::ui [private]

Definition at line 43 of file settingswidget.h.

Referenced by setStatus(), SettingsWidget(), setVersion(), showLayoutDialog(), updateModifiers(), and ~SettingsWidget().

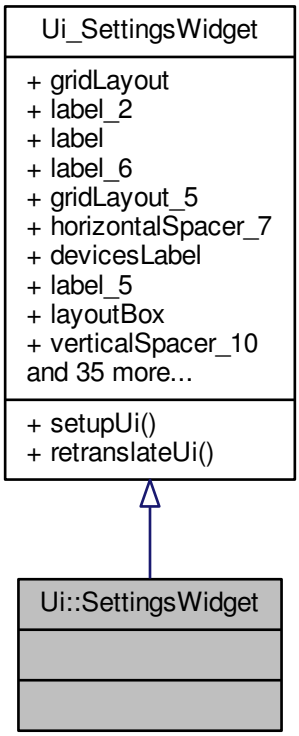
The documentation for this class was generated from the following files:

- src/ckb/settingswidget.h
- src/ckb/settingswidget.cpp

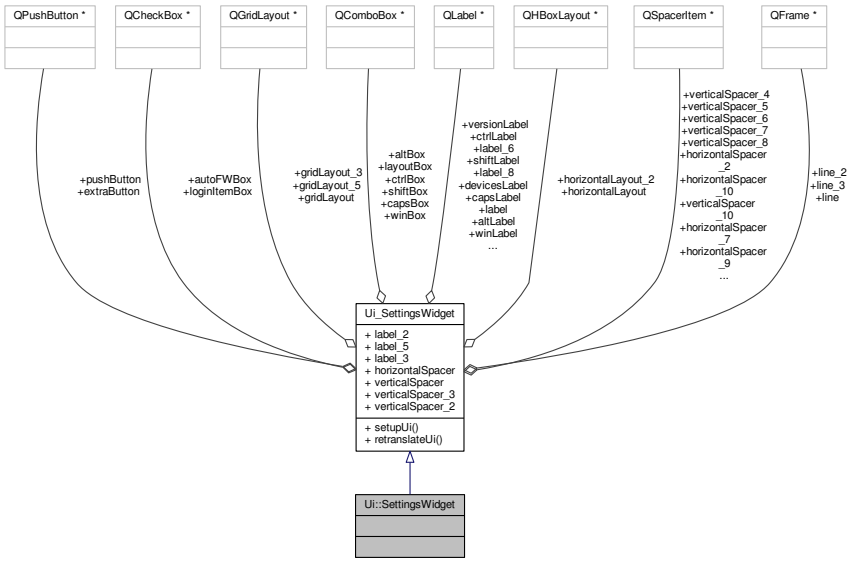
12.84 Ui::SettingsWidget Class Reference

`#include <src/ckb/ui_settingswidget.h>`

Inheritance diagram for Ui::SettingsWidget:



Collaboration diagram for Ui::SettingsWidget:



Public Member Functions

- void setupUi (QWidget *SettingsWidget)
- void retranslateUi (QWidget *SettingsWidget)

Data Fields

- QGridLayout * gridLayout
- QLabel * label_2
- QLabel * label
- QLabel * label_6
- QGridLayout * gridLayout_5
- QSpacerItem * horizontalSpacer_7
- QLabel * devicesLabel
- QLabel * label_5
- QComboBox * layoutBox
- QSpacerItem * verticalSpacer_10
- QHBoxLayout * horizontalLayout_2
- QLabel * label_3
- QSpacerItem * horizontalSpacer
- QPushButton * pushButton
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_8
- QCheckBox * loginItemBox
- QSpacerItem * verticalSpacer_6
- QCheckBox * autoFWBox
- QFrame * line_2
- QLabel * label_8
- QGridLayout * gridLayout_3
- QLabel * shiftLabel
- QLabel * capsLabel

- QSpacerItem * horizontalSpacer_10
- QComboBox * capsBox
- QLabel * ctrlLabel
- QSpacerItem * horizontalSpacer_9
- QComboBox * ctrlBox
- QComboBox * shiftBox
- QLabel * winLabel
- QComboBox * winBox
- QComboBox * altBox
- QLabel * altLabel
- QFrame * line
- QSpacerItem * verticalSpacer_3
- QSpacerItem * verticalSpacer_5
- QFrame * line_3
- QSpacerItem * verticalSpacer_4
- QSpacerItem * verticalSpacer_2
- QLabel * versionLabel
- QHBoxLayout * horizontalLayout
- QPushButton * extraButton
- QSpacerItem * horizontalSpacer_2
- QSpacerItem * verticalSpacer_7

12.84.1 Detailed Description

Definition at line 387 of file ui_settingswidget.h.

12.84.2 Member Function Documentation

12.84.2.1 void Ui_SettingsWidget::retranslateUi (QWidget * *SettingsWidget*) [inline], [inherited]

Definition at line 356 of file ui_settingswidget.h.

References Ui_SettingsWidget::altLabel, Ui_SettingsWidget::autoFWBox, Ui_SettingsWidget::capsLabel, Ui_SettingsWidget::ctrlLabel, Ui_SettingsWidget::devicesLabel, Ui_SettingsWidget::extraButton, Ui_SettingsWidget::label, Ui_SettingsWidget::label_2, Ui_SettingsWidget::label_3, Ui_SettingsWidget::label_5, Ui_SettingsWidget::label_6, Ui_SettingsWidget::label_8, Ui_SettingsWidget::loginItemBox, Ui_SettingsWidget::pushButton, Ui_SettingsWidget::shiftLabel, Ui_SettingsWidget::versionLabel, and Ui_SettingsWidget::winLabel.

Referenced by Ui_SettingsWidget::setupUi().

```
357     {
358         SettingsWidget->setWindowTitle(QApplication::translate("SettingsWidget", "Form", Q_NULLPTR));
359         label_2->setText(QApplication::translate("SettingsWidget", "\302\251 2014-2016 <a href=\"
      https://github.com/ccMSC/\" style=\"text-decoration:none;\">ccMSC</a>.<br/>\302\251 2017 <a href=\"
      https://github.com/mattanger/ckb-next/graphs/contributors\" style=\"text-decoration:none;\">The ckb-next development
       team</a>.", Q_NULLPTR));
360         label->setText(QApplication::translate("SettingsWidget", "Modifier keys", Q_NULLPTR));
361         label_6->setText(QApplication::translate("SettingsWidget", "Application", Q_NULLPTR));
362         devicesLabel->setText(QApplication::translate("SettingsWidget", "No devices connected",
      Q_NULLPTR));
363         label_5->setText(QApplication::translate("SettingsWidget", "Keyboard layout:", Q_NULLPTR));
364         label_3->setText(QApplication::translate("SettingsWidget", "<a href=\"
      https://github.com/mattanger/ckb-next\" style=\"text-decoration:none;\">https://github.com/mattanger/ckb-next</a>", Q_NU
365         pushButton->setText(QApplication::translate("SettingsWidget", "Quit", Q_NULLPTR));
366 #ifndef QT_NO_TOOLTIP
367         loginItemBox->setToolTip(QApplication::translate("SettingsWidget", "ckb-next will be
      started when you log in to your computer.", Q_NULLPTR));
368 #endif // QT_NO_TOOLTIP
369         loginItemBox->setText(QApplication::translate("SettingsWidget", "Start ckb-next at
      login", Q_NULLPTR));
370 #ifndef QT_NO_TOOLTIP
371         autoFWBox->setToolTip(QApplication::translate("SettingsWidget", "You will be notified when
      new firmware versions are available. You'll have the option to install them immediately or wait until
      later.", Q_NULLPTR));
```

```
372 #endif // QT_NO_TOOLTIP
373         autoFWBox->setText(QApplication::translate("SettingsWidget", "Check for new firmware
      automatically", Q_NULLPTR));
374         label_8->setText(QApplication::translate("SettingsWidget", "These will override the keyboard
      profile. See \"Binding\" tab for more settings.", Q_NULLPTR));
375         shiftLabel->setText(QApplication::translate("SettingsWidget", "Shift:", Q_NULLPTR));
376         capsLabel->setText(QApplication::translate("SettingsWidget", "Caps Lock:", Q_NULLPTR));
377         ctrlLabel->setText(QApplication::translate("SettingsWidget", "Ctrl:", Q_NULLPTR));
378         winLabel->setText(QApplication::translate("SettingsWidget", "Windows:", Q_NULLPTR));
379         altLabel->setText(QApplication::translate("SettingsWidget", "Alt:", Q_NULLPTR));
380         versionLabel->setText(QApplication::translate("SettingsWidget", "ckb-next", Q_NULLPTR))
    ;
381         extraButton->setText(QApplication::translate("SettingsWidget", "More settings",
     Q_NULLPTR));
382     } // retranslateUi
```

Here is the caller graph for this function:

12.84.2.2 void Ui_SettingsWidget::setupUi (QWidget * *SettingsWidget*) [inline],[inherited]

Definition at line 78 of file ui_settingswidget.h.

References Ui_SettingsWidget::altBox, Ui_SettingsWidget::altLabel, Ui_SettingsWidget::autoFWBox, Ui_SettingsWidget::capsBox, Ui_SettingsWidget::capsLabel, Ui_SettingsWidget::ctrlBox, Ui_SettingsWidget::ctrlLabel, Ui_SettingsWidget::devicesLabel, Ui_SettingsWidget::extraButton, Ui_SettingsWidget::gridLayout, Ui_SettingsWidget::gridLayout_3, Ui_SettingsWidget::gridLayout_5, Ui_SettingsWidget::horizontalLayout, Ui_SettingsWidget::horizontalLayout_2, Ui_SettingsWidget::horizontalSpacer, Ui_SettingsWidget::horizontalSpacer_10, Ui_SettingsWidget::horizontalSpacer_2, Ui_SettingsWidget::horizontalSpacer_7, Ui_SettingsWidget::horizontalSpacer_9, Ui_SettingsWidget::label, Ui_SettingsWidget::label_2, Ui_SettingsWidget::label_3, Ui_SettingsWidget::label_5, Ui_SettingsWidget::label_6, Ui_SettingsWidget::label_8, Ui_SettingsWidget::layoutBox, Ui_SettingsWidget::line, Ui_SettingsWidget::line_2, Ui_SettingsWidget::line_3, Ui_SettingsWidget::loginItemBox, Ui_SettingsWidget::pushButton, Ui_SettingsWidget::retranslateUi(), Ui_SettingsWidget::shiftBox, Ui_SettingsWidget::shiftLabel, Ui_SettingsWidget::versionLabel, Ui_SettingsWidget::verticalSpacer, Ui_SettingsWidget::verticalSpacer_10, Ui_SettingsWidget::verticalSpacer_2, Ui_SettingsWidget::verticalSpacer_3, Ui_SettingsWidget::verticalSpacer_4, Ui_SettingsWidget::verticalSpacer_5, Ui_SettingsWidget::verticalSpacer_6, Ui_SettingsWidget::verticalSpacer_7, Ui_SettingsWidget::verticalSpacer_8, Ui_SettingsWidget::winBox, and Ui_SettingsWidget::winLabel.

Referenced by SettingsWidget::SettingsWidget().

```
79     {
80         if (SettingsWidget->objectName().isEmpty())
81             SettingsWidget->setObjectName(QStringLiteral("SettingsWidget"));
82         SettingsWidget->resize(827, 654);
83         gridLayout = new QGridLayout(SettingsWidget);
84         gridLayout->setObjectName(QStringLiteral("gridLayout"));
85         label_2 = new QLabel(SettingsWidget);
86         label_2->setObjectName(QStringLiteral("label_2"));
87         label_2->setAlignment(Qt::AlignLeading|Qt::AlignLeft|Qt::AlignVCenter);
88         label_2->setOpenExternalLinks(true);
89
90         gridLayout->addWidget(label_2, 20, 0, 1, 2);
91
92         label = new QLabel(SettingsWidget);
93         label->setObjectName(QStringLiteral("label"));
94         QFont font;
95         font.setBold(true);
96         font.setWeight(75);
97         label->setFont(font);
98
99         gridLayout->addWidget(label, 6, 0, 1, 2);
100
```

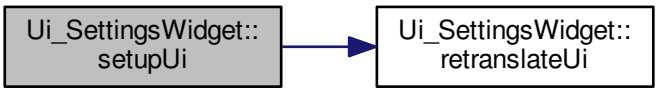
```
101         label_6 = new QLabel(SettingsWidget);
102         label_6->setObjectName(QStringLiteral("label_6"));
103         label_6->setFont(font);
104 
105         gridLayout->addWidget(label_6, 12, 0, 1, 2);
106 
107         gridLayout_5 = new QGridLayout();
108         gridLayout_5->setObjectName(QStringLiteral("gridLayout_5"));
109         horizontalSpacer_7 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
110 
111         gridLayout_5->addItem(horizontalSpacer_7, 2, 2, 1, 1);
112 
113         devicesLabel = new QLabel(SettingsWidget);
114         devicesLabel->setObjectName(QStringLiteral("devicesLabel"));
115 
116         gridLayout_5->addWidget(devicesLabel, 0, 0, 1, 3);
117 
118         label_5 = new QLabel(SettingsWidget);
119         label_5->setObjectName(QStringLiteral("label_5"));
120 
121         gridLayout_5->addWidget(label_5, 2, 0, 1, 1);
122 
123         layoutBox = new QComboBox(SettingsWidget);
124         layoutBox->setObjectName(QStringLiteral("layoutBox"));
125 
126         gridLayout_5->addWidget(layoutBox, 2, 1, 1, 1);
127 
128         verticalSpacer_10 = new QSpacerItem(20, 6, QSizePolicy::Minimum,
    QSizePolicy::Fixed);
129 
130         gridLayout_5->addItem(verticalSpacer_10, 1, 0, 1, 1);
131 
132 
133         gridLayout->addLayout(gridLayout_5, 4, 1, 1, 1);
134 
135         horizontalLayout_2 = new QHBoxLayout();
136         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
137         label_3 = new QLabel(SettingsWidget);
138         label_3->setObjectName(QStringLiteral("label_3"));
139         label_3->setOpenExternalLinks(true);
140 
141         horizontalLayout_2->addWidget(label_3);
142 
143         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
144 
145         horizontalLayout_2->addItem(horizontalSpacer);
146 
147         pushButton = new QPushButton(SettingsWidget);
148         pushButton->setObjectName(QStringLiteral("pushButton"));
149         pushButton->setFlat(false);
150 
151         horizontalLayout_2->addWidget(pushButton);
152 
153 
154         gridLayout->addLayout(horizontalLayout_2, 21, 0, 1, 2);
155 
156         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
157 
158         gridLayout->addItem(verticalSpacer, 19, 0, 1, 2);
159 
160         verticalSpacer_8 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
161 
162         gridLayout->addItem(verticalSpacer_8, 16, 0, 1, 1);
163 
164         loginItemBox = new QCheckBox(SettingsWidget);
165         loginItemBox->setObjectName(QStringLiteral("loginItemBox"));
166 
167         gridLayout->addWidget(loginItemBox, 16, 1, 1, 1);
168 
169         verticalSpacer_6 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
170 
171         gridLayout->addItem(verticalSpacer_6, 15, 0, 1, 1);
172 
173         autoFWBox = new QCheckBox(SettingsWidget);
174         autoFWBox->setObjectName(QStringLiteral("autoFWBox"));
175         autoFWBox->setChecked(true);
176 
177         gridLayout->addWidget(autoFWBox, 15, 1, 1, 1);
178 
179         line_2 = new QFrame(SettingsWidget);
180         line_2->setObjectName(QStringLiteral("line_2"));
181         line_2->setFrameShape(QFrame::HLine);
```

```
182         line_2->setFrameShadow(QFrame::Sunken);
183 
184         gridLayout->addWidget(line_2, 13, 0, 1, 2);
185 
186         label_8 = new QLabel(SettingsWidget);
187         label_8->setObjectName(QStringLiteral("label_8"));
188         QFont font1;
189         font1.setItalic(true);
190         label_8->setFont(font1);
191 
192         gridLayout->addWidget(label_8, 8, 1, 1, 1);
193 
194         gridLayout_3 = new QGridLayout();
195         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
196         shiftLabel = new QLabel(SettingsWidget);
197         shiftLabel->setObjectName(QStringLiteral("shiftLabel"));
198         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
199         sizePolicy.setHorizontalStretch(0);
200         sizePolicy.setVerticalStretch(0);
201         sizePolicy.setHeightForWidth(shiftLabel->sizePolicy().hasHeightForWidth());
202         shiftLabel->setSizePolicy(sizePolicy);
203         shiftLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
204 
205         gridLayout_3->addWidget(shiftLabel, 0, 3, 1, 1);
206 
207         capsLabel = new QLabel(SettingsWidget);
208         capsLabel->setObjectName(QStringLiteral("capsLabel"));
209         capsLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
210 
211         gridLayout_3->addWidget(capsLabel, 0, 1, 1, 1);
212 
213         horizontalSpacer_10 = new QSpacerItem(0, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
214 
215         gridLayout_3->addItem(horizontalSpacer_10, 0, 0, 1, 1);
216 
217         capsBox = new QComboBox(SettingsWidget);
218         capsBox->setObjectName(QStringLiteral("capsBox"));
219         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Fixed);
220         sizePolicy1.setHorizontalStretch(0);
221         sizePolicy1.setVerticalStretch(0);
222         sizePolicy1.setHeightForWidth(capsBox->sizePolicy().hasHeightForWidth());
223         capsBox->setSizePolicy(sizePolicy1);
224 
225         gridLayout_3->addWidget(capsBox, 0, 2, 1, 1);
226 
227         ctrlLabel = new QLabel(SettingsWidget);
228         ctrlLabel->setObjectName(QStringLiteral("ctrlLabel"));
229         ctrlLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
230 
231         gridLayout_3->addWidget(ctrlLabel, 1, 1, 1, 1);
232 
233         horizontalSpacer_9 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
234 
235         gridLayout_3->addItem(horizontalSpacer_9, 1, 7, 1, 1);
236 
237         ctrlBox = new QComboBox(SettingsWidget);
238         ctrlBox->setObjectName(QStringLiteral("ctrlBox"));
239 
240         gridLayout_3->addWidget(ctrlBox, 1, 2, 1, 1);
241 
242         shiftBox = new QComboBox(SettingsWidget);
243         shiftBox->setObjectName(QStringLiteral("shiftBox"));
244         sizePolicy1.setHeightForWidth(shiftBox->sizePolicy().hasHeightForWidth());
245         shiftBox->setSizePolicy(sizePolicy1);
246 
247         gridLayout_3->addWidget(shiftBox, 0, 4, 1, 1);
248 
249         winLabel = new QLabel(SettingsWidget);
250         winLabel->setObjectName(QStringLiteral("winLabel"));
251         sizePolicy.setHeightForWidth(winLabel->sizePolicy().hasHeightForWidth());
252         winLabel->setSizePolicy(sizePolicy);
253         winLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
254 
255         gridLayout_3->addWidget(winLabel, 1, 3, 1, 1);
256 
257         winBox = new QComboBox(SettingsWidget);
258         winBox->setObjectName(QStringLiteral("winBox"));
259         sizePolicy1.setHeightForWidth(winBox->sizePolicy().hasHeightForWidth());
260         winBox->setSizePolicy(sizePolicy1);
261 
262         gridLayout_3->addWidget(winBox, 1, 4, 1, 1);
263 
264         altBox = new QComboBox(SettingsWidget);
265         altBox->setObjectName(QStringLiteral("altBox"));
266         sizePolicy1.setHeightForWidth(altBox->sizePolicy().hasHeightForWidth());
```

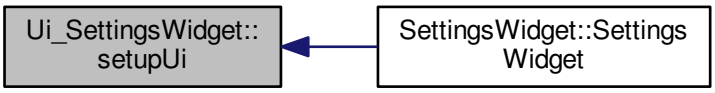
```
267         altBox->setSizePolicy(sizePolicy1);
268 
269         gridLayout_3->addWidget(altBox, 1, 6, 1, 1);
270 
271         altLabel = new QLabel(SettingsWidget);
272         altLabel->setObjectName(QStringLiteral("altLabel"));
273         sizePolicy.setHeightForWidth(altLabel->sizePolicy().hasHeightForWidth());
274         altLabel->setSizePolicy(sizePolicy);
275         altLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
276 
277         gridLayout_3->addWidget(altLabel, 1, 5, 1, 1);
278 
279 
280         gridLayout->addLayout(gridLayout_3, 10, 1, 1, 1);
281 
282         line = new QFrame(SettingsWidget);
283         line->setObjectName(QStringLiteral("line"));
284         line->setFrameShape(QFrame::HLine);
285         line->setFrameShadow(QFrame::Sunken);
286 
287         gridLayout->addWidget(line, 1, 0, 1, 2);
288 
289         verticalSpacer_3 = new QSpacerItem(0, 5, QSizePolicy::Minimum, QSizePolicy::Fixed);
290 
291         gridLayout->addItem(verticalSpacer_3, 2, 1, 1, 1);
292 
293         verticalSpacer_5 = new QSpacerItem(20, 12, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
294 
295         gridLayout->addItem(verticalSpacer_5, 11, 1, 1, 1);
296 
297         line_3 = new QFrame(SettingsWidget);
298         line_3->setObjectName(QStringLiteral("line_3"));
299         QSizePolicy sizePolicy2(QSizePolicy::Minimum, QSizePolicy::Fixed);
300         sizePolicy2.setHorizontalStretch(0);
301         sizePolicy2.setVerticalStretch(0);
302         sizePolicy2.setHeightForWidth(line_3->sizePolicy().hasHeightForWidth());
303         line_3->setSizePolicy(sizePolicy2);
304         line_3->setFrameShape(QFrame::HLine);
305         line_3->setFrameShadow(QFrame::Sunken);
306 
307         gridLayout->addWidget(line_3, 7, 0, 1, 2);
308 
309         verticalSpacer_4 = new QSpacerItem(20, 6, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
310 
311         gridLayout->addItem(verticalSpacer_4, 9, 1, 1, 1);
312 
313         verticalSpacer_2 = new QSpacerItem(20, 12, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
314 
315         gridLayout->addItem(verticalSpacer_2, 5, 1, 1, 1);
316 
317         versionLabel = new QLabel(SettingsWidget);
318         versionLabel->setObjectName(QStringLiteral("versionLabel"));
319         versionLabel->setFont(font);
320 
321         gridLayout->addWidget(versionLabel, 0, 0, 1, 2);
322 
323         horizontalLayout = new QHBoxLayout();
324         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
325         extraButton = new QPushButton(SettingsWidget);
326         extraButton->setObjectName(QStringLiteral("extraButton"));
327 
328         horizontalLayout->addWidget(extraButton);
329 
330         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
331 
332         horizontalLayout->addItem(horizontalSpacer_2);
333 
334 
335         gridLayout->addLayout(horizontalLayout, 17, 1, 1, 1);
336 
337         verticalSpacer_7 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
338 
339         gridLayout->addItem(verticalSpacer_7, 17, 0, 1, 1);
340 
341         QWidget::setTabOrder(layoutBox, capsBox);
342         QWidget::setTabOrder(capsBox, shiftBox);
343         QWidget::setTabOrder(shiftBox, ctrlBox);
344         QWidget::setTabOrder(ctrlBox, winBox);
345         QWidget::setTabOrder(winBox, altBox);
346         QWidget::setTabOrder(altBox, autoFWBox);
347         QWidget::setTabOrder(autoFWBox, loginItemBox);
348         QWidget::setTabOrder(loginItemBox, extraButton);
```

```
349         QWidget::setTabOrder(extraButton, pushButton);
350
351         retranslateUi(SettingsWidget);
352
353         QMetaObject::connectSlotsByName(SettingsWidget);
354     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.84.3 Field Documentation

12.84.3.1 QComboBox* Ui_SettingsWidget::altBox [inherited]

Definition at line 64 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), Ui_SettingsWidget::setupUi(), and SettingsWidget::updateModifiers().

12.84.3.2 QLabel* Ui_SettingsWidget::altLabel [inherited]

Definition at line 65 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.3 QCheckBox* Ui_SettingsWidget::autoFWBox [inherited]

Definition at line 50 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), SettingsWidget::SettingsWidget(), and Ui_SettingsWidget::setupUi().

12.84.3.4 QComboBox* Ui_SettingsWidget::capsBox [inherited]

Definition at line 57 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), Ui_SettingsWidget::setupUi(), and SettingsWidget::updateModifiers().

12.84.3.5 QLabel* Ui_SettingsWidget::capsLabel [inherited]

Definition at line 55 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.6 QComboBox* Ui_SettingsWidget::ctrlBox [inherited]

Definition at line 60 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), Ui_SettingsWidget::setupUi(), and SettingsWidget::updateModifiers().

12.84.3.7 QLabel* Ui_SettingsWidget::ctrlLabel [inherited]

Definition at line 58 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.8 QLabel* Ui_SettingsWidget::devicesLabel [inherited]

Definition at line 38 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), SettingsWidget::setStatus(), and Ui_SettingsWidget::setupUi().

12.84.3.9 QPushButton* Ui_SettingsWidget::extraButton [inherited]

Definition at line 74 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.10 QGridLayout* Ui_SettingsWidget::gridLayout [inherited]

Definition at line 32 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.11 QGridLayout* Ui_SettingsWidget::gridLayout_3 [inherited]

Definition at line 53 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.12 QGridLayout* Ui_SettingsWidget::gridLayout_5 [inherited]

Definition at line 36 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.13 QHBoxLayout* Ui_SettingsWidget::horizontalLayout [inherited]

Definition at line 73 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.14 QHBoxLayout* Ui_SettingsWidget::horizontalLayout_2 [inherited]

Definition at line 42 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.15 QSpacerItem* Ui_SettingsWidget::horizontalSpacer [inherited]

Definition at line 44 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.16 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_10 [inherited]

Definition at line 56 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.17 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_2 [inherited]

Definition at line 75 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.18 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_7 [inherited]

Definition at line 37 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.19 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_9 [inherited]

Definition at line 59 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.20 QLabel* Ui_SettingsWidget::label [inherited]

Definition at line 34 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.21 QLabel* Ui_SettingsWidget::label_2 [inherited]

Definition at line 33 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.22 QLabel* Ui_SettingsWidget::label_3 [inherited]

Definition at line 43 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.23 QLabel* Ui_SettingsWidget::label_5 [inherited]

Definition at line 39 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.24 QLabel* Ui_SettingsWidget::label_6 [inherited]

Definition at line 35 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.25 QLabel* Ui_SettingsWidget::label_8 [inherited]

Definition at line 52 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.26 QComboBox* Ui_SettingsWidget::layoutBox [inherited]

Definition at line 40 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), Ui_SettingsWidget::setupUi(), and SettingsWidget::showLayout-Dialog().

12.84.3.27 QFrame* Ui_SettingsWidget::line [inherited]

Definition at line 66 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.28 QFrame* Ui_SettingsWidget::line_2 [inherited]

Definition at line 51 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.29 QFrame* Ui_SettingsWidget::line_3 [inherited]

Definition at line 69 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.30 QCheckBox* Ui_SettingsWidget::loginItemBox [inherited]

Definition at line 48 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), SettingsWidget::SettingsWidget(), and Ui_SettingsWidget-::setupUi().

12.84.3.31 QPushButton* Ui_SettingsWidget::pushButton [inherited]

Definition at line 45 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.32 QComboBox* Ui_SettingsWidget::shiftBox [inherited]

Definition at line 61 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), Ui_SettingsWidget::setupUi(), and SettingsWidget::updateModifiers().

12.84.3.33 QLabel* Ui_SettingsWidget::shiftLabel [inherited]

Definition at line 54 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

12.84.3.34 QLabel* Ui_SettingsWidget::versionLabel [inherited]

Definition at line 72 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), Ui_SettingsWidget::setupUi(), and SettingsWidget::setVersion().

12.84.3.35 QSpacerItem* Ui_SettingsWidget::verticalSpacer [inherited]

Definition at line 46 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.36 QSpacerItem* Ui_SettingsWidget::verticalSpacer_10 [inherited]

Definition at line 41 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.37 QSpacerItem* Ui_SettingsWidget::verticalSpacer_2 [inherited]

Definition at line 71 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.38 QSpacerItem* Ui_SettingsWidget::verticalSpacer_3 [inherited]

Definition at line 67 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.39 QSpacerItem* Ui_SettingsWidget::verticalSpacer_4 [inherited]

Definition at line 70 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.40 QSpacerItem* Ui_SettingsWidget::verticalSpacer_5 [inherited]

Definition at line 68 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.41 QSpacerItem* Ui_SettingsWidget::verticalSpacer_6 [inherited]

Definition at line 49 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.42 QSpacerItem* Ui_SettingsWidget::verticalSpacer_7 [inherited]

Definition at line 76 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.43 QSpacerItem* Ui_SettingsWidget::verticalSpacer_8 [inherited]

Definition at line 47 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::setupUi().

12.84.3.44 QComboBox* Ui_SettingsWidget::winBox [inherited]

Definition at line 63 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), Ui_SettingsWidget::setupUi(), and SettingsWidget::updateModifiers().

12.84.3.45 QLabel* Ui_SettingsWidget::winLabel [inherited]

Definition at line 62 of file ui_settingswidget.h.

Referenced by Ui_SettingsWidget::retranslateUi(), and Ui_SettingsWidget::setupUi().

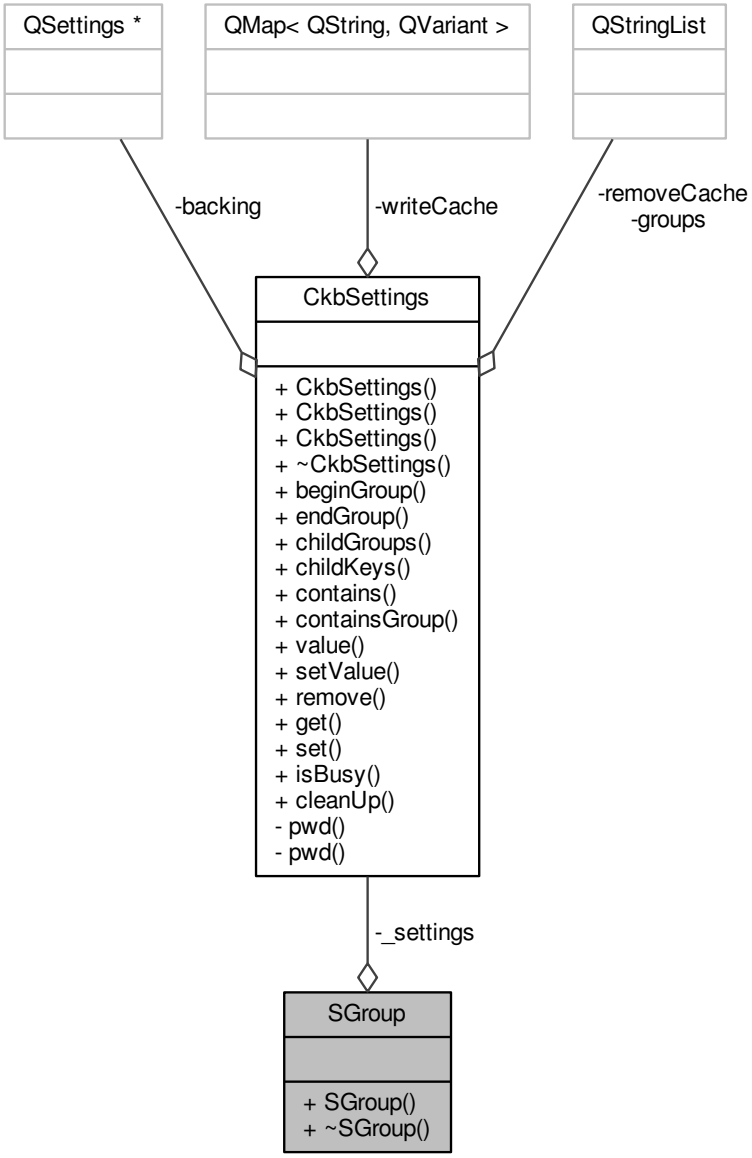
The documentation for this class was generated from the following file:

- src/ckb/ui_settingswidget.h

12.85 SGroup Class Reference

`#include <src/ckb/ckbsettings.h>`

Collaboration diagram for SGroup:



Public Member Functions

- SGroup (CkbSettings &settings, const QString &prefix)
- ~SGroup ()

Private Attributes

- CkbSettings & _settings

12.85.1 Detailed Description

Definition at line 57 of file ckbsettings.h.

12.85.2 Constructor & Destructor Documentation

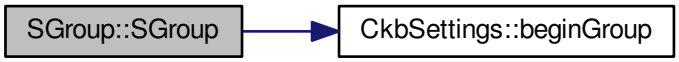
12.85.2.1 SGroup::SGroup (CkbSettings & *settings*, const QString & *prefix*) [inline]

Definition at line 60 of file ckbsettings.h.

References CkbSettings::beginGroup().

```
60 : _settings(settings) { settings.beginGroup(prefix); }
```

Here is the call graph for this function:



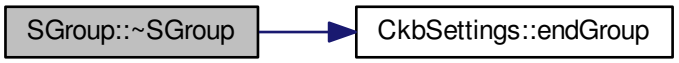
12.85.2.2 SGroup::~SGroup ( ) [inline]

Definition at line 61 of file ckbsettings.h.

References _settings, and CkbSettings::endGroup().

```
61 { _settings.endGroup(); }
```

Here is the call graph for this function:



12.85.3 Field Documentation

12.85.3.1 CkbSettings& SGroup::_settings [private]

Definition at line 64 of file ckbsettings.h.

Referenced by ~SGroup().

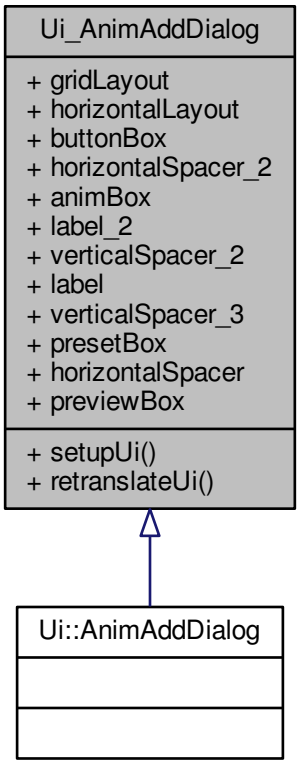
The documentation for this class was generated from the following file:

- src/ckb/ckbsettings.h

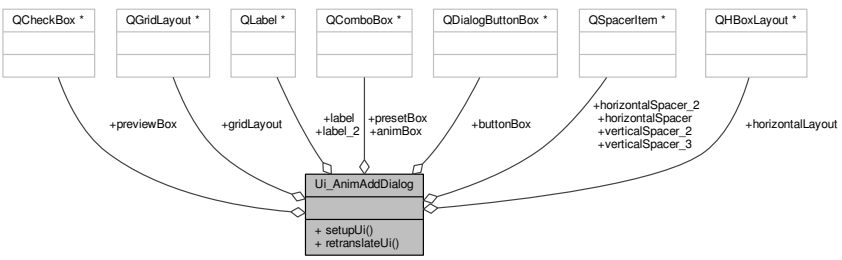
12.86 Ui_AnimAddDialog Class Reference

`#include <src/ckb/ui_animadddialog.h>`

Inheritance diagram for Ui_AnimAddDialog:



Collaboration diagram for Ui_AnimAddDialog:



Public Member Functions

- void setupUi (QDialog *AnimAddDialog)
- void retranslateUi (QDialog *AnimAddDialog)

Data Fields

- QGridLayout * gridLayout
- QHBoxLayout * horizontalLayout
- QDialogButtonBox * buttonBox
- QSpacerItem * horizontalSpacer_2
- QComboBox * animBox
- QLabel * label_2
- QSpacerItem * verticalSpacer_2
- QLabel * label
- QSpacerItem * verticalSpacer_3
- QComboBox * presetBox
- QSpacerItem * horizontalSpacer
- QCheckBox * previewBox

12.86.1 Detailed Description

Definition at line 28 of file ui_animadddialog.h.

12.86.2 Member Function Documentation

12.86.2.1 void Ui_AnimAddDialog::retranslateUi (QDialog * *AnimAddDialog*) [inline]

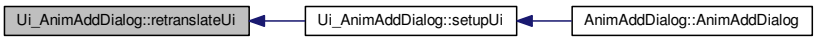
Definition at line 135 of file ui_animadddialog.h.

References label, label_2, and previewBox.

Referenced by setupUi().

```
136     {
137         AnimAddDialog->setWindowTitle(QApplication::translate("AnimAddDialog", "Add Animation", Q_NULLPTR))
    ;
138         label_2->setText(QApplication::translate("AnimAddDialog", "Preset:", Q_NULLPTR));
139         label->setText(QApplication::translate("AnimAddDialog", "Animation:", Q_NULLPTR));
140         previewBox->setText(QApplication::translate("AnimAddDialog", "Show preview", Q_NULLPTR));
141     } // retranslateUi
```

Here is the caller graph for this function:



12.86.2.2 void Ui_AnimAddDialog::setupUi (QDialog * *AnimAddDialog*) [inline]

Definition at line 44 of file ui_animadddialog.h.

References animBox, buttonBox, gridLayout, horizontalLayout, horizontalSpacer, horizontalSpacer_2, label, label_2, presetBox, previewBox, retranslateUi(), verticalSpacer_2, and verticalSpacer_3.

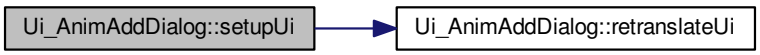
Referenced by AnimAddDialog::AnimAddDialog().

```
45     {
46         if (AnimAddDialog->objectName().isEmpty())
47             AnimAddDialog->setObjectName(QStringLiteral("AnimAddDialog"));
48         AnimAddDialog->resize(350, 205);
```

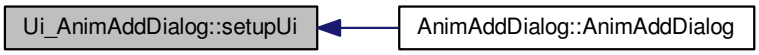
```
49          QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
50          sizePolicy.setHorizontalStretch(0);
51          sizePolicy.setVerticalStretch(0);
52          sizePolicy.setHeightForWidth(AnimAddDialog->sizePolicy().hasHeightForWidth());
53          AnimAddDialog->setSizePolicy(sizePolicy);
54          gridLayout = new QGridLayout(AnimAddDialog);
55          gridLayout->setObjectName(QStringLiteral("gridLayout"));
56          horizontalLayout = new QHBoxLayout();
57          horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
58          buttonBox = new QDialogButtonBox(AnimAddDialog);
59          buttonBox->setObjectName(QStringLiteral("buttonBox"));
60          buttonBox->setOrientation(Qt::Horizontal);
61          buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
62          buttonBox->setCenterButtons(false);
63
64          horizontalLayout->addWidget(buttonBox);
65
66
67          gridLayout->addLayout(horizontalLayout, 5, 0, 1, 4);
68
69          horizontalSpacer_2 = new QSpacerItem(50, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
70
71          gridLayout->addItem(horizontalSpacer_2, 1, 3, 1, 1);
72
73          animBox = new QComboBox(AnimAddDialog);
74          animBox->setObjectName(QStringLiteral("animBox"));
75          QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Fixed);
76          sizePolicy1.setHorizontalStretch(0);
77          sizePolicy1.setVerticalStretch(0);
78          sizePolicy1.setHeightForWidth(animBox->sizePolicy().hasHeightForWidth());
79          animBox->setSizePolicy(sizePolicy1);
80
81          gridLayout->addWidget(animBox, 1, 2, 1, 1);
82
83          label_2 = new QLabel(AnimAddDialog);
84          label_2->setObjectName(QStringLiteral("label_2"));
85          QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Preferred);
86          sizePolicy2.setHorizontalStretch(0);
87          sizePolicy2.setVerticalStretch(0);
88          sizePolicy2.setHeightForWidth(label_2->sizePolicy().hasHeightForWidth());
89          label_2->setSizePolicy(sizePolicy2);
90          label_2->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
91
92          gridLayout->addWidget(label_2, 2, 1, 1, 1);
93
94          verticalSpacer_2 = new QSpacerItem(20, 0, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
95
96          gridLayout->addItem(verticalSpacer_2, 4, 2, 1, 1);
97
98          label = new QLabel(AnimAddDialog);
99          label->setObjectName(QStringLiteral("label"));
100          sizePolicy2.setHeightForWidth(label->sizePolicy().hasHeightForWidth());
101          label->setSizePolicy(sizePolicy2);
102          label->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
103
104          gridLayout->addWidget(label, 1, 1, 1, 1);
105
106          verticalSpacer_3 = new QSpacerItem(20, 0, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
107
108          gridLayout->addItem(verticalSpacer_3, 0, 2, 1, 1);
109
110          presetBox = new QComboBox(AnimAddDialog);
111          presetBox->setObjectName(QStringLiteral("presetBox"));
112          sizePolicy1.setHeightForWidth(presetBox->sizePolicy().hasHeightForWidth());
113          presetBox->setSizePolicy(sizePolicy1);
114
115          gridLayout->addWidget(presetBox, 2, 2, 1, 1);
116
117          horizontalSpacer = new QSpacerItem(50, 20, QSizePolicy::Fixed, QSizePolicy::Minimum
    );
118
119          gridLayout->addItem(horizontalSpacer, 1, 0, 1, 1);
120
121          previewBox = new QCheckBox(AnimAddDialog);
122          previewBox->setObjectName(QStringLiteral("previewBox"));
123          previewBox->setChecked(true);
124
125          gridLayout->addWidget(previewBox, 3, 2, 1, 1);
126
127
128          retranslateUi(AnimAddDialog);
129          QObject::connect(buttonBox, SIGNAL(accepted()), AnimAddDialog, SLOT(accept()));
130          QObject::connect(buttonBox, SIGNAL(rejected()), AnimAddDialog, SLOT(reject()));
131
```

```
132         QMetaObject::connectSlotsByName(AnimAddDialog);
133     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.86.3 Field Documentation

12.86.3.1 QComboBox* Ui_AnimAddDialog::animBox

Definition at line 35 of file ui_animadddialog.h.

Referenced by AnimAddDialog::AnimAddDialog(), AnimAddDialog::chosenScript(), and setupUi().

12.86.3.2 QDialogButtonBox* Ui_AnimAddDialog::buttonBox

Definition at line 33 of file ui_animadddialog.h.

Referenced by setupUi().

12.86.3.3 QGridLayout* Ui_AnimAddDialog::gridLayout

Definition at line 31 of file ui_animadddialog.h.

Referenced by setupUi().

12.86.3.4 QHBoxLayout* Ui_AnimAddDialog::horizontalLayout

Definition at line 32 of file ui_animadddialog.h.

Referenced by setupUi().

12.86.3.5 QSpacerItem* Ui_AnimAddDialog::horizontalSpacer

Definition at line 41 of file ui_animadddialog.h.

Referenced by setupUi().

12.86.3.6 QSpacerItem* Ui_AnimAddDialog::horizontalSpacer_2

Definition at line 34 of file ui_animadddialog.h.

Referenced by setupUi().

12.86.3.7 QLabel* Ui_AnimAddDialog::label

Definition at line 38 of file ui_animadddialog.h.

Referenced by retranslateUi(), and setupUi().

12.86.3.8 QLabel* Ui_AnimAddDialog::label_2

Definition at line 36 of file ui_animadddialog.h.

Referenced by retranslateUi(), and setupUi().

12.86.3.9 QComboBox* Ui_AnimAddDialog::presetBox

Definition at line 40 of file ui_animadddialog.h.

Referenced by AnimAddDialog::chosenPreset(), AnimAddDialog::on_animBox_activated(), AnimAddDialog::on_previewBox_clicked(), and setupUi().

12.86.3.10 QCheckBox* Ui_AnimAddDialog::previewBox

Definition at line 42 of file ui_animadddialog.h.

Referenced by retranslateUi(), and setupUi().

12.86.3.11 QSpacerItem* Ui_AnimAddDialog::verticalSpacer_2

Definition at line 37 of file ui_animadddialog.h.

Referenced by setupUi().

12.86.3.12 QSpacerItem* Ui_AnimAddDialog::verticalSpacer_3

Definition at line 39 of file ui_animadddialog.h.

Referenced by setupUi().

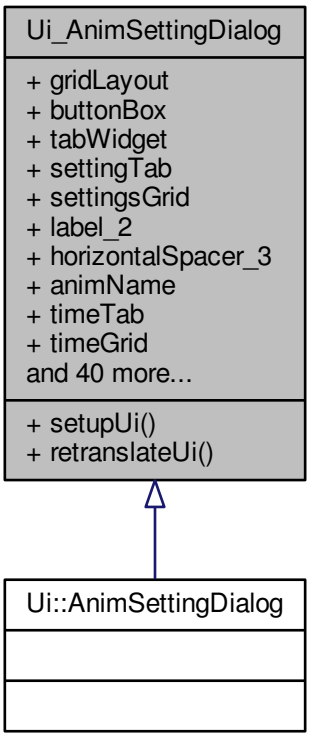
The documentation for this class was generated from the following file:

- src/ckb/ui_animadddialog.h

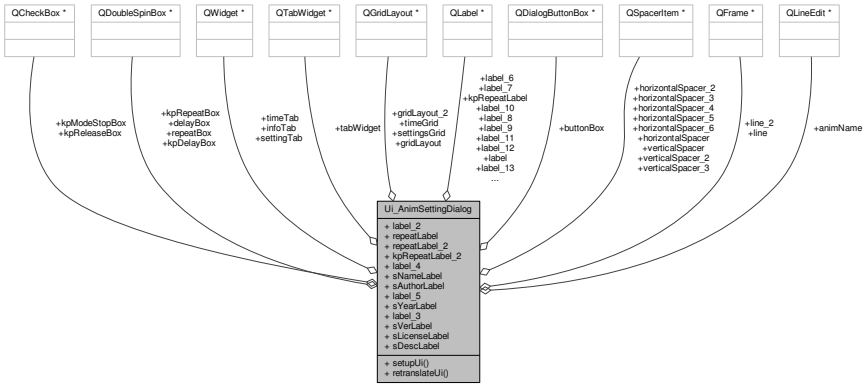
12.87 Ui_AnimSettingDialog Class Reference

`#include <src/ckb/ui_animsettingdialog.h>`

Inheritance diagram for Ui_AnimSettingDialog:



Collaboration diagram for Ui_AnimSettingDialog:



Public Member Functions

- void setupUi (QDialog *AnimSettingDialog)
- void retranslateUi (QDialog *AnimSettingDialog)

Data Fields

- QGridLayout * gridLayout
- QDialogButtonBox * buttonBox
- QTabWidget * tabWidget
- QWidget * settingTab
- QGridLayout * settingsGrid
- QLabel * label_2
- QSpacerItem * horizontalSpacer_3
- QLineEdit * animName
- QWidget * timeTab
- QGridLayout * timeGrid
- QSpacerItem * horizontalSpacer_5
- QLabel * kpRepeatLabel
- QLabel * repeatLabel
- QDoubleSpinBox * repeatBox
- QLabel * repeatLabel_2
- QFrame * line
- QLabel * kpRepeatLabel_2
- QLabel * label_4
- QLabel * label_8
- QDoubleSpinBox * kpDelayBox
- QLabel * label_13
- QLabel * label_12
- QDoubleSpinBox * delayBox
- QCheckBox * kpModeStopBox
- QLabel * label_10
- QSpacerItem * verticalSpacer_3
- QSpacerItem * horizontalSpacer_4
- QCheckBox * kpReleaseBox
- QLabel * label_6
- QDoubleSpinBox * kpRepeatBox
- QSpacerItem * verticalSpacer_2
- QFrame * line_2
- QSpacerItem * horizontalSpacer_6
- QWidget * infoTab
- QGridLayout * gridLayout_2
- QLabel * sNameLabel
- QLabel * label_7
- QLabel * label_11
- QLabel * sAuthorLabel
- QLabel * label_9
- QLabel * label_5
- QLabel * sYearLabel
- QLabel * label_3
- QSpacerItem * horizontalSpacer
- QSpacerItem * horizontalSpacer_2
- QLabel * sVerLabel
- QLabel * sLicenseLabel
- QLabel * label
- QSpacerItem * verticalSpacer
- QLabel * sDescLabel

12.87.1 Detailed Description

Definition at line 31 of file ui_animsettingdialog.h.

12.87.2 Member Function Documentation

12.87.2.1 void Ui_AnimSettingDialog::retranslateUi (QDialog * *AnimSettingDialog*) [inline]

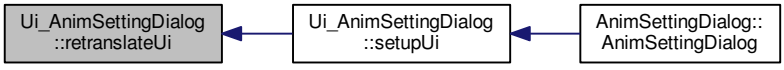
Definition at line 356 of file ui_animsettingdialog.h.

References infoTab, kpModeStopBox, kpReleaseBox, kpRepeatLabel, kpRepeatLabel_2, label, label_10, label_11, label_12, label_13, label_2, label_3, label_4, label_5, label_6, label_7, label_8, label_9, repeatLabel, repeatLabel_2, sAuthorLabel, sDescLabel, settingTab, sLicenseLabel, sNameLabel, sVerLabel, sYearLabel, tabWidget, and timeTab.

Referenced by setupUi().

```
357     {
358         AnimSettingDialog->setWindowTitle(QApplication::translate("AnimSettingDialog", "Dialog", Q_NULLPTR)
    );
359         label_2->setText(QApplication::translate("AnimSettingDialog", "Name:", Q_NULLPTR));
360         tabWidget->setTabText(tabWidget->indexOf(settingTab),
    QApplication::translate("AnimSettingDialog", "Settings", Q_NULLPTR));
361         kpRepeatLabel->setText(QApplication::translate("AnimSettingDialog", "Repeat every:",
    Q_NULLPTR));
362         repeatLabel->setText(QApplication::translate("AnimSettingDialog", "Repeat every:",
    Q_NULLPTR));
363         repeatLabel_2->setText(QApplication::translate("AnimSettingDialog", "seconds",
    Q_NULLPTR));
364         kpRepeatLabel_2->setText(QApplication::translate("AnimSettingDialog", "seconds",
    Q_NULLPTR));
365         label_4->setText(QApplication::translate("AnimSettingDialog", "<b>On mode start</b>",
    Q_NULLPTR));
366         label_8->setText(QApplication::translate("AnimSettingDialog", "seconds", Q_NULLPTR));
367         label_13->setText(QApplication::translate("AnimSettingDialog", "seconds", Q_NULLPTR));
368         label_12->setText(QApplication::translate("AnimSettingDialog", "Start delay:", Q_NULLPTR));
369         kpModeStopBox->setText(QApplication::translate("AnimSettingDialog", "Stop on key press
    ", Q_NULLPTR));
370         label_10->setText(QApplication::translate("AnimSettingDialog", "Start delay:", Q_NULLPTR));
371         kpReleaseBox->setText(QApplication::translate("AnimSettingDialog", "Stop on key release
    ", Q_NULLPTR));
372         label_6->setText(QApplication::translate("AnimSettingDialog", "<b>On key press</b>",
    Q_NULLPTR));
373         tabWidget->setTabText(tabWidget->indexOf(timeTab), QApplication::translate
    ("AnimSettingDialog", "Timing", Q_NULLPTR));
374         sNameLabel->setText(QApplication::translate("AnimSettingDialog", "Name", Q_NULLPTR));
375         label_7->setText(QApplication::translate("AnimSettingDialog", "Year:", Q_NULLPTR));
376         label_11->setText(QApplication::translate("AnimSettingDialog", "Description:", Q_NULLPTR));
377         sAuthorLabel->setText(QApplication::translate("AnimSettingDialog", "Author", Q_NULLPTR)
    );
378         label_9->setText(QApplication::translate("AnimSettingDialog", "License:", Q_NULLPTR));
379         label_5->setText(QApplication::translate("AnimSettingDialog", "Version:", Q_NULLPTR));
380         sYearLabel->setText(QApplication::translate("AnimSettingDialog", "Year", Q_NULLPTR));
381         label_3->setText(QApplication::translate("AnimSettingDialog", "Author:", Q_NULLPTR));
382         sVerLabel->setText(QApplication::translate("AnimSettingDialog", "Version", Q_NULLPTR));
383         sLicenseLabel->setText(QApplication::translate("AnimSettingDialog", "License",
    Q_NULLPTR));
384         label->setText(QApplication::translate("AnimSettingDialog", "Script name:", Q_NULLPTR));
385         sDescLabel->setText(QApplication::translate("AnimSettingDialog", "Description", Q_NULLPTR
    ));
386         tabWidget->setTabText(tabWidget->indexOf(infoTab), QApplication::translate
    ("AnimSettingDialog", "About", Q_NULLPTR));
387     } // retranslateUi
```

Here is the caller graph for this function:



12.87.2.2 void Ui_AnimSettingDialog::setupUi (QDialog * *AnimSettingDialog*) [inline]

Definition at line 85 of file ui_animsettingdialog.h.

References animName, buttonBox, delayBox, gridLayout, gridLayout_2, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, horizontalSpacer_4, horizontalSpacer_5, horizontalSpacer_6, infoTab, kpDelayBox, kpModeStopBox, kpReleaseBox, kpRepeatBox, kpRepeatLabel, kpRepeatLabel_2, label, label_10, label_11, label_12, label_13, label_2, label_3, label_4, label_5, label_6, label_7, label_8, label_9, line, line_2, repeatBox, repeatLabel, repeatLabel_2, retranslateUi(), sAuthorLabel, sDescLabel, settingsGrid, settingTab, sLicenseLabel, sNameLabel, sVerLabel, sYearLabel, tabWidget, timeGrid, timeTab, verticalSpacer, verticalSpacer_2, and verticalSpacer_3.

Referenced by AnimSettingDialog::AnimSettingDialog().

```
86     {
87         if (AnimSettingDialog->objectName().isEmpty())
88             AnimSettingDialog->setObjectName(QStringLiteral("AnimSettingDialog"));
89         AnimSettingDialog->resize(480, 457);
90         AnimSettingDialog->setMinimumSize(QSize(480, 0));
91         AnimSettingDialog->setFocusPolicy(Qt::ClickFocus);
92         gridLayout = new QGridLayout(AnimSettingDialog);
93         gridLayout->setObjectName(QStringLiteral("gridLayout"));
94         buttonBox = new QDialogButtonBox(AnimSettingDialog);
95         buttonBox->setObjectName(QStringLiteral("buttonBox"));
96         buttonBox->setOrientation(Qt::Horizontal);
97         buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
98 
99         gridLayout->addWidget(buttonBox, 2, 0, 1, 1);
100 
101         tabWidget = new QTabWidget(AnimSettingDialog);
102         tabWidget->setObjectName(QStringLiteral("tabWidget"));
103         settingTab = new QWidget();
104         settingTab->setObjectName(QStringLiteral("settingTab"));
105         settingsGrid = new QGridLayout(settingTab);
106         settingsGrid->setSpacing(12);
107         settingsGrid->setObjectName(QStringLiteral("settingsGrid"));
108         settingsGrid->setContentsMargins(12, 12, 12, 12);
109         label_2 = new QLabel(settingTab);
110         label_2->setObjectName(QStringLiteral("label_2"));
111 
112         settingsGrid->addWidget(label_2, 0, 1, 1, 1);
113 
114         horizontalSpacer_3 = new QSpacerItem(5, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
115 
116         settingsGrid->addItem(horizontalSpacer_3, 0, 2, 1, 1);
117 
118         animName = new QLineEdit(settingTab);
119         animName->setObjectName(QStringLiteral("animName"));
120 
121         settingsGrid->addWidget(animName, 0, 3, 1, 3);
122 
123         tabWidget->addTab(settingTab, QString());
124         timeTab = new QWidget();
125         timeTab->setObjectName(QStringLiteral("timeTab"));
126         timeGrid = new QGridLayout(timeTab);
127         timeGrid->setSpacing(12);
128         timeGrid->setObjectName(QStringLiteral("timeGrid"));
129         timeGrid->setContentsMargins(12, 12, 12, 12);
130         horizontalSpacer_5 = new QSpacerItem(15, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
131 
132         timeGrid->addItem(horizontalSpacer_5, 3, 2, 1, 1);
133 
134         kpRepeatLabel = new QLabel(timeTab);
135         kpRepeatLabel->setObjectName(QStringLiteral("kpRepeatLabel"));
136 
137         timeGrid->addWidget(kpRepeatLabel, 11, 1, 1, 1);
138 
139         repeatLabel = new QLabel(timeTab);
140         repeatLabel->setObjectName(QStringLiteral("repeatLabel"));
141 
142         timeGrid->addWidget(repeatLabel, 3, 1, 1, 1);
143 
144         repeatBox = new QDoubleSpinBox(timeTab);
145         repeatBox->setObjectName(QStringLiteral("repeatBox"));
146         repeatBox->setDecimals(1);
147         repeatBox->setMinimum(0.1);
148         repeatBox->setMaximum(86400);
149 
150         timeGrid->addWidget(repeatBox, 3, 3, 1, 1);
151 
152         repeatLabel_2 = new QLabel(timeTab);
153         repeatLabel_2->setObjectName(QStringLiteral("repeatLabel_2"));
```

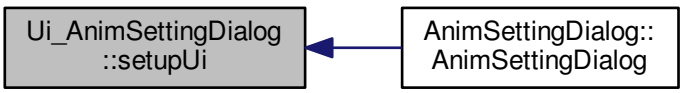
```
154
155         timeGrid->addWidget(repeatLabel_2, 3, 4, 1, 1);
156
157         line = new QFrame(timeTab);
158         line->setObjectName(QStringLiteral("line"));
159         line->setFrameShape(QFrame::HLine);
160         line->setFrameShadow(QFrame::Sunken);
161
162         timeGrid->addWidget(line, 1, 0, 1, 6);
163
164         kpRepeatLabel_2 = new QLabel(timeTab);
165         kpRepeatLabel_2->setObjectName(QStringLiteral("kpRepeatLabel_2"));
166
167         timeGrid->addWidget(kpRepeatLabel_2, 11, 4, 1, 1);
168
169         label_4 = new QLabel(timeTab);
170         label_4->setObjectName(QStringLiteral("label_4"));
171
172         timeGrid->addWidget(label_4, 0, 0, 1, 6);
173
174         label_8 = new QLabel(timeTab);
175         label_8->setObjectName(QStringLiteral("label_8"));
176
177         timeGrid->addWidget(label_8, 2, 4, 1, 1);
178
179         kpDelayBox = new QDoubleSpinBox(timeTab);
180         kpDelayBox->setObjectName(QStringLiteral("kpDelayBox"));
181         kpDelayBox->setDecimals(1);
182         kpDelayBox->setMaximum(86400);
183
184         timeGrid->addWidget(kpDelayBox, 10, 3, 1, 1);
185
186         label_13 = new QLabel(timeTab);
187         label_13->setObjectName(QStringLiteral("label_13"));
188
189         timeGrid->addWidget(label_13, 10, 4, 1, 1);
190
191         label_12 = new QLabel(timeTab);
192         label_12->setObjectName(QStringLiteral("label_12"));
193
194         timeGrid->addWidget(label_12, 10, 1, 1, 1);
195
196         delayBox = new QDoubleSpinBox(timeTab);
197         delayBox->setObjectName(QStringLiteral("delayBox"));
198         delayBox->setDecimals(1);
199         delayBox->setMaximum(86400);
200
201         timeGrid->addWidget(delayBox, 2, 3, 1, 1);
202
203         kpModeStopBox = new QCheckBox(timeTab);
204         kpModeStopBox->setObjectName(QStringLiteral("kpModeStopBox"));
205
206         timeGrid->addWidget(kpModeStopBox, 5, 1, 1, 3);
207
208         label_10 = new QLabel(timeTab);
209         label_10->setObjectName(QStringLiteral("label_10"));
210
211         timeGrid->addWidget(label_10, 2, 1, 1, 1);
212
213         verticalSpacer_3 = new QSpacerItem(0, 10, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
214
215         timeGrid->addItem(verticalSpacer_3, 7, 1, 1, 1);
216
217         horizontalSpacer_4 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
218
219         timeGrid->addItem(horizontalSpacer_4, 2, 5, 1, 1);
220
221         kpReleaseBox = new QCheckBox(timeTab);
222         kpReleaseBox->setObjectName(QStringLiteral("kpReleaseBox"));
223
224         timeGrid->addWidget(kpReleaseBox, 14, 1, 1, 3);
225
226         label_6 = new QLabel(timeTab);
227         label_6->setObjectName(QStringLiteral("label_6"));
228
229         timeGrid->addWidget(label_6, 8, 0, 1, 6);
230
231         kpRepeatBox = new QDoubleSpinBox(timeTab);
232         kpRepeatBox->setObjectName(QStringLiteral("kpRepeatBox"));
233         kpRepeatBox->setDecimals(1);
234         kpRepeatBox->setMinimum(0.1);
235         kpRepeatBox->setMaximum(86400);
236
237         timeGrid->addWidget(kpRepeatBox, 11, 3, 1, 1);
238
```

```
239         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
240
241         timeGrid->addItem(verticalSpacer_2, 15, 1, 1, 1);
242
243         line_2 = new QFrame(timeTab);
244         line_2->setObjectName(QStringLiteral("line_2"));
245         line_2->setFrameShape(QFrame::HLine);
246         line_2->setFrameShadow(QFrame::Sunken);
247
248         timeGrid->addWidget(line_2, 9, 0, 1, 6);
249
250         horizontalSpacer_6 = new QSpacerItem(0, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
251
252         timeGrid->addItem(horizontalSpacer_6, 12, 2, 1, 1);
253
254         tabWidget->addTab(timeTab, QString());
255         infoTab = new QWidget();
256         infoTab->setObjectName(QStringLiteral("infoTab"));
257         gridLayout_2 = new QGridLayout(infoTab);
258         gridLayout_2->setSpacing(12);
259         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
260         gridLayout_2->setContentsMargins(12, 12, 12, 12);
261         sNameLabel = new QLabel(infoTab);
262         sNameLabel->setObjectName(QStringLiteral("sNameLabel"));
263
264         gridLayout_2->addWidget(sNameLabel, 0, 2, 1, 1);
265
266         label_7 = new QLabel(infoTab);
267         label_7->setObjectName(QStringLiteral("label_7"));
268
269         gridLayout_2->addWidget(label_7, 3, 0, 1, 1);
270
271         label_11 = new QLabel(infoTab);
272         label_11->setObjectName(QStringLiteral("label_11"));
273
274         gridLayout_2->addWidget(label_11, 5, 0, 1, 1);
275
276         sAuthorLabel = new QLabel(infoTab);
277         sAuthorLabel->setObjectName(QStringLiteral("sAuthorLabel"));
278
279         gridLayout_2->addWidget(sAuthorLabel, 2, 2, 1, 1);
280
281         label_9 = new QLabel(infoTab);
282         label_9->setObjectName(QStringLiteral("label_9"));
283
284         gridLayout_2->addWidget(label_9, 4, 0, 1, 1);
285
286         label_5 = new QLabel(infoTab);
287         label_5->setObjectName(QStringLiteral("label_5"));
288
289         gridLayout_2->addWidget(label_5, 1, 0, 1, 1);
290
291         sYearLabel = new QLabel(infoTab);
292         sYearLabel->setObjectName(QStringLiteral("sYearLabel"));
293
294         gridLayout_2->addWidget(sYearLabel, 3, 2, 1, 1);
295
296         label_3 = new QLabel(infoTab);
297         label_3->setObjectName(QStringLiteral("label_3"));
298
299         gridLayout_2->addWidget(label_3, 2, 0, 1, 1);
300
301         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
302
303         gridLayout_2->addItem(horizontalSpacer, 0, 3, 1, 1);
304
305         horizontalSpacer_2 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
306
307         gridLayout_2->addItem(horizontalSpacer_2, 0, 1, 1, 1);
308
309         sVerLabel = new QLabel(infoTab);
310         sVerLabel->setObjectName(QStringLiteral("sVerLabel"));
311
312         gridLayout_2->addWidget(sVerLabel, 1, 2, 1, 1);
313
314         sLicenseLabel = new QLabel(infoTab);
315         sLicenseLabel->setObjectName(QStringLiteral("sLicenseLabel"));
316
317         gridLayout_2->addWidget(sLicenseLabel, 4, 2, 1, 1);
318
319         label = new QLabel(infoTab);
320         label->setObjectName(QStringLiteral("label"));
321
```

```
322         gridLayout_2->addWidget(label, 0, 0, 1, 1);
323
324         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
325
326         gridLayout_2->addItem(verticalSpacer, 6, 0, 1, 1);
327
328         sDescLabel = new QLabel(infoTab);
329         sDescLabel->setObjectName(QStringLiteral("sDescLabel"));
330         sDescLabel->setAlignment(Qt::AlignLeading|Qt::AlignLeft|Qt::AlignTop);
331
332         gridLayout_2->addWidget(sDescLabel, 5, 2, 2, 2);
333
334         tabWidget->addTab(infoTab, QString());
335
336         gridLayout->addWidget(tabWidget, 0, 0, 1, 1);
337
338         QWidget::setTabOrder(tabWidget, animName);
339         QWidget::setTabOrder(animName, delayBox);
340         QWidget::setTabOrder(delayBox, repeatBox);
341         QWidget::setTabOrder(repeatBox, kpModeStopBox);
342         QWidget::setTabOrder(kpModeStopBox, kpDelayBox);
343         QWidget::setTabOrder(kpDelayBox, kpRepeatBox);
344         QWidget::setTabOrder(kpRepeatBox, kpReleaseBox);
345
346         retranslateUi(AnimSettingDialog);
347         QObject::connect(buttonBox, SIGNAL(accepted()), AnimSettingDialog, SLOT(accept()));
348         QObject::connect(buttonBox, SIGNAL(rejected()), AnimSettingDialog, SLOT(reject()));
349
350         tabWidget->setCurrentIndex(0);
351
352
353         QMetaObject::connectSlotsByName(AnimSettingDialog);
354     } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.87.3 Field Documentation

12.87.3.1 QLineEdit* Ui_AnimSettingDialog::animName

Definition at line 41 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), AnimSettingDialog::name(), and setupUi().

12.87.3.2 QDialogButtonBox* Ui_AnimSettingDialog::buttonBox

Definition at line 35 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.3 QDoubleSpinBox* Ui_AnimSettingDialog::delayBox

Definition at line 56 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and setupUi().

12.87.3.4 QGridLayout* Ui_AnimSettingDialog::gridLayout

Definition at line 34 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.5 QGridLayout* Ui_AnimSettingDialog::gridLayout_2

Definition at line 68 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.6 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer

Definition at line 77 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.7 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_2

Definition at line 78 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.8 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_3

Definition at line 40 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.9 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_4

Definition at line 60 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.10 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_5

Definition at line 44 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.11 QSpacerItem* Ui_AnimSettingDialog::horizontalSpacer_6

Definition at line 66 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.12 QWidget* Ui_AnimSettingDialog::infoTab

Definition at line 67 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.13 QDoubleSpinBox* Ui_AnimSettingDialog::kpDelayBox

Definition at line 53 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and setupUi().

12.87.3.14 QCheckBox* Ui_AnimSettingDialog::kpModeStopBox

Definition at line 57 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.15 QCheckBox* Ui_AnimSettingDialog::kpReleaseBox

Definition at line 61 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.16 QDoubleSpinBox* Ui_AnimSettingDialog::kpRepeatBox

Definition at line 63 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and setupUi().

12.87.3.17 QLabel* Ui_AnimSettingDialog::kpRepeatLabel

Definition at line 45 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.18 QLabel* Ui_AnimSettingDialog::kpRepeatLabel_2

Definition at line 50 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.19 QLabel* Ui_AnimSettingDialog::label

Definition at line 81 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.20 QLabel* Ui_AnimSettingDialog::label_10

Definition at line 58 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.21 QLabel* Ui_AnimSettingDialog::label_11

Definition at line 71 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.22 QLabel* Ui_AnimSettingDialog::label_12

Definition at line 55 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.23 QLabel* Ui_AnimSettingDialog::label_13

Definition at line 54 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.24 QLabel* Ui_AnimSettingDialog::label_2

Definition at line 39 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.25 QLabel* Ui_AnimSettingDialog::label_3

Definition at line 76 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.26 QLabel* Ui_AnimSettingDialog::label_4

Definition at line 51 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.27 QLabel* Ui_AnimSettingDialog::label_5

Definition at line 74 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.28 QLabel* Ui_AnimSettingDialog::label_6

Definition at line 62 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.29 QLabel* Ui_AnimSettingDialog::label_7

Definition at line 70 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.30 QLabel* Ui_AnimSettingDialog::label_8

Definition at line 52 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.31 QLabel* Ui_AnimSettingDialog::label_9

Definition at line 73 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.32 QFrame* Ui_AnimSettingDialog::line

Definition at line 49 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.33 QFrame* Ui_AnimSettingDialog::line_2

Definition at line 65 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.34 QDoubleSpinBox* Ui_AnimSettingDialog::repeatBox

Definition at line 47 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and setupUi().

12.87.3.35 QLabel* Ui_AnimSettingDialog::repeatLabel

Definition at line 46 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.36 QLabel* Ui_AnimSettingDialog::repeatLabel_2

Definition at line 48 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.37 QLabel* Ui_AnimSettingDialog::sAuthorLabel

Definition at line 72 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.38 QLabel* Ui_AnimSettingDialog::sDescLabel

Definition at line 83 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.39 QGridLayout* Ui_AnimSettingDialog::settingsGrid

Definition at line 38 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and setupUi().

12.87.3.40 QWidget* Ui_AnimSettingDialog::settingTab

Definition at line 37 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.41 QLabel* Ui_AnimSettingDialog::sLicenseLabel

Definition at line 80 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.42 QLabel* Ui_AnimSettingDialog::sNameLabel

Definition at line 69 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.43 QLabel* Ui_AnimSettingDialog::sVerLabel

Definition at line 79 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.44 QLabel* Ui_AnimSettingDialog::sYearLabel

Definition at line 75 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), retranslateUi(), and setupUi().

12.87.3.45 QTabWidget* Ui_AnimSettingDialog::tabWidget

Definition at line 36 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.46 QGridLayout* Ui_AnimSettingDialog::timeGrid

Definition at line 43 of file ui_animsettingdialog.h.

Referenced by AnimSettingDialog::AnimSettingDialog(), and setupUi().

12.87.3.47 QWidget* Ui_AnimSettingDialog::timeTab

Definition at line 42 of file ui_animsettingdialog.h.

Referenced by retranslateUi(), and setupUi().

12.87.3.48 QSpacerItem* Ui_AnimSettingDialog::verticalSpacer

Definition at line 82 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.49 QSpacerItem* Ui_AnimSettingDialog::verticalSpacer_2

Definition at line 64 of file ui_animsettingdialog.h.

Referenced by setupUi().

12.87.3.50 QSpacerItem* Ui_AnimSettingDialog::verticalSpacer_3

Definition at line 59 of file ui_animsettingdialog.h.

Referenced by setupUi().

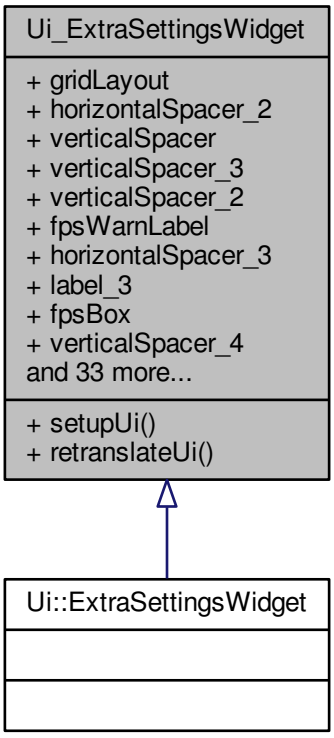
The documentation for this class was generated from the following file:

- src/ckb/ui_animsettingdialog.h

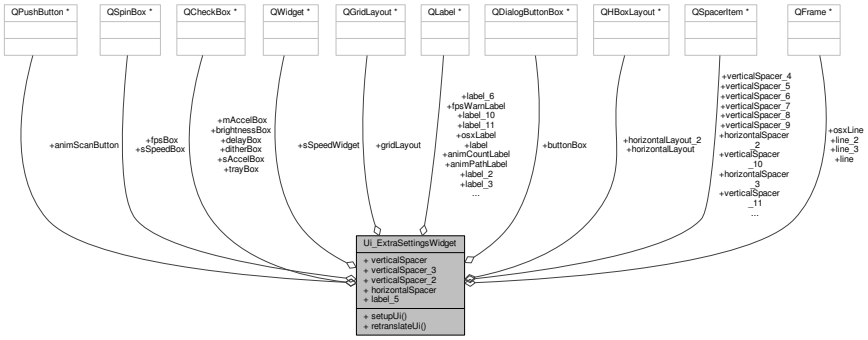
12.88 Ui_ExtraSettingsWidget Class Reference

`#include <src/ckb/ui_extrasettingswidget.h>`

Inheritance diagram for Ui_ExtraSettingsWidget:



Collaboration diagram for Ui_ExtraSettingsWidget:



Public Member Functions

- void setupUi (QDialog *ExtraSettingsWidget)
- void retranslateUi (QDialog *ExtraSettingsWidget)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer_2
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_3
- QSpacerItem * verticalSpacer_2
- QLabel * fpsWarnLabel
- QSpacerItem * horizontalSpacer_3
- QLabel * label_3
- QSpinBox * fpsBox
- QSpacerItem * verticalSpacer_4
- QSpacerItem * verticalSpacer_9
- QLabel * label_6
- QSpacerItem * verticalSpacer_6
- QSpacerItem * verticalSpacer_5
- QCheckBox * sAccelBox
- QSpacerItem * verticalSpacer_7
- QSpacerItem * verticalSpacer_8
- QSpacerItem * verticalSpacer_10
- QSpacerItem * verticalSpacer_11
- QSpacerItem * horizontalSpacer
- QFrame * line_2
- QHBoxLayout * horizontalLayout
- QLabel * animPathLabel
- QLabel * animCountLabel
- QPushButton * animScanButton
- QCheckBox * brightnessBox
- QCheckBox * trayBox
- QLabel * label
- QFrame * line
- QLabel * label_5
- QFrame * line_3
- QLabel * label_2
- QCheckBox * ditherBox
- QCheckBox * delayBox
- QLabel * osxLabel
- QFrame * osxLine
- QCheckBox * mAccelBox
- QWidget * sSpeedWidget
- QHBoxLayout * horizontalLayout_2
- QLabel * label_10
- QSpinBox * sSpeedBox
- QLabel * label_11
- QDialogButtonBox * buttonBox

12.88.1 Detailed Description

Definition at line 31 of file ui_extrasettingswidget.h.

12.88.2 Member Function Documentation

12.88.2.1 void Ui_ExtraSettingsWidget::retranslateUi (QDialog * *ExtraSettingsWidget*) [inline]

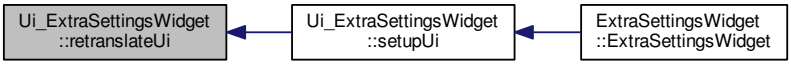
Definition at line 330 of file ui_extrasettingswidget.h.

References animCountLabel, animPathLabel, animScanButton, brightnessBox, delayBox, ditherBox, fpsWarnLabel, label, label_10, label_11, label_2, label_3, label_5, label_6, mAccelBox, osxLabel, sAccelBox, and trayBox.

Referenced by setupUi().

```
331     {
332         ExtraSettingsWidget->setWindowTitle(QApplication::translate("ExtraSettingsWidget", "Application
      Settings", Q_NULLPTR));
333         fpsWarnLabel->setText(QApplication::translate("ExtraSettingsWidget", "Warning: high
      frame rates may cause stability issues", Q_NULLPTR));
334         label_3->setText(QApplication::translate("ExtraSettingsWidget", "Frame rate (FPS):",
     Q_NULLPTR));
335         label_6->setText(QApplication::translate("ExtraSettingsWidget", "Location:", Q_NULLPTR));
336 #ifndef QT_NO_TOOLTIP
337         sAccelBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "Try this if you're
      having problems with the scroll wheel.", Q_NULLPTR));
338 #endif // QT_NO_TOOLTIP
339         sAccelBox->setText(QApplication::translate("ExtraSettingsWidget", "Disable scroll
     acceleration", Q_NULLPTR));
340         animPathLabel->setText(QApplication::translate("ExtraSettingsWidget", "
     /path/to/animations", Q_NULLPTR));
341         animCountLabel->setText(QApplication::translate("ExtraSettingsWidget", "0 animations
     found", Q_NULLPTR));
342         animScanButton->setText(QApplication::translate("ExtraSettingsWidget", "Re-scan",
     Q_NULLPTR));
343 #ifndef QT_NO_TOOLTIP
344         brightnessBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "By default,
      the same brightness level will be applied to all profiles and all devices. Enable this to store it with the
      lighting mode instead.", Q_NULLPTR));
345 #endif // QT_NO_TOOLTIP
346         brightnessBox->setText(QApplication::translate("ExtraSettingsWidget", "Set brightness
     per-mode", Q_NULLPTR));
347 #ifndef QT_NO_TOOLTIP
348         trayBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "The tray icon will not
     be displayed. The application will still run in the background; re-launch the app to see the GUI again.",
     Q_NULLPTR));
349 #endif // QT_NO_TOOLTIP
350         trayBox->setText(QApplication::translate("ExtraSettingsWidget", "Hide tray icon", Q_NULLPTR)
    );
351         label->setText(QApplication::translate("ExtraSettingsWidget", "Behavior", Q_NULLPTR));
352         label_5->setText(QApplication::translate("ExtraSettingsWidget", "Animation scripts",
     Q_NULLPTR));
353         label_2->setText(QApplication::translate("ExtraSettingsWidget", "Hardware", Q_NULLPTR));
354 #ifndef QT_NO_TOOLTIP
355         ditherBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "May improve
     appearance on some keyboards.", Q_NULLPTR));
356 #endif // QT_NO_TOOLTIP
357         ditherBox->setText(QApplication::translate("ExtraSettingsWidget", "Use spatial dithering
     to simulate extra color resolution", Q_NULLPTR));
358 #ifndef QT_NO_TOOLTIP
359         delayBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "When using macros with
      strings longer than 25 chars, some OS may lose characters (e.g. Mint 17.2). Select to prevent that bug.",
     Q_NULLPTR));
360 #endif // QT_NO_TOOLTIP
361         delayBox->setText(QApplication::translate("ExtraSettingsWidget", "Use delay for very long
     macros", Q_NULLPTR));
362         osxLabel->setText(QApplication::translate("ExtraSettingsWidget", "OS X tweaks", Q_NULLPTR))
    ;
363 #ifndef QT_NO_TOOLTIP
364         mAccelBox->setToolTip(QApplication::translate("ExtraSettingsWidget", "Try this if you're
     having problems with mouse movement.", Q_NULLPTR));
365 #endif // QT_NO_TOOLTIP
366         mAccelBox->setText(QApplication::translate("ExtraSettingsWidget", "Disable mouse
     acceleration", Q_NULLPTR));
367         label_10->setText(QApplication::translate("ExtraSettingsWidget", "Scroll", Q_NULLPTR));
368         label_11->setText(QApplication::translate("ExtraSettingsWidget", "line(s) at a time",
     Q_NULLPTR));
369     } // retranslateUi
```

Here is the caller graph for this function:



12.88.2.2 void Ui_ExtraSettingsWidget::setupUi (QDialog * *ExtraSettingsWidget*) [inline]

Definition at line 78 of file ui_extrasettingswidget.h.

References animCountLabel, animPathLabel, animScanButton, brightnessBox, buttonBox, delayBox, ditherBox, fpsBox, fpsWarnLabel, gridLayout, horizontalLayout, horizontalLayout_2, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, label, label_10, label_11, label_2, label_3, label_5, label_6, line, line_2, line_3, mAccelBox, osxLabel, osxLine, retranslateUi(), sAccelBox, sSpeedBox, sSpeedWidget, trayBox, verticalSpacer, verticalSpacer_10, verticalSpacer_11, verticalSpacer_2, verticalSpacer_3, verticalSpacer_4, verticalSpacer_5, verticalSpacer_6, verticalSpacer_7, verticalSpacer_8, and verticalSpacer_9.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget().

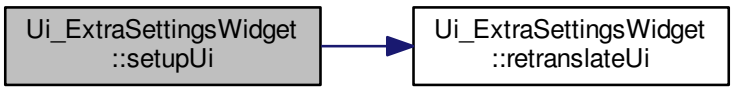
```
79     {
80         if (ExtraSettingsWidget->objectName().isEmpty())
81             ExtraSettingsWidget->setObjectName(QStringLiteral("ExtraSettingsWidget"));
82         ExtraSettingsWidget->resize(650, 515);
83         gridLayout = new QGridLayout(ExtraSettingsWidget);
84         gridLayout->setObjectName(QStringLiteral("gridLayout"));
85         horizontalSpacer_2 = new QSpacerItem(30, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
86
87         gridLayout->addItem(horizontalSpacer_2, 12, 2, 1, 1);
88
89         verticalSpacer = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed);
90
91         gridLayout->addItem(verticalSpacer, 2, 0, 1, 1);
92
93         verticalSpacer_3 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
94
95         gridLayout->addItem(verticalSpacer_3, 3, 0, 1, 1);
96
97         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
98
99         gridLayout->addItem(verticalSpacer_2, 19, 5, 1, 1);
100
101         fpsWarnLabel = new QLabel(ExtraSettingsWidget);
102         fpsWarnLabel->setObjectName(QStringLiteral("fpsWarnLabel"));
103         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
104         sizePolicy.setHorizontalStretch(0);
105         sizePolicy.setVerticalStretch(0);
106         sizePolicy.setHeightForWidth(fpsWarnLabel->sizePolicy().hasHeightForWidth());
107         fpsWarnLabel->setSizePolicy(sizePolicy);
108
109         gridLayout->addWidget(fpsWarnLabel, 12, 5, 1, 1);
110
111         horizontalSpacer_3 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
112
113         gridLayout->addItem(horizontalSpacer_3, 12, 4, 1, 1);
114
115         label_3 = new QLabel(ExtraSettingsWidget);
116         label_3->setObjectName(QStringLiteral("label_3"));
117
118         gridLayout->addWidget(label_3, 12, 1, 1, 1);
119
120         fpsBox = new QSpinBox(ExtraSettingsWidget);
121         fpsBox->setObjectName(QStringLiteral("fpsBox"));
122         fpsBox->setMinimum(5);
123         fpsBox->setMaximum(60);
124         fpsBox->setValue(30);
125
```

```
126         gridLayout->addWidget(fpsBox, 12, 3, 1, 1);
127 
128         verticalSpacer_4 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
129 
130         gridLayout->addItem(verticalSpacer_4, 7, 0, 1, 1);
131 
132         verticalSpacer_9 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
133 
134         gridLayout->addItem(verticalSpacer_9, 12, 0, 1, 1);
135 
136         label_6 = new QLabel(ExtraSettingsWidget);
137         label_6->setObjectName(QStringLiteral("label_6"));
138 
139         gridLayout->addWidget(label_6, 7, 1, 1, 1);
140 
141         verticalSpacer_6 = new QSpacerItem(20, 10, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
142 
143         gridLayout->addItem(verticalSpacer_6, 8, 1, 1, 1);
144 
145         verticalSpacer_5 = new QSpacerItem(20, 10, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
146 
147         gridLayout->addItem(verticalSpacer_5, 4, 1, 1, 1);
148 
149         sAccelBox = new QCheckBox(ExtraSettingsWidget);
150         sAccelBox->setObjectName(QStringLiteral("sAccelBox"));
151 
152         gridLayout->addWidget(sAccelBox, 18, 1, 1, 4);
153 
154         verticalSpacer_7 = new QSpacerItem(20, 10, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
155 
156         gridLayout->addItem(verticalSpacer_7, 14, 1, 1, 1);
157 
158         verticalSpacer_8 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
159 
160         gridLayout->addItem(verticalSpacer_8, 13, 0, 1, 1);
161 
162         verticalSpacer_10 = new QSpacerItem(0, 28, QSizePolicy::Minimum,
    QSizePolicy::Fixed);
163 
164         gridLayout->addItem(verticalSpacer_10, 17, 0, 1, 1);
165 
166         verticalSpacer_11 = new QSpacerItem(0, 28, QSizePolicy::Minimum,
    QSizePolicy::Fixed);
167 
168         gridLayout->addItem(verticalSpacer_11, 18, 0, 1, 1);
169 
170         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
171 
172         gridLayout->addItem(horizontalSpacer, 12, 6, 1, 1);
173 
174         line_2 = new QFrame(ExtraSettingsWidget);
175         line_2->setObjectName(QStringLiteral("line_2"));
176         line_2->setFrameShape(QFrame::HLine);
177         line_2->setFrameShadow(QFrame::Sunken);
178 
179         gridLayout->addWidget(line_2, 10, 0, 1, 7);
180 
181         horizontalLayout = new QHBoxLayout();
182         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
183         animPathLabel = new QLabel(ExtraSettingsWidget);
184         animPathLabel->setObjectName(QStringLiteral("animPathLabel"));
185 
186         horizontalLayout->addWidget(animPathLabel);
187 
188         animCountLabel = new QLabel(ExtraSettingsWidget);
189         animCountLabel->setObjectName(QStringLiteral("animCountLabel"));
190         sizePolicy.setHeightForWidth(animCountLabel->sizePolicy().hasHeightForWidth());
191         animCountLabel->setSizePolicy(sizePolicy);
192         animCountLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
193 
194         horizontalLayout->addWidget(animCountLabel);
195 
196         animScanButton = new QPushButton(ExtraSettingsWidget);
197         animScanButton->setObjectName(QStringLiteral("animScanButton"));
198 
199         horizontalLayout->addWidget(animScanButton);
200 
201 
202         gridLayout->addLayout(horizontalLayout, 7, 3, 1, 4);
203 
```

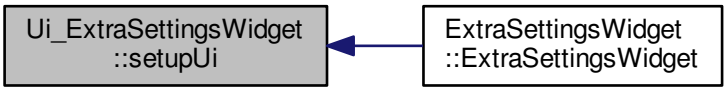
```
204         brightnessBox = new QCheckBox(ExtraSettingsWidget);
205         brightnessBox->setObjectName(QStringLiteral("brightnessBox"));
206 
207         gridLayout->addWidget(brightnessBox, 3, 1, 1, 6);
208 
209         trayBox = new QCheckBox(ExtraSettingsWidget);
210         trayBox->setObjectName(QStringLiteral("trayBox"));
211 
212         gridLayout->addWidget(trayBox, 2, 1, 1, 6);
213 
214         label = new QLabel(ExtraSettingsWidget);
215         label->setObjectName(QStringLiteral("label"));
216         QFont font;
217         font.setBold(true);
218         font.setWeight(75);
219         label->setFont(font);
220 
221         gridLayout->addWidget(label, 0, 0, 1, 7);
222 
223         line = new QFrame(ExtraSettingsWidget);
224         line->setObjectName(QStringLiteral("line"));
225         line->setFrameShape(QFrame::HLine);
226         line->setFrameShadow(QFrame::Sunken);
227 
228         gridLayout->addWidget(line, 1, 0, 1, 7);
229 
230         label_5 = new QLabel(ExtraSettingsWidget);
231         label_5->setObjectName(QStringLiteral("label_5"));
232         label_5->setFont(font);
233 
234         gridLayout->addWidget(label_5, 5, 0, 1, 7);
235 
236         line_3 = new QFrame(ExtraSettingsWidget);
237         line_3->setObjectName(QStringLiteral("line_3"));
238         line_3->setFrameShape(QFrame::HLine);
239         line_3->setFrameShadow(QFrame::Sunken);
240 
241         gridLayout->addWidget(line_3, 6, 0, 1, 7);
242 
243         label_2 = new QLabel(ExtraSettingsWidget);
244         label_2->setObjectName(QStringLiteral("label_2"));
245         label_2->setFont(font);
246 
247         gridLayout->addWidget(label_2, 9, 0, 1, 7);
248 
249         ditherBox = new QCheckBox(ExtraSettingsWidget);
250         ditherBox->setObjectName(QStringLiteral("ditherBox"));
251 
252         gridLayout->addWidget(ditherBox, 13, 1, 1, 6);
253 
254         delayBox = new QCheckBox(ExtraSettingsWidget);
255         delayBox->setObjectName(QStringLiteral("delayBox"));
256 
257         gridLayout->addWidget(delayBox, 14, 1, 1, 6);
258 
259         osxLabel = new QLabel(ExtraSettingsWidget);
260         osxLabel->setObjectName(QStringLiteral("osxLabel"));
261         osxLabel->setFont(font);
262 
263         gridLayout->addWidget(osxLabel, 15, 0, 1, 7);
264 
265         osxLine = new QFrame(ExtraSettingsWidget);
266         osxLine->setObjectName(QStringLiteral("osxLine"));
267         osxLine->setFrameShape(QFrame::HLine);
268         osxLine->setFrameShadow(QFrame::Sunken);
269 
270         gridLayout->addWidget(osxLine, 16, 0, 1, 7);
271 
272         mAccelBox = new QCheckBox(ExtraSettingsWidget);
273         mAccelBox->setObjectName(QStringLiteral("mAccelBox"));
274 
275         gridLayout->addWidget(mAccelBox, 17, 1, 1, 6);
276 
277         sSpeedWidget = new QWidget(ExtraSettingsWidget);
278         sSpeedWidget->setObjectName(QStringLiteral("sSpeedWidget"));
279         sizePolicy.setHeightForWidth(sSpeedWidget->sizePolicy().hasHeightForWidth());
280         sSpeedWidget->setSizePolicy(sizePolicy);
281         horizontalLayout_2 = new QHBoxLayout(sSpeedWidget);
282         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
283         horizontalLayout_2->setContentsMargins(0, 0, 0, 0);
284         label_10 = new QLabel(sSpeedWidget);
285         label_10->setObjectName(QStringLiteral("label_10"));
286 
287         horizontalLayout_2->addWidget(label_10);
288 
289         sSpeedBox = new QSpinBox(sSpeedWidget);
290         sSpeedBox->setObjectName(QStringLiteral("sSpeedBox"));
```

```
291         sSpeedBox->setMinimum(1);
292         sSpeedBox->setMaximum(10);
293         sSpeedBox->setValue(3);
294 
295         horizontalLayout_2->addWidget(sSpeedBox);
296 
297         label_11 = new QLabel(sSpeedWidget);
298         label_11->setObjectName(QStringLiteral("label_11"));
299         sizePolicy.setHeightForWidth(label_11->sizePolicy().hasHeightForWidth());
300         label_11->setSizePolicy(sizePolicy);
301 
302         horizontalLayout_2->addWidget(label_11);
303 
304 
305         gridLayout->addWidget(sSpeedWidget, 18, 5, 1, 2);
306 
307         buttonBox = new QDialogButtonBox(ExtraSettingsWidget);
308         buttonBox->setObjectName(QStringLiteral("buttonBox"));
309         buttonBox->setOrientation(Qt::Horizontal);
310         buttonBox->setStandardButtons(QDialogButtonBox::Ok);
311 
312         gridLayout->addWidget(buttonBox, 20, 0, 1, 7);
313 
314         QWidget::setTabOrder(trayBox, brightnessBox);
315         QWidget::setTabOrder(brightnessBox, animScanButton);
316         QWidget::setTabOrder(animScanButton, fpsBox);
317         QWidget::setTabOrder(fpsBox, ditherBox);
318         QWidget::setTabOrder(ditherBox, delayBox);
319         QWidget::setTabOrder(delayBox, mAccelBox);
320         QWidget::setTabOrder(mAccelBox, sAccelBox);
321         QWidget::setTabOrder(sAccelBox, sSpeedBox);
322 
323         retranslateUi(ExtraSettingsWidget);
324         QObject::connect(buttonBox, SIGNAL(accepted()), ExtraSettingsWidget, SLOT(accept()));
325         QObject::connect(buttonBox, SIGNAL(rejected()), ExtraSettingsWidget, SLOT(reject()));
326 
327         QMetaObject::connectSlotsByName(ExtraSettingsWidget);
328     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.88.3 Field Documentation

12.88.3.1 QLabel* Ui_ExtraSettingsWidget::animCountLabel

Definition at line 57 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::on_animScanButton_clicked(), retranslateUi(), and setupUi().

12.88.3.2 QLabel* Ui_ExtraSettingsWidget::animPathLabel

Definition at line 56 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.3 QPushButton* Ui_ExtraSettingsWidget::animScanButton

Definition at line 58 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.4 QCheckBox* Ui_ExtraSettingsWidget::brightnessBox

Definition at line 59 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.5 QDialogButtonBox* Ui_ExtraSettingsWidget::buttonBox

Definition at line 76 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.6 QCheckBox* Ui_ExtraSettingsWidget::delayBox

Definition at line 67 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.7 QCheckBox* Ui_ExtraSettingsWidget::ditherBox

Definition at line 66 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.8 QSpinBox* Ui_ExtraSettingsWidget::fpsBox

Definition at line 42 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and setupUi().

12.88.3.9 QLabel* Ui_ExtraSettingsWidget::fpsWarnLabel

Definition at line 39 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_fpsBox_valueChanged(), retranslateUi(), and setupUi().

12.88.3.10 QGridLayout* Ui_ExtraSettingsWidget::gridLayout

Definition at line 34 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.11 QHBoxLayout* Ui_ExtraSettingsWidget::horizontalLayout

Definition at line 55 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.12 QHBoxLayout* Ui_ExtraSettingsWidget::horizontalLayout_2

Definition at line 72 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.13 QSpacerItem* Ui_ExtraSettingsWidget::horizontalSpacer

Definition at line 53 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.14 QSpacerItem* Ui_ExtraSettingsWidget::horizontalSpacer_2

Definition at line 35 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.15 QSpacerItem* Ui_ExtraSettingsWidget::horizontalSpacer_3

Definition at line 40 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.16 QLabel* Ui_ExtraSettingsWidget::label

Definition at line 61 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.17 QLabel* Ui_ExtraSettingsWidget::label_10

Definition at line 73 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.18 QLabel* Ui_ExtraSettingsWidget::label_11

Definition at line 75 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.19 QLabel* Ui_ExtraSettingsWidget::label_2

Definition at line 65 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.20 QLabel* Ui_ExtraSettingsWidget::label_3

Definition at line 41 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.21 QLabel* Ui_ExtraSettingsWidget::label_5

Definition at line 63 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.22 QLabel* Ui_ExtraSettingsWidget::label_6

Definition at line 45 of file ui_extrasettingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.88.3.23 QFrame* Ui_ExtraSettingsWidget::line

Definition at line 62 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.24 QFrame* Ui_ExtraSettingsWidget::line_2

Definition at line 54 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.25 QFrame* Ui_ExtraSettingsWidget::line_3

Definition at line 64 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.26 QCheckBox* Ui_ExtraSettingsWidget::mAccelBox

Definition at line 70 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.27 QLabel* Ui_ExtraSettingsWidget::osxLabel

Definition at line 68 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.28 QFrame* Ui_ExtraSettingsWidget::osxLine

Definition at line 69 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and setupUi().

12.88.3.29 QCheckBox* Ui_ExtraSettingsWidget::sAccelBox

Definition at line 48 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sSpeedBox_valueChanged(), retranslateUi(), and setupUi().

12.88.3.30 QSpinBox* Ui_ExtraSettingsWidget::sSpeedBox

Definition at line 74 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sAccelBox_clicked(), and setupUi().

12.88.3.31 QWidget* Ui_ExtraSettingsWidget::sSpeedWidget

Definition at line 71 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), ExtraSettingsWidget::on_sAccelBox_clicked(), and setupUi().

12.88.3.32 QCheckBox* Ui_ExtraSettingsWidget::trayBox

Definition at line 60 of file ui_extrasettingswidget.h.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), retranslateUi(), and setupUi().

12.88.3.33 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer

Definition at line 36 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.34 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_10

Definition at line 51 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.35 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_11

Definition at line 52 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.36 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_2

Definition at line 38 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.37 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_3

Definition at line 37 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.38 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_4

Definition at line 43 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.39 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_5

Definition at line 47 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.40 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_6

Definition at line 46 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.41 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_7

Definition at line 49 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.42 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_8

Definition at line 50 of file ui_extrasettingswidget.h.

Referenced by setupUi().

12.88.3.43 QSpacerItem* Ui_ExtraSettingsWidget::verticalSpacer_9

Definition at line 44 of file ui_extrasettingswidget.h.

Referenced by setupUi().

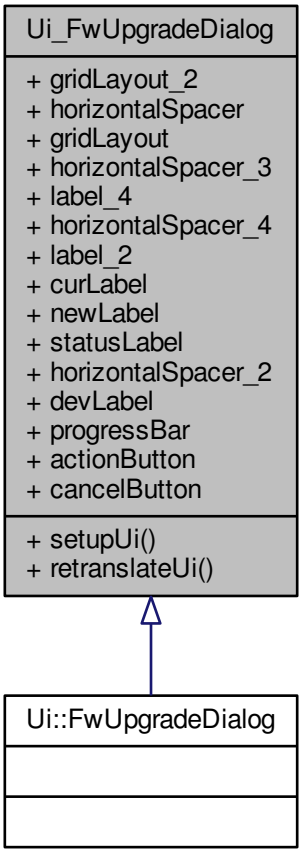
The documentation for this class was generated from the following file:

- src/ckb/ui_extrasettingswidget.h

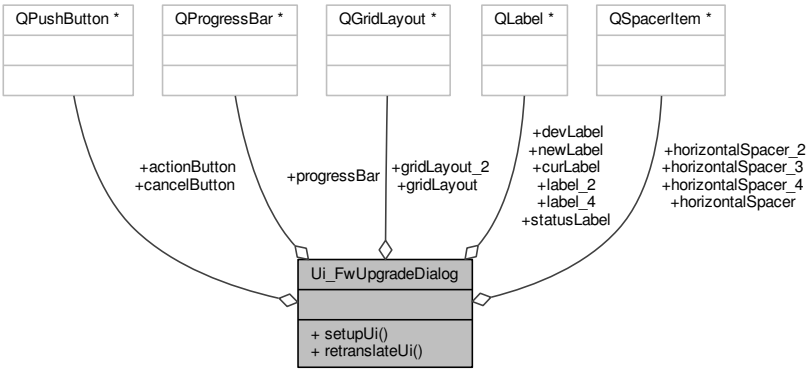
12.89 Ui_FwUpgradeDialog Class Reference

`#include <src/ckb/ui_fwupgradedialog.h>`

Inheritance diagram for Ui_FwUpgradeDialog:



Collaboration diagram for Ui_FwUpgradeDialog:



Public Member Functions

- void setupUi (QDialog *FwUpgradeDialog)
- void retranslateUi (QDialog *FwUpgradeDialog)

Data Fields

- QGridLayout * gridLayout_2
- QSpacerItem * horizontalSpacer
- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer_3
- QLabel * label_4
- QSpacerItem * horizontalSpacer_4
- QLabel * label_2
- QLabel * curLabel
- QLabel * newLabel
- QLabel * statusLabel
- QSpacerItem * horizontalSpacer_2
- QLabel * devLabel
- QProgressBar * progressBar
- QPushButton * actionButton
- QPushButton * cancelButton

12.89.1 Detailed Description

Definition at line 26 of file ui_fwupgradedialog.h.

12.89.2 Member Function Documentation

12.89.2.1 void Ui_FwUpgradeDialog::retranslateUi (QDialog * *FwUpgradeDialog*) [inline]

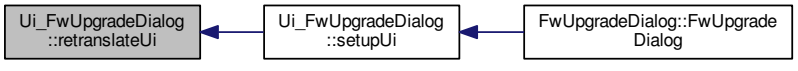
Definition at line 156 of file ui_fwupgradedialog.h.

References actionButton, cancelButton, curLabel, devLabel, label_2, label_4, newLabel, and statusLabel.

Referenced by setupUi().

```
157     {
158         FwUpgradeDialog->setWindowTitle(QApplication::translate("FwUpgradeDialog", "Firmware update",
    Q_NULLPTR));
159         label_4->setText(QApplication::translate("FwUpgradeDialog", "New:", Q_NULLPTR));
160         label_2->setText(QApplication::translate("FwUpgradeDialog", "Current:", Q_NULLPTR));
161         curLabel->setText(QApplication::translate("FwUpgradeDialog", "0.00", Q_NULLPTR));
162         newLabel->setText(QApplication::translate("FwUpgradeDialog", "0.00", Q_NULLPTR));
163         statusLabel->setText(QApplication::translate("FwUpgradeDialog", "Downloading new
     firmware, please wait...", Q_NULLPTR));
164         devLabel->setText(QApplication::translate("FwUpgradeDialog", "Device", Q_NULLPTR));
165         actionButton->setText(QApplication::translate("FwUpgradeDialog", "Start", Q_NULLPTR));
166         cancelButton->setText(QApplication::translate("FwUpgradeDialog", "Cancel", Q_NULLPTR));
167     } // retranslateUi
```

Here is the caller graph for this function:



12.89.2.2 void Ui_FwUpgradeDialog::setupUi (QDialog * *FwUpgradeDialog*) [inline]

Definition at line 45 of file ui_fwupgradedialog.h.

References actionButton, cancelButton, curLabel, devLabel, gridLayout, gridLayout_2, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, horizontalSpacer_4, label_2, label_4, newLabel, progressBar, retranslateUi(), and statusLabel.

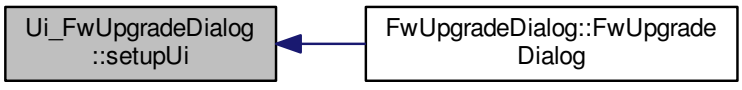
Referenced by FwUpgradeDialog::FwUpgradeDialog().

```
46     {
47         if (FwUpgradeDialog->objectName().isEmpty())
48             FwUpgradeDialog->setObjectName(QStringLiteral("FwUpgradeDialog"));
49         FwUpgradeDialog->resize(420, 280);
50         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
51         sizePolicy.setHorizontalStretch(0);
52         sizePolicy.setVerticalStretch(0);
53         sizePolicy.setHeightForWidth(FwUpgradeDialog->sizePolicy().hasHeightForWidth());
54         FwUpgradeDialog->setSizePolicy(sizePolicy);
55         gridLayout_2 = new QGridLayout(FwUpgradeDialog);
56         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
57         horizontalSpacer = new QSpacerItem(127, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
58
59         gridLayout_2->addItem(horizontalSpacer, 1, 0, 1, 1);
60
61         gridLayout = new QGridLayout();
62         gridLayout->setObjectName(QStringLiteral("gridLayout"));
63         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
64
65         gridLayout->addItem(horizontalSpacer_3, 1, 0, 1, 1);
66
67         label_4 = new QLabel(FwUpgradeDialog);
68         label_4->setObjectName(QStringLiteral("label_4"));
69         QFont font;
70         font.setBold(true);
71         font.setWeight(75);
72         label_4->setFont(font);
73
74         gridLayout->addWidget(label_4, 1, 4, 1, 1);
75
76         horizontalSpacer_4 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
77
78         gridLayout->addItem(horizontalSpacer_4, 1, 6, 1, 1);
79
80         label_2 = new QLabel(FwUpgradeDialog);
81         label_2->setObjectName(QStringLiteral("label_2"));
82         label_2->setFont(font);
83
84         gridLayout->addWidget(label_2, 1, 1, 1, 1);
85
86         curLabel = new QLabel(FwUpgradeDialog);
87         curLabel->setObjectName(QStringLiteral("curLabel"));
88
89         gridLayout->addWidget(curLabel, 1, 2, 1, 1);
90
91         newLabel = new QLabel(FwUpgradeDialog);
92         newLabel->setObjectName(QStringLiteral("newLabel"));
93
94         gridLayout->addWidget(newLabel, 1, 5, 1, 1);
95
96         statusLabel = new QLabel(FwUpgradeDialog);
97         statusLabel->setObjectName(QStringLiteral("statusLabel"));
98         QSizePolicy sizePolicy1(QSizePolicy::Preferred, QSizePolicy::Expanding);
99         sizePolicy1.setHorizontalStretch(0);
100         sizePolicy1.setVerticalStretch(0);
101         sizePolicy1.setHeightForWidth(statusLabel->sizePolicy().hasHeightForWidth());
102         statusLabel->setSizePolicy(sizePolicy1);
103         statusLabel->setAlignment(Qt::AlignCenter);
104         statusLabel->setWordWrap(true);
105
106         gridLayout->addWidget(statusLabel, 2, 0, 1, 7);
107
108         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
109
110         gridLayout->addItem(horizontalSpacer_2, 1, 3, 1, 1);
111
112         devLabel = new QLabel(FwUpgradeDialog);
113         devLabel->setObjectName(QStringLiteral("devLabel"));
114         QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Minimum);
115         sizePolicy2.setHorizontalStretch(0);
```

```
116         sizePolicy2.setVerticalStretch(0);
117         sizePolicy2.setHeightForWidth(devLabel->sizePolicy().hasHeightForWidth());
118         devLabel->setSizePolicy(sizePolicy2);
119         devLabel->setFont(font);
120         devLabel->setAlignment(Qt::AlignCenter);
121
122         gridLayout->addWidget(devLabel, 0, 0, 1, 7);
123
124         progressBar = new QProgressBar(FwUpgradeDialog);
125         progressBar->setObjectName(QStringLiteral("progressBar"));
126         progressBar->setMaximum(0);
127         progressBar->setValue(0);
128
129         gridLayout->addWidget(progressBar, 3, 0, 1, 7);
130
131
132         gridLayout_2->addLayout(gridLayout, 0, 0, 1, 3);
133
134         actionButton = new QPushButton(FwUpgradeDialog);
135         actionButton->setObjectName(QStringLiteral("actionButton"));
136         actionButton->setEnabled(false);
137
138         gridLayout_2->addWidget(actionButton, 1, 2, 1, 1);
139
140         cancelButton = new QPushButton(FwUpgradeDialog);
141         cancelButton->setObjectName(QStringLiteral("cancelButton"));
142         cancelButton->setEnabled(false);
143
144         gridLayout_2->addWidget(cancelButton, 1, 1, 1, 1);
145
146         QWidget::setTabOrder(actionButton, cancelButton);
147
148         retranslateUi(FwUpgradeDialog);
149
150         actionButton->setDefault(true);
151
152
153         QMetaObject::connectSlotsByName(FwUpgradeDialog);
154     } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.89.3 Field Documentation

12.89.3.1 QPushButton* Ui_FwUpgradeDialog::actionButton

Definition at line 42 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::fwUpdateFinished(), FwUpgradeDialog::on_actionButton_clicked(), retranslateUi(), and setupUi().

12.89.3.2 QPushButton* Ui_FwUpgradeDialog::cancelButton

Definition at line 43 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::on_actionButton_clicked(), retranslateUi(), and setupUi().

12.89.3.3 QLabel* Ui_FwUpgradeDialog::curLabel

Definition at line 36 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), retranslateUi(), and setupUi().

12.89.3.4 QLabel* Ui_FwUpgradeDialog::devLabel

Definition at line 40 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), retranslateUi(), and setupUi().

12.89.3.5 QGridLayout* Ui_FwUpgradeDialog::gridLayout

Definition at line 31 of file ui_fwupgradedialog.h.

Referenced by setupUi().

12.89.3.6 QGridLayout* Ui_FwUpgradeDialog::gridLayout_2

Definition at line 29 of file ui_fwupgradedialog.h.

Referenced by setupUi().

12.89.3.7 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer

Definition at line 30 of file ui_fwupgradedialog.h.

Referenced by setupUi().

12.89.3.8 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer_2

Definition at line 39 of file ui_fwupgradedialog.h.

Referenced by setupUi().

12.89.3.9 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer_3

Definition at line 32 of file ui_fwupgradedialog.h.

Referenced by setupUi().

12.89.3.10 QSpacerItem* Ui_FwUpgradeDialog::horizontalSpacer_4

Definition at line 34 of file ui_fwupgradedialog.h.

Referenced by setupUi().

12.89.3.11 QLabel* Ui_FwUpgradeDialog::label_2

Definition at line 35 of file ui_fwupgradedialog.h.

Referenced by retranslateUi(), and setupUi().

12.89.3.12 QLabel* Ui_FwUpgradeDialog::label_4

Definition at line 33 of file ui_fwupgradedialog.h.

Referenced by retranslateUi(), and setupUi().

12.89.3.13 QLabel* Ui_FwUpgradeDialog::newLabel

Definition at line 37 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::FwUpgradeDialog(), retranslateUi(), and setupUi().

12.89.3.14 QProgressBar* Ui_FwUpgradeDialog::progressBar

Definition at line 41 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::fwUpdateFinished(), FwUpgradeDialog::fwUpdateProgress(), FwUpgradeDialog::on_actionButton_clicked(), and setupUi().

12.89.3.15 QLabel* Ui_FwUpgradeDialog::statusLabel

Definition at line 38 of file ui_fwupgradedialog.h.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::fwUpdateFinished(), FwUpgradeDialog::on_actionButton_clicked(), retranslateUi(), and setupUi().

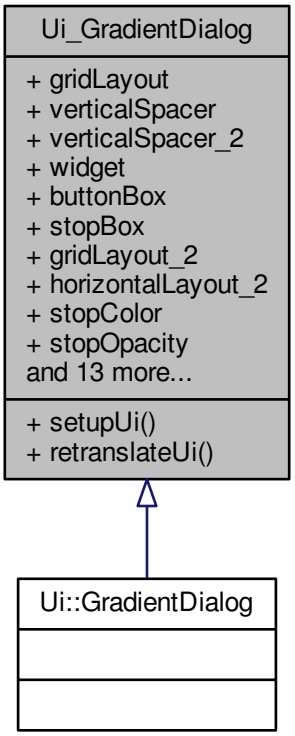
The documentation for this class was generated from the following file:

- src/ckb/ui_fwupgradedialog.h

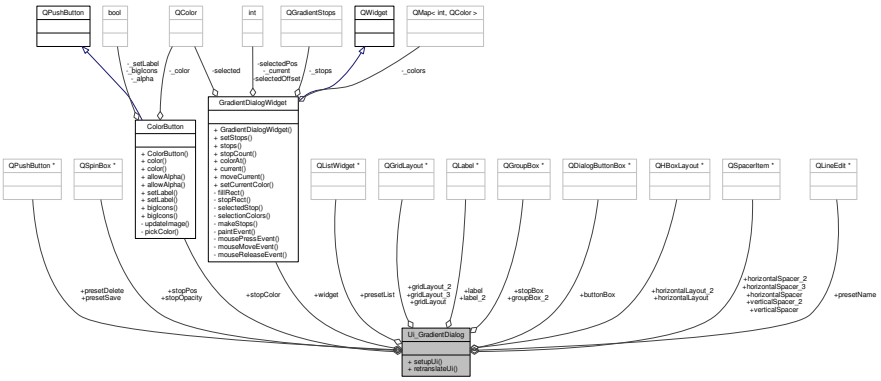
12.90 Ui_GradientDialog Class Reference

`#include <src/ckb/ui_gradientdialog.h>`

Inheritance diagram for Ui_GradientDialog:



Collaboration diagram for Ui_GradientDialog:



Public Member Functions

- void setupUi (QDialog *GradientDialog)
- void retranslateUi (QDialog *GradientDialog)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_2
- GradientDialogWidget * widget
- QDialogButtonBox * buttonBox
- QGroupBox * stopBox
- QGridLayout * gridLayout_2
- QHBoxLayout * horizontalLayout_2
- ColorButton * stopColor
- QSpinBox * stopOpacity
- QSpacerItem * horizontalSpacer_2
- QLabel * label
- QHBoxLayout * horizontalLayout
- QSpinBox * stopPos
- QSpacerItem * horizontalSpacer
- QLabel * label_2
- QSpacerItem * horizontalSpacer_3
- QGroupBox * groupBox_2
- QGridLayout * gridLayout_3
- QListWidget * presetList
- QLineEdit * presetName
- QPushButton * presetSave
- QPushButton * presetDelete

12.90.1 Detailed Description

Definition at line 33 of file ui_gradientdialog.h.

12.90.2 Member Function Documentation

12.90.2.1 void Ui_GradientDialog::retranslateUi (QDialog * *GradientDialog*) [inline]

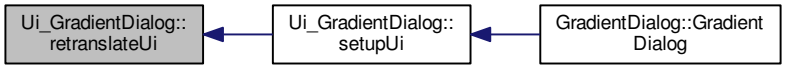
Definition at line 213 of file ui_gradientdialog.h.

References groupBox_2, label, label_2, presetDelete, presetName, presetSave, stopBox, stopColor, stopOpacity, and stopPos.

Referenced by setupUi().

```
214     {
215         GradientDialog->setWindowTitle(QApplication::translate("GradientDialog", "Pick Gradient", Q_NULLPTR
    ));
216         stopBox->setTitle(QApplication::translate("GradientDialog", "Point X of Y", Q_NULLPTR));
217         stopColor->setText(QApplication::translate("GradientDialog", "Pick Color...", Q_NULLPTR));
218         stopOpacity->setSuffix(QApplication::translate("GradientDialog", "%", Q_NULLPTR));
219         label->setText(QApplication::translate("GradientDialog", "Position:", Q_NULLPTR));
220         stopPos->setSuffix(QApplication::translate("GradientDialog", "%", Q_NULLPTR));
221         label_2->setText(QApplication::translate("GradientDialog", "Color:", Q_NULLPTR));
222         groupBox_2->setTitle(QApplication::translate("GradientDialog", "Presets", Q_NULLPTR));
223         presetName->setPlaceholderText(QApplication::translate("GradientDialog", "Name",
    Q_NULLPTR));
224         presetSave->setText(QApplication::translate("GradientDialog", "Save", Q_NULLPTR));
225         presetDelete->setText(QApplication::translate("GradientDialog", "Delete", Q_NULLPTR));
226     } // retranslateUi
```

Here is the caller graph for this function:



12.90.2.2 void Ui_GradientDialog::setupUi (QDialog * *GradientDialog*) [inline]

Definition at line 60 of file ui_gradientdialog.h.

References buttonBox, gridLayout, gridLayout_2, gridLayout_3, groupBox_2, horizontalLayout, horizontalLayout_2, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, label, label_2, presetDelete, presetList, presetName, presetSave, retranslateUi(), stopBox, stopColor, stopOpacity, stopPos, verticalSpacer, verticalSpacer_2, and widget.

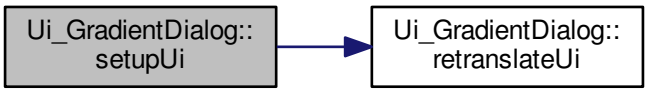
Referenced by GradientDialog::GradientDialog().

```
61     {
62         if (GradientDialog->objectName().isEmpty())
63             GradientDialog->setObjectName(QStringLiteral("GradientDialog"));
64         GradientDialog->resize(400, 440);
65         GradientDialog->setMinimumSize(QSize(400, 440));
66         GradientDialog->setMaximumSize(QSize(400, 440));
67         GradientDialog->setFocusPolicy(Qt::ClickFocus);
68         gridLayout = new QGridLayout(GradientDialog);
69         gridLayout->setObjectName(QStringLiteral("gridLayout"));
70         verticalSpacer = new QSpacerItem(0, 0, QSizePolicy::Minimum, QSizePolicy::Expanding);
71 
72         gridLayout->addItem(verticalSpacer, 1, 0, 1, 1);
73 
74         verticalSpacer_2 = new QSpacerItem(0, 15, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
75 
76         gridLayout->addItem(verticalSpacer_2, 3, 0, 1, 1);
77 
78         widget = new GradientDialogWidget(GradientDialog);
79         widget->setObjectName(QStringLiteral("widget"));
80         QSizePolicy sizePolicy(QSizePolicy::Preferred, QSizePolicy::Fixed);
81         sizePolicy.setHorizontalStretch(0);
82         sizePolicy.setVerticalStretch(0);
83         sizePolicy.setHeightForWidth(widget->sizePolicy().hasHeightForWidth());
84         widget->setSizePolicy(sizePolicy);
85         widget->setMinimumSize(QSize(0, 74));
86         widget->setMaximumSize(QSize(16777215, 74));
87 
88         gridLayout->addWidget(widget, 2, 0, 1, 1);
89 
90         buttonBox = new QDialogButtonBox(GradientDialog);
91         buttonBox->setObjectName(QStringLiteral("buttonBox"));
92         buttonBox->setOrientation(Qt::Horizontal);
93         buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
94 
95         gridLayout->addWidget(buttonBox, 5, 0, 1, 1);
96 
97         stopBox = new QGroupBox(GradientDialog);
98         stopBox->setObjectName(QStringLiteral("stopBox"));
99         sizePolicy.setHeightForWidth(stopBox->sizePolicy().hasHeightForWidth());
100         stopBox->setSizePolicy(sizePolicy);
101         gridLayout_2 = new QGridLayout(stopBox);
102         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
103         horizontalLayout_2 = new QHBoxLayout();
104         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
105         stopColor = new ColorButton(stopBox);
106         stopColor->setObjectName(QStringLiteral("stopColor"));
107 
108         horizontalLayout_2->addWidget(stopColor);
109 
110         stopOpacity = new QSpinBox(stopBox);
111         stopOpacity->setObjectName(QStringLiteral("stopOpacity"));
112         stopOpacity->setMaximum(100);
113 
```

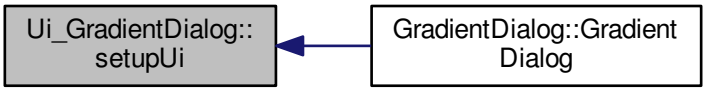
```
114         horizontalLayout_2->addWidget(stopOpacity);
115 
116         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
117 
118         horizontalLayout_2->addItem(horizontalSpacer_2);
119 
120 
121         gridLayout_2->addLayout(horizontalLayout_2, 1, 2, 1, 1);
122 
123         label = new QLabel(stopBox);
124         label->setObjectName(QStringLiteral("label"));
125 
126         gridLayout_2->addWidget(label, 0, 0, 1, 1);
127 
128         horizontalLayout = new QHBoxLayout();
129         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
130         stopPos = new QSpinBox(stopBox);
131         stopPos->setObjectName(QStringLiteral("stopPos"));
132         stopPos->setMaximum(100);
133 
134         horizontalLayout->addWidget(stopPos);
135 
136         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
137 
138         horizontalLayout->addItem(horizontalSpacer);
139 
140 
141         gridLayout_2->addLayout(horizontalLayout, 0, 2, 1, 1);
142 
143         label_2 = new QLabel(stopBox);
144         label_2->setObjectName(QStringLiteral("label_2"));
145 
146         gridLayout_2->addWidget(label_2, 1, 0, 1, 1);
147 
148         horizontalSpacer_3 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
149 
150         gridLayout_2->addItem(horizontalSpacer_3, 0, 1, 1, 1);
151 
152 
153         gridLayout->addWidget(stopBox, 4, 0, 1, 1);
154 
155         groupBox_2 = new QGroupBox(GradientDialog);
156         groupBox_2->setObjectName(QStringLiteral("groupBox_2"));
157         sizePolicy.setHeightForWidth(groupBox_2->sizePolicy().hasHeightForWidth());
158         groupBox_2->setSizePolicy(sizePolicy);
159         gridLayout_3 = new QGridLayout(groupBox_2);
160         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
161         presetList = new QListWidget(groupBox_2);
162         presetList->setObjectName(QStringLiteral("presetList"));
163         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Preferred);
164         sizePolicy1.setHorizontalStretch(0);
165         sizePolicy1.setVerticalStretch(0);
166         sizePolicy1.setHeightForWidth(presetList->sizePolicy().hasHeightForWidth());
167         presetList->setSizePolicy(sizePolicy1);
168         presetList->setFrameShadow(QFrame::Plain);
169         presetList->setHorizontalScrollBarPolicy(Qt::ScrollBarAlwaysOn);
170         presetList->setIconSize(QSize(64, 64));
171         presetList->setHorizontalScrollMode(QAbstractItemView::ScrollPerPixel);
172         presetList->setFlow(QListView::TopToBottom);
173         presetList->setViewMode(QListView::IconMode);
174         presetList->setUniformItemSizes(true);
175 
176         gridLayout_3->addWidget(presetList, 0, 0, 3, 1);
177 
178         presetName = new QLineEdit(groupBox_2);
179         presetName->setObjectName(QStringLiteral("presetName"));
180         presetName->setMaxLength(30);
181 
182         gridLayout_3->addWidget(presetName, 0, 1, 1, 1);
183 
184         presetSave = new QPushButton(groupBox_2);
185         presetSave->setObjectName(QStringLiteral("presetSave"));
186         presetSave->setAutoDefault(false);
187 
188         gridLayout_3->addWidget(presetSave, 1, 1, 1, 1);
189 
190         presetDelete = new QPushButton(groupBox_2);
191         presetDelete->setObjectName(QStringLiteral("presetDelete"));
192         presetDelete->setAutoDefault(false);
193 
194         gridLayout_3->addWidget(presetDelete, 2, 1, 1, 1);
195 
196 
197         gridLayout->addWidget(groupBox_2, 0, 0, 1, 1);
```

```
198
199         QWidget::setTabOrder(presetList, presetName);
200         QWidget::setTabOrder(presetName, presetSave);
201         QWidget::setTabOrder(presetSave, presetDelete);
202         QWidget::setTabOrder(presetDelete, stopPos);
203         QWidget::setTabOrder(stopPos, stopColor);
204         QWidget::setTabOrder(stopColor, stopOpacity);
205
206         retranslateUi(GradientDialog);
207         QObject::connect(buttonBox, SIGNAL(accepted()), GradientDialog, SLOT(accept()));
208         QObject::connect(buttonBox, SIGNAL(rejected()), GradientDialog, SLOT(reject()));
209
210         QMetaObject::connectSlotsByName(GradientDialog);
211     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.90.3 Field Documentation

12.90.3.1 QDialogButtonBox* Ui_GradientDialog::buttonBox

Definition at line 40 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.2 QGridLayout* Ui_GradientDialog::gridLayout

Definition at line 36 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.3 QGridLayout* Ui_GradientDialog::gridLayout_2

Definition at line 42 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.4 QGridLayout* Ui_GradientDialog::gridLayout_3

Definition at line 54 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.5 QGroupBox* Ui_GradientDialog::groupBox_2

Definition at line 53 of file ui_gradientdialog.h.

Referenced by retranslateUi(), and setupUi().

12.90.3.6 QHBoxLayout* Ui_GradientDialog::horizontalLayout

Definition at line 48 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.7 QHBoxLayout* Ui_GradientDialog::horizontalLayout_2

Definition at line 43 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.8 QSpacerItem* Ui_GradientDialog::horizontalSpacer

Definition at line 50 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.9 QSpacerItem* Ui_GradientDialog::horizontalSpacer_2

Definition at line 46 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.10 QSpacerItem* Ui_GradientDialog::horizontalSpacer_3

Definition at line 52 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.11 QLabel* Ui_GradientDialog::label

Definition at line 47 of file ui_gradientdialog.h.

Referenced by retranslateUi(), and setupUi().

12.90.3.12 QLabel* Ui_GradientDialog::label_2

Definition at line 51 of file ui_gradientdialog.h.

Referenced by retranslateUi(), and setupUi().

12.90.3.13 QPushButton* Ui_GradientDialog::presetDelete

Definition at line 58 of file ui_gradientdialog.h.

Referenced by GradientDialog::getGradient(), GradientDialog::on_presetDelete_clicked(), GradientDialog::on_presetSave_clicked(), retranslateUi(), GradientDialog::setPreset(), and setupUi().

12.90.3.14 QListWidget* Ui_GradientDialog::presetList

Definition at line 55 of file ui_gradientdialog.h.

Referenced by GradientDialog::setPreset(), setupUi(), and GradientDialog::updatePresets().

12.90.3.15 QLineEdit* Ui_GradientDialog::presetName

Definition at line 56 of file ui_gradientdialog.h.

Referenced by GradientDialog::on_presetSave_clicked(), retranslateUi(), GradientDialog::setPreset(), and setupUi().

12.90.3.16 QPushButton* Ui_GradientDialog::presetSave

Definition at line 57 of file ui_gradientdialog.h.

Referenced by GradientDialog::getGradient(), GradientDialog::on_presetDelete_clicked(), GradientDialog::on_presetSave_clicked(), retranslateUi(), GradientDialog::setPreset(), and setupUi().

12.90.3.17 QGroupBox* Ui_GradientDialog::stopBox

Definition at line 41 of file ui_gradientdialog.h.

Referenced by GradientDialog::currentChanged(), retranslateUi(), and setupUi().

12.90.3.18 ColorButton* Ui_GradientDialog::stopColor

Definition at line 44 of file ui_gradientdialog.h.

Referenced by GradientDialog::currentChanged(), GradientDialog::GradientDialog(), GradientDialog::on_stopOpacity_valueChanged(), retranslateUi(), and setupUi().

12.90.3.19 QSpinBox* Ui_GradientDialog::stopOpacity

Definition at line 45 of file ui_gradientdialog.h.

Referenced by GradientDialog::colorChanged(), GradientDialog::currentChanged(), retranslateUi(), and setupUi().

12.90.3.20 QSpinBox* Ui_GradientDialog::stopPos

Definition at line 49 of file ui_gradientdialog.h.

Referenced by GradientDialog::currentChanged(), GradientDialog::on_stopPos_valueChanged(), retranslateUi(), and setupUi().

12.90.3.21 QSpacerItem* Ui_GradientDialog::verticalSpacer

Definition at line 37 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.22 QSpacerItem* Ui_GradientDialog::verticalSpacer_2

Definition at line 38 of file ui_gradientdialog.h.

Referenced by setupUi().

12.90.3.23 GradientDialogWidget* Ui_GradientDialog::widget

Definition at line 39 of file ui_gradientdialog.h.

Referenced by GradientDialog::colorChanged(), GradientDialog::currentChanged(), GradientDialog::getGradient(), GradientDialog::GradientDialog(), GradientDialog::on_presetSave_clicked(), GradientDialog::on_stopPos_valueChanged(), GradientDialog::setPreset(), and setupUi().

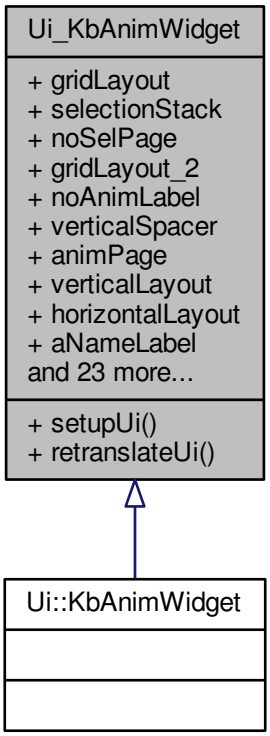
The documentation for this class was generated from the following file:

- src/ckb/ui_gradientdialog.h

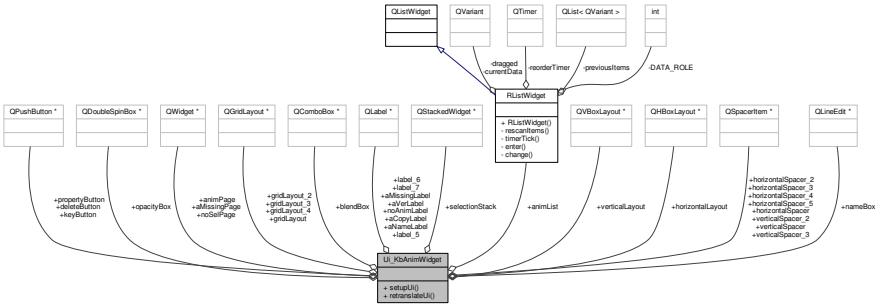
12.91 Ui_KbAnimWidget Class Reference

`#include <src/ckb/ui_kbanimwidget.h>`

Inheritance diagram for Ui_KbAnimWidget:



Collaboration diagram for Ui_KbAnimWidget:



Public Member Functions

- void setupUi (QWidget *KbAnimWidget)
- void retranslateUi (QWidget *KbAnimWidget)

Data Fields

- QGridLayout * gridLayout
- QStackedWidget * selectionStack
- QWidget * noSelPage
- QGridLayout * gridLayout_2
- QLabel * noAnimLabel
- QSpacerItem * verticalSpacer
- QWidget * animPage
- QVBoxLayout * verticalLayout
- QHBoxLayout * horizontalLayout
- QLabel * aNameLabel
- QLabel * aVerLabel
- QSpacerItem * horizontalSpacer
- QLabel * aCopyLabel
- QSpacerItem * verticalSpacer_3
- QGridLayout * gridLayout_3
- QLineEdit * nameBox
- QDoubleSpinBox * opacityBox
- QComboBox * blendBox
- QLabel * label_6
- QSpacerItem * horizontalSpacer_4
- QSpacerItem * horizontalSpacer_2
- QLabel * label_5
- QSpacerItem * horizontalSpacer_3
- QLabel * label_7
- QPushButton * keyButton
- QPushButton * deleteButton
- QPushButton * propertyButton
- QSpacerItem * horizontalSpacer_5
- QSpacerItem * verticalSpacer_2
- QWidget * aMissingPage
- QGridLayout * gridLayout_4
- QLabel * aMissingLabel
- RListWidget * animList

12.91.1 Detailed Description

Definition at line 32 of file ui_kbanimwidget.h.

12.91.2 Member Function Documentation

12.91.2.1 void Ui_KbAnimWidget::retranslateUi (QWidget * *KbAnimWidget*) [inline]

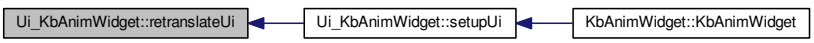
Definition at line 241 of file ui_kbanimwidget.h.

References aCopyLabel, aMissingLabel, aNameLabel, aVerLabel, blendBox, deleteButton, keyButton, label_5, label_6, label_7, noAnimLabel, opacityBox, and propertyButton.

Referenced by setupUi().

```
242     {
243         KbAnimWidget->setWindowTitle(QApplication::translate("KbAnimWidget", "Form", Q_NULLPTR));
244         noAnimLabel->setText(QApplication::translate("KbAnimWidget", "No animations in this mode
    ", Q_NULLPTR));
245         aNameLabel->setText(QApplication::translate("KbAnimWidget", "Animation name", Q_NULLPTR))
    ;
246         aVerLabel->setText(QApplication::translate("KbAnimWidget", "v0.0", Q_NULLPTR));
247         aCopyLabel->setText(QApplication::translate("KbAnimWidget", "Copyright \302\251",
    Q_NULLPTR));
248         opacityBox->setSuffix(QApplication::translate("KbAnimWidget", "%", Q_NULLPTR));
249         blendBox->clear();
250         blendBox->insertItems(0, QStringList()
251          << QApplication::translate("KbAnimWidget", "Normal", Q_NULLPTR)
252          << QApplication::translate("KbAnimWidget", "Add", Q_NULLPTR)
253          << QApplication::translate("KbAnimWidget", "Subtract", Q_NULLPTR)
254          << QApplication::translate("KbAnimWidget", "Multiply", Q_NULLPTR)
255          << QApplication::translate("KbAnimWidget", "Divide", Q_NULLPTR)
256         );
257         label_6->setText(QApplication::translate("KbAnimWidget", "Blend mode:", Q_NULLPTR));
258         label_5->setText(QApplication::translate("KbAnimWidget", "Opacity:", Q_NULLPTR));
259         label_7->setText(QApplication::translate("KbAnimWidget", "Name:", Q_NULLPTR));
260         keyButton->setText(QApplication::translate("KbAnimWidget", "Set keys", Q_NULLPTR));
261         deleteButton->setText(QApplication::translate("KbAnimWidget", "Delete animation",
    Q_NULLPTR));
262         propertyButton->setText(QApplication::translate("KbAnimWidget", "Edit properties",
    Q_NULLPTR));
263         aMissingLabel->setText(QApplication::translate("KbAnimWidget", "Mising animation",
    Q_NULLPTR));
264     } // retranslateUi
```

Here is the caller graph for this function:



12.91.2.2 void Ui_KbAnimWidget::setupUi (QWidget * *KbAnimWidget*) [inline]

Definition at line 69 of file ui_kbanimwidget.h.

References aCopyLabel, aMissingLabel, aMissingPage, aNameLabel, animList, animPage, aVerLabel, blendBox, deleteButton, gridLayout, gridLayout_2, gridLayout_3, gridLayout_4, horizontalLayout, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, horizontalSpacer_4, horizontalSpacer_5, keyButton, label_5, label_6, label_7, nameBox, noAnimLabel, noSelPage, opacityBox, propertyButton, retranslateUi(), selectionStack, verticalLayout, verticalSpacer, verticalSpacer_2, and verticalSpacer_3.

Referenced by KbAnimWidget::KbAnimWidget().

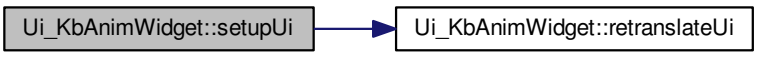
```
70     {
```

```
71          if (KbAnimWidget->objectName().isEmpty())
72              KbAnimWidget->setObjectName(QStringLiteral("KbAnimWidget"));
73          KbAnimWidget->resize(613, 300);
74          gridLayout = new QGridLayout(KbAnimWidget);
75          gridLayout->setObjectName(QStringLiteral("gridLayout"));
76          gridLayout->setContentsMargins(0, 0, 0, 0);
77          selectionStack = new QStackedWidget(KbAnimWidget);
78          selectionStack->setObjectName(QStringLiteral("selectionStack"));
79          noSelPage = new QWidget();
80          noSelPage->setObjectName(QStringLiteral("noSelPage"));
81          gridLayout_2 = new QGridLayout(noSelPage);
82          gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
83          gridLayout_2->setContentsMargins(6, 6, 6, 6);
84          noAnimLabel = new QLabel(noSelPage);
85          noAnimLabel->setObjectName(QStringLiteral("noAnimLabel"));
86
87          gridLayout_2->addWidget(noAnimLabel, 0, 0, 1, 1);
88
89          verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
90
91          gridLayout_2->addItem(verticalSpacer, 1, 0, 1, 1);
92
93          selectionStack->addWidget(noSelPage);
94          animPage = new QWidget();
95          animPage->setObjectName(QStringLiteral("animPage"));
96          verticalLayout = new QVBoxLayout(animPage);
97          verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
98          verticalLayout->setContentsMargins(6, 0, 0, 0);
99          horizontalLayout = new QHBoxLayout();
100         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
101         aNameLabel = new QLabel(animPage);
102         aNameLabel->setObjectName(QStringLiteral("aNameLabel"));
103         QFont font;
104         font.setBold(true);
105         font.setWeight(75);
106         aNameLabel->setFont(font);
107
108         horizontalLayout->addWidget(aNameLabel);
109
110         aVerLabel = new QLabel(animPage);
111         aVerLabel->setObjectName(QStringLiteral("aVerLabel"));
112
113         horizontalLayout->addWidget(aVerLabel);
114
115         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
116
117         horizontalLayout->addItem(horizontalSpacer);
118
119         aCopyLabel = new QLabel(animPage);
120         aCopyLabel->setObjectName(QStringLiteral("aCopyLabel"));
121
122         horizontalLayout->addWidget(aCopyLabel);
123
124
125         verticalLayout->addLayout(horizontalLayout);
126
127         verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
128
129         verticalLayout->addItem(verticalSpacer_3);
130
131         gridLayout_3 = new QGridLayout();
132         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
133         nameBox = new QLineEdit(animPage);
134         nameBox->setObjectName(QStringLiteral("nameBox"));
135         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
136         sizePolicy.setHorizontalStretch(0);
137         sizePolicy.setVerticalStretch(0);
138         sizePolicy.setHeightForWidth(nameBox->sizePolicy().hasHeightForWidth());
139         nameBox->setSizePolicy(sizePolicy);
140         nameBox->setMinimumSize(QSize(150, 0));
141         nameBox->setMaximumSize(QSize(150, 16777215));
142
143         gridLayout_3->addWidget(nameBox, 0, 3, 1, 2);
144
145         opacityBox = new QDoubleSpinBox(animPage);
146         opacityBox->setObjectName(QStringLiteral("opacityBox"));
147         opacityBox->setDecimals(1);
148         opacityBox->setValue(100);
149
150         gridLayout_3->addWidget(opacityBox, 1, 3, 1, 1);
151
152         blendBox = new QComboBox(animPage);
153         blendBox->setObjectName(QStringLiteral("blendBox"));
154
```

```
155         gridLayout_3->addWidget(blendBox, 2, 3, 1, 2);
156 
157         label_6 = new QLabel(animPage);
158         label_6->setObjectName(QStringLiteral("label_6"));
159 
160         gridLayout_3->addWidget(label_6, 2, 1, 1, 1);
161 
162         horizontalSpacer_4 = new QSpacerItem(0, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
163 
164         gridLayout_3->addItem(horizontalSpacer_4, 0, 0, 1, 1);
165 
166         horizontalSpacer_2 = new QSpacerItem(30, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
167 
168         gridLayout_3->addItem(horizontalSpacer_2, 0, 2, 1, 1);
169 
170         label_5 = new QLabel(animPage);
171         label_5->setObjectName(QStringLiteral("label_5"));
172 
173         gridLayout_3->addWidget(label_5, 1, 1, 1, 1);
174 
175         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
176 
177         gridLayout_3->addItem(horizontalSpacer_3, 0, 5, 1, 1);
178 
179         label_7 = new QLabel(animPage);
180         label_7->setObjectName(QStringLiteral("label_7"));
181 
182         gridLayout_3->addWidget(label_7, 0, 1, 1, 1);
183 
184         keyButton = new QPushButton(animPage);
185         keyButton->setObjectName(QStringLiteral("keyButton"));
186 
187         gridLayout_3->addWidget(keyButton, 1, 6, 1, 1);
188 
189         deleteButton = new QPushButton(animPage);
190         deleteButton->setObjectName(QStringLiteral("deleteButton"));
191 
192         gridLayout_3->addWidget(deleteButton, 2, 6, 1, 1);
193 
194         propertyButton = new QPushButton(animPage);
195         propertyButton->setObjectName(QStringLiteral("propertyButton"));
196 
197         gridLayout_3->addWidget(propertyButton, 0, 6, 1, 1);
198 
199         horizontalSpacer_5 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
200 
201         gridLayout_3->addItem(horizontalSpacer_5, 1, 4, 1, 1);
202 
203 
204         verticalLayout->addLayout(gridLayout_3);
205 
206         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
207 
208         verticalLayout->addItem(verticalSpacer_2);
209 
210         selectionStack->addWidget(animPage);
211         aMissingPage = new QWidget();
212         aMissingPage->setObjectName(QStringLiteral("aMissingPage"));
213         gridLayout_4 = new QGridLayout(aMissingPage);
214         gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
215         aMissingLabel = new QLabel(aMissingPage);
216         aMissingLabel->setObjectName(QStringLiteral("aMissingLabel"));
217         aMissingLabel->setFont(font);
218 
219         gridLayout_4->addWidget(aMissingLabel, 0, 0, 1, 1);
220 
221         selectionStack->addWidget(aMissingPage);
222 
223         gridLayout->addWidget(selectionStack, 0, 2, 2, 1);
224 
225         animList = new RListWidget(KbAnimWidget);
226         animList->setObjectName(QStringLiteral("animList"));
227         animList->setMaximumSize(QSize(130, 16777215));
228         animList->setContextMenuPolicy(Qt::CustomContextMenu);
229         animList->setFrameShadow(QFrame::Plain);
230         animList->setDragDropMode(QAbstractItemView::InternalMove);
231         animList->setMovement(QListView::Snap);
232 
233         gridLayout->addWidget(animList, 0, 0, 2, 1);
234 
235 
236         retranslateUi(KbAnimWidget);
```

```
237
238         QMetaObject::connectSlotsByName(KbAnimWidget);
239     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.91.3 Field Documentation

12.91.3.1 QLabel* Ui_KbAnimWidget::aCopyLabel

Definition at line 47 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.2 QLabel* Ui_KbAnimWidget::aMissingLabel

Definition at line 66 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.3 QWidget* Ui_KbAnimWidget::aMissingPage

Definition at line 64 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.4 QLabel* Ui_KbAnimWidget::aNameLabel

Definition at line 44 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.5 RListWidget* Ui_KbAnimWidget::animList

Definition at line 67 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::addAnim(), KbAnimWidget::clearSelection(), KbAnimWidget::duplicateAnim(), KbAnimWidget::KbAnimWidget(), KbAnimWidget::on_animList_customContextMenuRequested(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::on_nameBox_textEdited(), KbAnimWidget::refreshList(), KbAnimWidget::reorderAnims(), and setupUi().

12.91.3.6 QWidget* Ui_KbAnimWidget::animPage

Definition at line 41 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.7 QLabel* Ui_KbAnimWidget::aVerLabel

Definition at line 45 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.8 QComboBox* Ui_KbAnimWidget::blendBox

Definition at line 52 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.9 QPushButton* Ui_KbAnimWidget::deleteButton

Definition at line 60 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), and setupUi().

12.91.3.10 QGridLayout* Ui_KbAnimWidget::gridLayout

Definition at line 35 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.11 QGridLayout* Ui_KbAnimWidget::gridLayout_2

Definition at line 38 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.12 QGridLayout* Ui_KbAnimWidget::gridLayout_3

Definition at line 49 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.13 QGridLayout* Ui_KbAnimWidget::gridLayout_4

Definition at line 65 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.14 QHBoxLayout* Ui_KbAnimWidget::horizontalLayout

Definition at line 43 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.15 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer

Definition at line 46 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.16 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_2

Definition at line 55 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.17 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_3

Definition at line 57 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.18 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_4

Definition at line 54 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.19 QSpacerItem* Ui_KbAnimWidget::horizontalSpacer_5

Definition at line 62 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.20 QPushButton* Ui_KbAnimWidget::keyButton

Definition at line 59 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setSelectedKeys(), and setupUi().

12.91.3.21 QLabel* Ui_KbAnimWidget::label_5

Definition at line 56 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), and setupUi().

12.91.3.22 QLabel* Ui_KbAnimWidget::label_6

Definition at line 53 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), and setupUi().

12.91.3.23 QLabel* Ui_KbAnimWidget::label_7

Definition at line 58 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), and setupUi().

12.91.3.24 QLineEdit* Ui_KbAnimWidget::nameBox

Definition at line 50 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::on_animList_itemChanged(), KbAnimWidget::on_propertyButton_clicked(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.25 QLabel* Ui_KbAnimWidget::noAnimLabel

Definition at line 39 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::addAnim(), KbAnimWidget::duplicateAnim(), KbAnimWidget::on_deleteButton_clicked(), KbAnimWidget::refreshList(), retranslateUi(), and setupUi().

12.91.3.26 QWidget* Ui_KbAnimWidget::noSelPage

Definition at line 37 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.27 QDoubleSpinBox* Ui_KbAnimWidget::opacityBox

Definition at line 51 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), KbAnimWidget::setCurrent(), and setupUi().

12.91.3.28 QPushButton* Ui_KbAnimWidget::propertyButton

Definition at line 61 of file ui_kbanimwidget.h.

Referenced by retranslateUi(), and setupUi().

12.91.3.29 QStackedWidget* Ui_KbAnimWidget::selectionStack

Definition at line 36 of file ui_kbanimwidget.h.

Referenced by KbAnimWidget::setCurrent(), and setupUi().

12.91.3.30 QVBoxLayout* Ui_KbAnimWidget::verticalLayout

Definition at line 42 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.31 QSpacerItem* Ui_KbAnimWidget::verticalSpacer

Definition at line 40 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.32 QSpacerItem* Ui_KbAnimWidget::verticalSpacer_2

Definition at line 63 of file ui_kbanimwidget.h.

Referenced by setupUi().

12.91.3.33 QSpacerItem* Ui_KbAnimWidget::verticalSpacer_3

Definition at line 48 of file ui_kbanimwidget.h.

Referenced by setupUi().

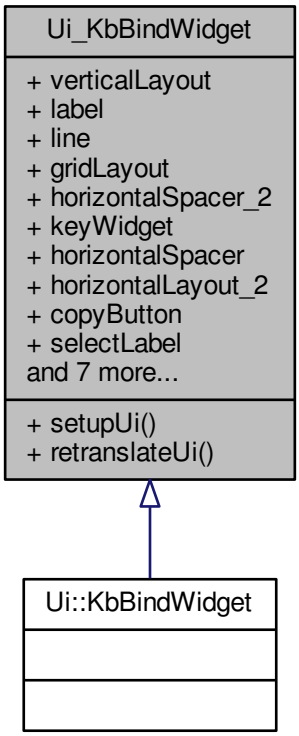
The documentation for this class was generated from the following file:

- src/ckb/ui_kbanimwidget.h

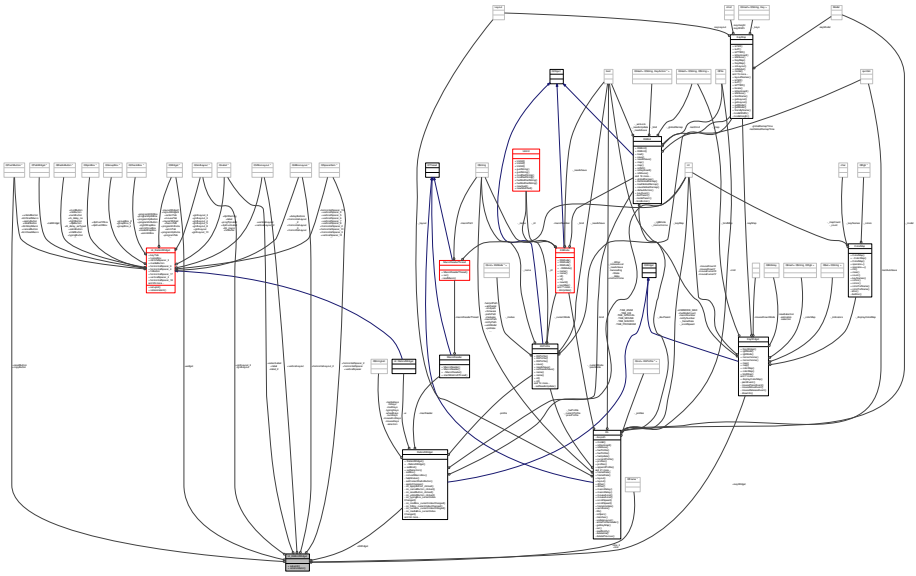
12.92 Ui_KbBindWidget Class Reference

`#include <src/ckb/ui_kbbindwidget.h>`

Inheritance diagram for Ui_KbBindWidget:



Collaboration diagram for Ui_KbBindWidget:



Public Member Functions

- void setupUi (QWidget *KbBindWidget)
- void retranslateUi (QWidget *KbBindWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QLabel * label
- QFrame * line
- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer_2
- KeyWidget * keyWidget
- QSpacerItem * horizontalSpacer
- QHBoxLayout * horizontalLayout_2
- QPushButton * copyButton
- QLabel * selectLabel
- QPushButton * resetButton
- QSpacerItem * verticalSpacer
- QLabel * label_2
- QFrame * line_2
- QWidget * widget
- QGridLayout * gridLayout_2
- RebindWidget * rbWidget

12.92.1 Detailed Description

Definition at line 30 of file ui_kbbindwidget.h.

12.92.2 Member Function Documentation

12.92.2.1 void Ui_KbBindWidget::retranslateUi (QWidget * *KbBindWidget*) [inline]

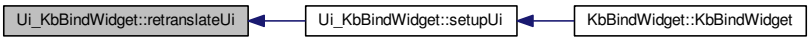
Definition at line 189 of file ui_kbbindwidget.h.

References copyButton, label, label_2, resetButton, and selectLabel.

Referenced by setupUi().

```
190     {
191         KbBindWidget->setWindowTitle(QApplication::translate("KbBindWidget", "Form", Q_NULLPTR));
192         label->setText(QApplication::translate("KbBindWidget", "Keys", Q_NULLPTR));
193         copyButton->setText(QApplication::translate("KbBindWidget", "Copy to mode...", Q_NULLPTR)
    );
194         selectLabel->setText(QApplication::translate("KbBindWidget", "Click to select keys",
    Q_NULLPTR));
195         resetButton->setText(QApplication::translate("KbBindWidget", "Reset", Q_NULLPTR));
196         label_2->setText(QApplication::translate("KbBindWidget", "Binding", Q_NULLPTR));
197     } // retranslateUi
```

Here is the caller graph for this function:



12.92.2.2 void Ui_KbBindWidget::setupUi (QWidget * *KbBindWidget*) [inline]

Definition at line 51 of file ui_kbbindwidget.h.

References copyButton, gridLayout, gridLayout_2, horizontalLayout_2, horizontalSpacer, horizontalSpacer_2, keyWidget, label, label_2, line, line_2, rbWidget, resetButton, retranslateUi(), selectLabel, verticalLayout, verticalSpacer, and widget.

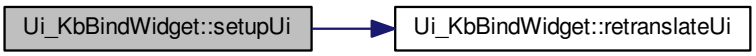
Referenced by KbBindWidget::KbBindWidget().

```
52     {
53         if (KbBindWidget->objectName().isEmpty())
54             KbBindWidget->setObjectName(QStringLiteral("KbBindWidget"));
55         KbBindWidget->resize(581, 473);
56         verticalLayout = new QVBoxLayout(KbBindWidget);
57         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
58         verticalLayout->setContentsMargins(0, 0, 0, -1);
59         label = new QLabel(KbBindWidget);
60         label->setObjectName(QStringLiteral("label"));
61         QFont font;
62         font.setBold(true);
63         font.setWeight(75);
64         label->setFont(font);
65
66         verticalLayout->addWidget(label);
67
68         line = new QFrame(KbBindWidget);
69         line->setObjectName(QStringLiteral("line"));
70         line->setFrameShape(QFrame::HLine);
71         line->setFrameShadow(QFrame::Sunken);
72
73         verticalLayout->addWidget(line);
74
75         gridLayout = new QGridLayout();
76         gridLayout->setObjectName(QStringLiteral("gridLayout"));
77         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
78
79         gridLayout->addItem(horizontalSpacer_2, 0, 4, 1, 1);
80
81         keyWidget = new KeyWidget(KbBindWidget);
82         keyWidget->setObjectName(QStringLiteral("keyWidget"));
```

```
83          QSizePolicy sizePolicy(QSizePolicy::Preferred, QSizePolicy::Preferred);
84          sizePolicy.setHorizontalStretch(0);
85          sizePolicy.setVerticalStretch(0);
86          sizePolicy.setHeightForWidth(keyWidget->sizePolicy().hasHeightForWidth());
87          keyWidget->setSizePolicy(sizePolicy);
88
89          gridLayout->addWidget(keyWidget, 0, 1, 1, 3);
90
91          horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
92
93          gridLayout->addItem(horizontalSpacer, 0, 0, 1, 1);
94
95          horizontalLayout_2 = new QHBoxLayout();
96          horizontalLayout_2->setSpacing(0);
97          horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
98          copyButton = new QPushButton(KbBindWidget);
99          copyButton->setObjectName(QStringLiteral("copyButton"));
100          QSizePolicy sizePolicy1(QSizePolicy::Fixed, QSizePolicy::Fixed);
101          sizePolicy1.setHorizontalStretch(0);
102          sizePolicy1.setVerticalStretch(0);
103          sizePolicy1.setHeightForWidth(copyButton->sizePolicy().hasHeightForWidth());
104          copyButton->setSizePolicy(sizePolicy1);
105          copyButton->setMinimumSize(QSize(140, 0));
106          copyButton->setMaximumSize(QSize(140, 16777215));
107
108          horizontalLayout_2->addWidget(copyButton);
109
110          selectLabel = new QLabel(KbBindWidget);
111          selectLabel->setObjectName(QStringLiteral("selectLabel"));
112          QSizePolicy sizePolicy2(QSizePolicy::Expanding, QSizePolicy::Preferred);
113          sizePolicy2.setHorizontalStretch(0);
114          sizePolicy2.setVerticalStretch(0);
115          sizePolicy2.setHeightForWidth(selectLabel->sizePolicy().hasHeightForWidth());
116          selectLabel->setSizePolicy(sizePolicy2);
117          selectLabel->setAlignment(Qt::AlignCenter);
118
119          horizontalLayout_2->addWidget(selectLabel);
120
121          resetButton = new QPushButton(KbBindWidget);
122          resetButton->setObjectName(QStringLiteral("resetButton"));
123          QSizePolicy sizePolicy3(QSizePolicy::Expanding, QSizePolicy::Fixed);
124          sizePolicy3.setHorizontalStretch(0);
125          sizePolicy3.setVerticalStretch(0);
126          sizePolicy3.setHeightForWidth(resetButton->sizePolicy().hasHeightForWidth());
127          resetButton->setSizePolicy(sizePolicy3);
128          resetButton->setMinimumSize(QSize(140, 0));
129          resetButton->setMaximumSize(QSize(140, 16777215));
130
131          horizontalLayout_2->addWidget(resetButton);
132
133
134          gridLayout->addLayout(horizontalLayout_2, 1, 1, 1, 3);
135
136
137          verticalLayout->addLayout(gridLayout);
138
139          verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
140
141          verticalLayout->addItem(verticalSpacer);
142
143          label_2 = new QLabel(KbBindWidget);
144          label_2->setObjectName(QStringLiteral("label_2"));
145          label_2->setFont(font);
146
147          verticalLayout->addWidget(label_2);
148
149          line_2 = new QFrame(KbBindWidget);
150          line_2->setObjectName(QStringLiteral("line_2"));
151          line_2->setFrameShape(QFrame::HLine);
152          line_2->setFrameShadow(QFrame::Sunken);
153
154          verticalLayout->addWidget(line_2);
155
156          widget = new QWidget(KbBindWidget);
157          widget->setObjectName(QStringLiteral("widget"));
158          QSizePolicy sizePolicy4(QSizePolicy::Preferred, QSizePolicy::Fixed);
159          sizePolicy4.setHorizontalStretch(0);
160          sizePolicy4.setVerticalStretch(0);
161          sizePolicy4.setHeightForWidth(widget->sizePolicy().hasHeightForWidth());
162          widget->setSizePolicy(sizePolicy4);
163          widget->setMinimumSize(QSize(0, 280));
164          widget->setMaximumSize(QSize(16777215, 280));
165          gridLayout_2 = new QGridLayout(widget);
166          gridLayout_2->setSpacing(0);
167          gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
```

```
168         gridLayout_2->setContentsMargins(0, 0, 0, 0);
169         rbWidget = new RebindWidget(widget);
170         rbWidget->setObjectName(QStringLiteral("rbWidget"));
171         QSizePolicy sizePolicy5(QSizePolicy::Expanding, QSizePolicy::Expanding);
172         sizePolicy5.setHorizontalStretch(0);
173         sizePolicy5.setVerticalStretch(0);
174         sizePolicy5.setHeightForWidth(rbWidget->sizePolicy().hasHeightForWidth());
175         rbWidget->setSizePolicy(sizePolicy5);
176
177         gridLayout_2->addWidget(rbWidget, 0, 0, 1, 1);
178
179
180         verticalLayout->addWidget(widget);
181
182         QWidget::setTabOrder(copyButton, resetButton);
183
184         retranslateUi(KbBindWidget);
185
186         QMetaObject::connectSlotsByName(KbBindWidget);
187     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.92.3 Field Documentation

12.92.3.1 QPushButton* Ui_KbBindWidget::copyButton

Definition at line 41 of file ui_kbbindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.92.3.2 QGridLayout* Ui_KbBindWidget::gridLayout

Definition at line 36 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.3 QGridLayout* Ui_KbBindWidget::gridLayout_2

Definition at line 48 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.4 QHBoxLayout* Ui_KbBindWidget::horizontalLayout_2

Definition at line 40 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.5 QSpacerItem* Ui_KbBindWidget::horizontalSpacer

Definition at line 39 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.6 QSpacerItem* Ui_KbBindWidget::horizontalSpacer_2

Definition at line 37 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.7 KeyWidget* Ui_KbBindWidget::keyWidget

Definition at line 38 of file ui_kbbindwidget.h.

Referenced by KbBindWidget::KbBindWidget(), KbBindWidget::newLayout(), KbBindWidget::setBind(), setupUi(), and KbBindWidget::updateBind().

12.92.3.8 QLabel* Ui_KbBindWidget::label

Definition at line 34 of file ui_kbbindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.92.3.9 QLabel* Ui_KbBindWidget::label_2

Definition at line 45 of file ui_kbbindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.92.3.10 QFrame* Ui_KbBindWidget::line

Definition at line 35 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.11 QFrame* Ui_KbBindWidget::line_2

Definition at line 46 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.12 RebindWidget* Ui_KbBindWidget::rbWidget

Definition at line 49 of file ui_kbbindwidget.h.

Referenced by KbBindWidget::newLayout(), KbBindWidget::newSelection(), KbBindWidget::setBind(), setupUi(), and KbBindWidget::updateBind().

12.92.3.13 QPushButton* Ui_KbBindWidget::resetButton

Definition at line 43 of file ui_kbbindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.92.3.14 QLabel* Ui_KbBindWidget::selectLabel

Definition at line 42 of file ui_kbbindwidget.h.

Referenced by retranslateUi(), setupUi(), and KbBindWidget::updateSelDisplay().

12.92.3.15 QVBoxLayout* Ui_KbBindWidget::verticalLayout

Definition at line 33 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.16 QSpacerItem* Ui_KbBindWidget::verticalSpacer

Definition at line 44 of file ui_kbbindwidget.h.

Referenced by setupUi().

12.92.3.17 QWidget* Ui_KbBindWidget::widget

Definition at line 47 of file ui_kbbindwidget.h.

Referenced by setupUi().

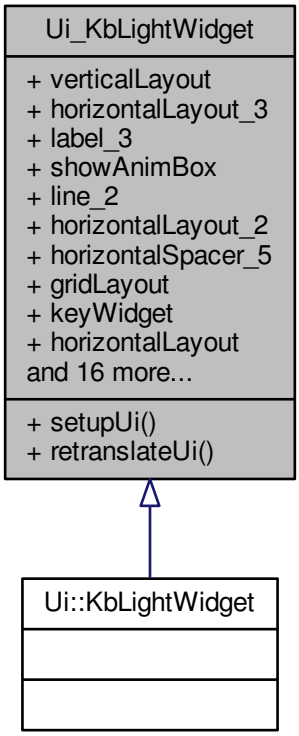
The documentation for this class was generated from the following file:

- src/ckb/ui_kbbindwidget.h

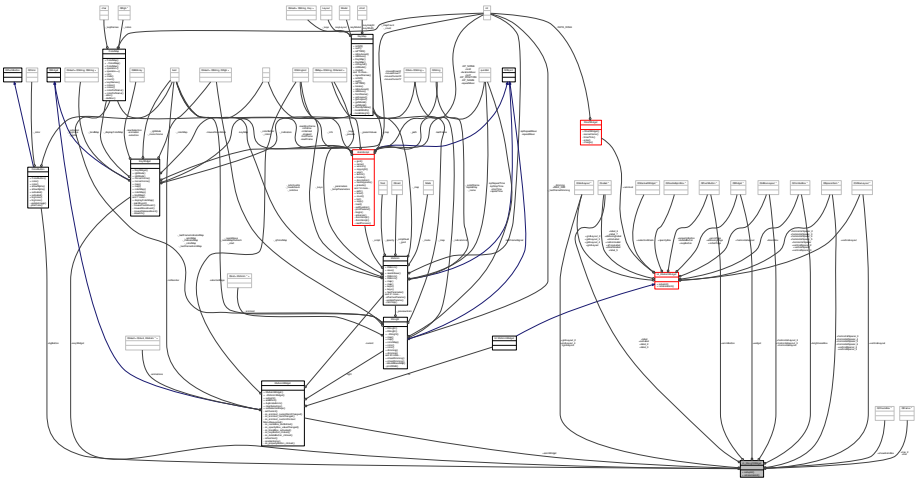
12.93 Ui_KbLightWidget Class Reference

`#include <src/ckb/ui_kblightwidget.h>`

Inheritance diagram for Ui_KbLightWidget:



Collaboration diagram for Ui_KbLightWidget:



Public Member Functions

- void setupUi (QWidget *KbLightWidget)
- void retranslateUi (QWidget *KbLightWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QHBoxLayout * horizontalLayout_3
- QLabel * label_3
- QCheckBox * showAnimBox
- QFrame * line_2
- QHBoxLayout * horizontalLayout_2
- QSpacerItem * horizontalSpacer_5
- QGridLayout * gridLayout
- KeyWidget * keyWidget
- QHBoxLayout * horizontalLayout
- ColorButton * bgButton
- QLabel * selLabel
- QPushButton * animButton
- QSpacerItem * horizontalSpacer_6
- QSpacerItem * verticalSpacer_2
- QGridLayout * gridLayout_2
- QComboBox * brightnessBox
- QLabel * label
- QSpacerItem * horizontalSpacer_3
- QSpacerItem * horizontalSpacer_4
- QSpacerItem * verticalSpacer_3
- QLabel * label_2
- QFrame * line
- QWidget * widget
- QGridLayout * gridLayout_4
- KbAnimWidget * animWidget

12.93.1 Detailed Description

Definition at line 33 of file ui_kblightwidget.h.

12.93.2 Member Function Documentation

12.93.2.1 void Ui_KbLightWidget::retranslateUi (QWidget * *KbLightWidget*) [inline]

Definition at line 241 of file ui_kblightwidget.h.

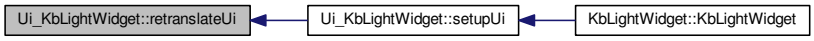
References animButton, bgButton, brightnessBox, label, label_2, label_3, selLabel, and showAnimBox.

Referenced by setupUi().

```
242     {
243         KbLightWidget->setWindowTitle(QApplication::translate("KbLightWidget", "Form", Q_NULLPTR));
244         label_3->setText(QApplication::translate("KbLightWidget", "Colors", Q_NULLPTR));
245 #ifndef QT_NO_TOOLTIP
246         showAnimBox->setToolTip(QApplication::translate("KbLightWidget", "If enabled, the same
      colors will be displayed here as on the keyboard. If disabled, the base colors will be shown.", Q_NULLPTR));
247 #endif // QT_NO_TOOLTIP
248         showAnimBox->setText(QApplication::translate("KbLightWidget", "Show animated", Q_NULLPTR
    ));
249         bgButton->setText(QApplication::translate("KbLightWidget", "Change color...", Q_NULLPTR));
250         selLabel->setText(QApplication::translate("KbLightWidget", "Click to select", Q_NULLPTR));
251         animButton->setText(QApplication::translate("KbLightWidget", "New animation...",
    Q_NULLPTR));
252         brightnessBox->clear();
253         brightnessBox->insertItems(0, QStringList()
254          << QApplication::translate("KbLightWidget", "100%", Q_NULLPTR)
255          << QApplication::translate("KbLightWidget", "67%", Q_NULLPTR)
256          << QApplication::translate("KbLightWidget", "33%", Q_NULLPTR)
257          << QApplication::translate("KbLightWidget", "Off", Q_NULLPTR)
```

```
258         );
259         label->setText(QApplication::translate("KbLightWidget", "\342\230\274", Q_NULLPTR));
260         label_2->setText(QApplication::translate("KbLightWidget", "Animations", Q_NULLPTR));
261     } // retranslateUi
```

Here is the caller graph for this function:



12.93.2.2 void Ui_KbLightWidget::setupUi (QWidget * *KbLightWidget*) [inline]

Definition at line 63 of file ui_kblightwidget.h.

References animButton, animWidget, bgButton, brightnessBox, gridLayout, gridLayout_2, gridLayout_4, horizontalLayout, horizontalLayout_2, horizontalLayout_3, horizontalSpacer_3, horizontalSpacer_4, horizontalSpacer_5, horizontalSpacer_6, keyWidget, label, label_2, label_3, line, line_2, retranslateUi(), selLabel, showAnimBox, verticalLayout, verticalSpacer_2, verticalSpacer_3, and widget.

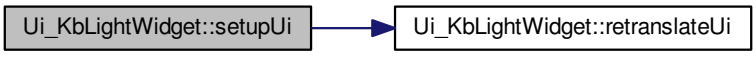
Referenced by KbLightWidget::KbLightWidget().

```
64     {
65         if (KbLightWidget->objectName().isEmpty())
66             KbLightWidget->setObjectName(QStringLiteral("KbLightWidget"));
67         KbLightWidget->resize(632, 540);
68         KbLightWidget->setFocusPolicy(Qt::ClickFocus);
69         verticalLayout = new QVBoxLayout(KbLightWidget);
70         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
71         verticalLayout->setContentsMargins(0, 0, 0, -1);
72         horizontalLayout_3 = new QHBoxLayout();
73         horizontalLayout_3->setObjectName(QStringLiteral("horizontalLayout_3"));
74         label_3 = new QLabel(KbLightWidget);
75         label_3->setObjectName(QStringLiteral("label_3"));
76         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
77         sizePolicy.setHorizontalStretch(0);
78         sizePolicy.setVerticalStretch(0);
79         sizePolicy.setHeightForWidth(label_3->sizePolicy().hasHeightForWidth());
80         label_3->setSizePolicy(sizePolicy);
81         QFont font;
82         font.setBold(true);
83         font.setWeight(75);
84         label_3->setFont(font);
85
86         horizontalLayout_3->addWidget(label_3);
87
88         showAnimBox = new QCheckBox(KbLightWidget);
89         showAnimBox->setObjectName(QStringLiteral("showAnimBox"));
90         showAnimBox->setChecked(true);
91
92         horizontalLayout_3->addWidget(showAnimBox);
93
94
95         verticalLayout->addLayout(horizontalLayout_3);
96
97         line_2 = new QFrame(KbLightWidget);
98         line_2->setObjectName(QStringLiteral("line_2"));
99         line_2->setFrameShape(QFrame::HLine);
100         line_2->setFrameShadow(QFrame::Sunken);
101
102         verticalLayout->addWidget(line_2);
103
104         horizontalLayout_2 = new QHBoxLayout();
105         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
106         horizontalSpacer_5 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
      QSizePolicy::Minimum);
107
108         horizontalLayout_2->addItem(horizontalSpacer_5);
109
110         gridLayout = new QGridLayout();
111         gridLayout->setObjectName(QStringLiteral("gridLayout"));
112         keyWidget = new KeyWidget(KbLightWidget);
```

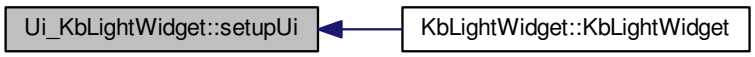
```
113         keyWidget->setObjectName(QStringLiteral("keyWidget"));
114         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Expanding);
115         sizePolicy1.setHorizontalStretch(0);
116         sizePolicy1.setVerticalStretch(0);
117         sizePolicy1.setHeightForWidth(keyWidget->sizePolicy().hasHeightForWidth());
118         keyWidget->setSizePolicy(sizePolicy1);
119         keyWidget->setMinimumSize(QSize(100, 100));
120         keyWidget->setFocusPolicy(Qt::ClickFocus);
121 
122         gridLayout->addWidget(keyWidget, 0, 0, 1, 3);
123 
124         horizontalLayout = new QHBoxLayout();
125         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
126         bgButton = new ColorButton(KbLightWidget);
127         bgButton->setObjectName(QStringLiteral("bgButton"));
128         QSizePolicy sizePolicy2(QSizePolicy::Fixed, QSizePolicy::Fixed);
129         sizePolicy2.setHorizontalStretch(0);
130         sizePolicy2.setVerticalStretch(0);
131         sizePolicy2.setHeightForWidth(bgButton->sizePolicy().hasHeightForWidth());
132         bgButton->setSizePolicy(sizePolicy2);
133         bgButton->setMinimumSize(QSize(140, 0));
134         bgButton->setMaximumSize(QSize(140, 16777215));
135 
136         horizontalLayout->addWidget(bgButton);
137 
138         selLabel = new QLabel(KbLightWidget);
139         selLabel->setObjectName(QStringLiteral("selLabel"));
140         sizePolicy.setHeightForWidth(selLabel->sizePolicy().hasHeightForWidth());
141         selLabel->setSizePolicy(sizePolicy);
142         selLabel->setAlignment(Qt::AlignCenter);
143 
144         horizontalLayout->addWidget(selLabel);
145 
146         animButton = new QPushButton(KbLightWidget);
147         animButton->setObjectName(QStringLiteral("animButton"));
148         sizePolicy2.setHeightForWidth(animButton->sizePolicy().hasHeightForWidth());
149         animButton->setSizePolicy(sizePolicy2);
150         animButton->setMinimumSize(QSize(140, 0));
151         animButton->setMaximumSize(QSize(140, 16777215));
152 
153         horizontalLayout->addWidget(animButton);
154 
155 
156         gridLayout->addLayout(horizontalLayout, 1, 0, 1, 3);
157 
158 
159         horizontalLayout_2->addLayout(gridLayout);
160 
161         horizontalSpacer_6 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
162 
163         horizontalLayout_2->addItem(horizontalSpacer_6);
164 
165 
166         verticalLayout->addLayout(horizontalLayout_2);
167 
168         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
169 
170         verticalLayout->addItem(verticalSpacer_2);
171 
172         gridLayout_2 = new QGridLayout();
173         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
174         brightnessBox = new QComboBox(KbLightWidget);
175         brightnessBox->setObjectName(QStringLiteral("brightnessBox"));
176         sizePolicy2.setHeightForWidth(brightnessBox->sizePolicy().hasHeightForWidth());
177         brightnessBox->setSizePolicy(sizePolicy2);
178 
179         gridLayout_2->addWidget(brightnessBox, 1, 3, 1, 1);
180 
181         label = new QLabel(KbLightWidget);
182         label->setObjectName(QStringLiteral("label"));
183         QFont font1;
184         font1.setPointSize(20);
185         label->setFont(font1);
186         label->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
187 
188         gridLayout_2->addWidget(label, 1, 2, 1, 1);
189 
190         horizontalSpacer_3 = new QSpacerItem(0, 0, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
191 
192         gridLayout_2->addItem(horizontalSpacer_3, 0, 4, 1, 1);
193 
194         horizontalSpacer_4 = new QSpacerItem(0, 0, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
195 
```

```
196         gridLayout_2->addItem(horizontalSpacer_4, 0, 2, 1, 1);
197
198
199         verticalLayout->addLayout(gridLayout_2);
200
201         verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
202
203         verticalLayout->addItem(verticalSpacer_3);
204
205         label_2 = new QLabel(KbLightWidget);
206         label_2->setObjectName(QStringLiteral("label_2"));
207         label_2->setFont(font);
208
209         verticalLayout->addWidget(label_2);
210
211         line = new QFrame(KbLightWidget);
212         line->setObjectName(QStringLiteral("line"));
213         line->setFrameShape(QFrame::HLine);
214         line->setFrameShadow(QFrame::Sunken);
215
216         verticalLayout->addWidget(line);
217
218         widget = new QWidget(KbLightWidget);
219         widget->setObjectName(QStringLiteral("widget"));
220         widget->setMinimumSize(QSize(0, 170));
221         gridLayout_4 = new QGridLayout(widget);
222         gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
223         gridLayout_4->setContentsMargins(-1, 0, -1, 0);
224         animWidget = new KbAnimWidget(widget);
225         animWidget->setObjectName(QStringLiteral("animWidget"));
226
227         gridLayout_4->addWidget(animWidget, 0, 0, 1, 1);
228
229
230         verticalLayout->addWidget(widget);
231
232         QWidget::setTabOrder(showAnimBox, bgButton);
233         QWidget::setTabOrder(bgButton, animButton);
234         QWidget::setTabOrder(animButton, brightnessBox);
235
236         retranslateUi(KbLightWidget);
237
238         QMetaObject::connectSlotsByName(KbLightWidget);
239     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.93.3 Field Documentation

12.93.3.1 QPushButton* Ui_KbLightWidget::animButton

Definition at line 48 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), retranslateUi(), and setupUi().

12.93.3.2 KbAnimWidget* Ui_KbLightWidget::animWidget

Definition at line 61 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), KbLightWidget::newSelection(), KbLightWidget::on_animButton_-clicked(), KbLightWidget::setLight(), and setupUi().

12.93.3.3 ColorButton* Ui_KbLightWidget::bgButton

Definition at line 46 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), KbLightWidget::newSelection(), retranslateUi(), and setupUi().

12.93.3.4 QComboBox* Ui_KbLightWidget::brightnessBox

Definition at line 52 of file ui_kblightwidget.h.

Referenced by retranslateUi(), setupUi(), and KbLightWidget::updateLight().

12.93.3.5 QGridLayout* Ui_KbLightWidget::gridLayout

Definition at line 43 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.6 QGridLayout* Ui_KbLightWidget::gridLayout_2

Definition at line 51 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.7 QGridLayout* Ui_KbLightWidget::gridLayout_4

Definition at line 60 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.8 QHBoxLayout* Ui_KbLightWidget::horizontalLayout

Definition at line 45 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.9 QHBoxLayout* Ui_KbLightWidget::horizontalLayout_2

Definition at line 41 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.10 QHBoxLayout* Ui_KbLightWidget::horizontalLayout_3

Definition at line 37 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.11 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_3

Definition at line 54 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.12 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_4

Definition at line 55 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.13 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_5

Definition at line 42 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.14 QSpacerItem* Ui_KbLightWidget::horizontalSpacer_6

Definition at line 49 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.15 KeyWidget* Ui_KbLightWidget::keyWidget

Definition at line 44 of file ui_kblightwidget.h.

Referenced by KbLightWidget::changeAnim(), KbLightWidget::changeAnimKeys(), KbLightWidget::changeColor(), KbLightWidget::KbLightWidget(), KbLightWidget::on_animButton_clicked(), KbLightWidget::on_bgButton_clicked(), KbLightWidget::on_showAnimBox_clicked(), KbLightWidget::setLight(), KbLightWidget::setMonochrome(), setupUi(), KbLightWidget::toggleSidelight(), and KbLightWidget::updateLight().

12.93.3.16 QLabel* Ui_KbLightWidget::label

Definition at line 53 of file ui_kblightwidget.h.

Referenced by retranslateUi(), and setupUi().

12.93.3.17 QLabel* Ui_KbLightWidget::label_2

Definition at line 57 of file ui_kblightwidget.h.

Referenced by retranslateUi(), and setupUi().

12.93.3.18 QLabel* Ui_KbLightWidget::label_3

Definition at line 38 of file ui_kblightwidget.h.

Referenced by retranslateUi(), and setupUi().

12.93.3.19 QFrame* Ui_KbLightWidget::line

Definition at line 58 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.20 QFrame* Ui_KbLightWidget::line_2

Definition at line 40 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.21 QLabel* Ui_KbLightWidget::selLabel

Definition at line 47 of file ui_kblightwidget.h.

Referenced by KbLightWidget::newSelection(), retranslateUi(), and setupUi().

12.93.3.22 QCheckBox* Ui_KbLightWidget::showAnimBox

Definition at line 39 of file ui_kblightwidget.h.

Referenced by KbLightWidget::KbLightWidget(), retranslateUi(), KbLightWidget::setLight(), and setupUi().

12.93.3.23 QVBoxLayout* Ui_KbLightWidget::verticalLayout

Definition at line 36 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.24 QSpacerItem* Ui_KbLightWidget::verticalSpacer_2

Definition at line 50 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.25 QSpacerItem* Ui_KbLightWidget::verticalSpacer_3

Definition at line 56 of file ui_kblightwidget.h.

Referenced by setupUi().

12.93.3.26 QWidget* Ui_KbLightWidget::widget

Definition at line 59 of file ui_kblightwidget.h.

Referenced by setupUi().

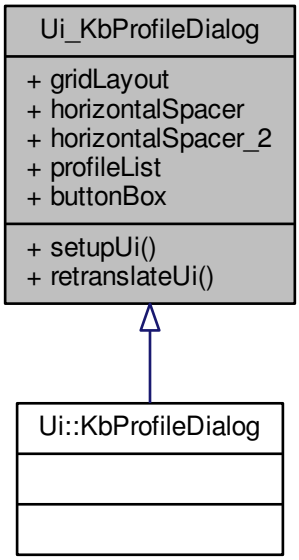
The documentation for this class was generated from the following file:

- src/ckb/ui_kblightwidget.h

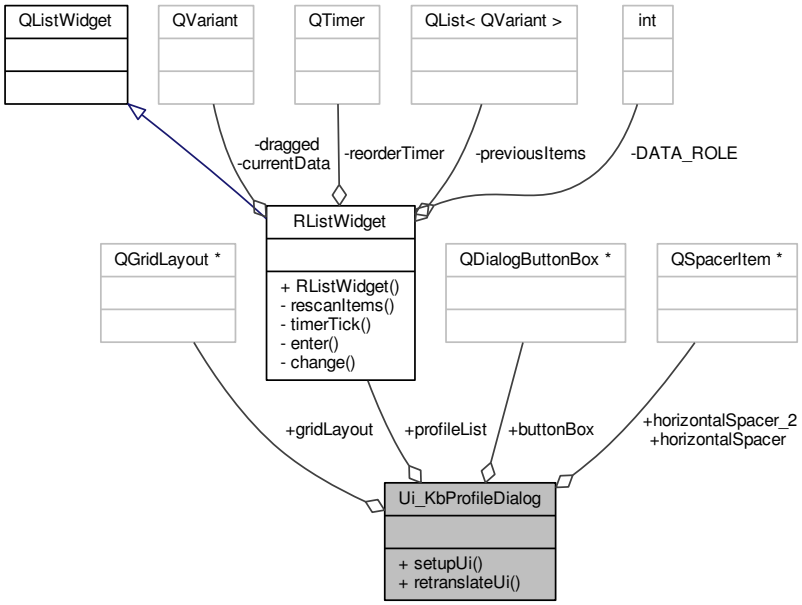
12.94 Ui_KbProfileDialog Class Reference

`#include <src/ckb/ui_kbprofiledialog.h>`

Inheritance diagram for Ui_KbProfileDialog:



Collaboration diagram for Ui_KbProfileDialog:



Public Member Functions

- void setupUi (QDialog *KbProfileDialog)
- void retranslateUi (QDialog *KbProfileDialog)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * horizontalSpacer
- QSpacerItem * horizontalSpacer_2
- RListWidget * profileList
- QDialogButtonBox * buttonBox

12.94.1 Detailed Description

Definition at line 25 of file ui_kbprofiledialog.h.

12.94.2 Member Function Documentation

12.94.2.1 void Ui_KbProfileDialog::retranslateUi (QDialog * *KbProfileDialog*) [inline]

Definition at line 73 of file ui_kbprofiledialog.h.

Referenced by setupUi().

```
74     {
75         KbProfileDialog->setWindowTitle(QApplication::translate("KbProfileDialog", "Keyboard Profiles",
       Q_NULLPTR));
76     } // retranslateUi
```

Here is the caller graph for this function:

12.94.2.2 void Ui_KbProfileDialog::setupUi (QDialog * *KbProfileDialog*) [inline]

Definition at line 34 of file ui_kbprofiledialog.h.

References buttonBox, gridLayout, horizontalSpacer, horizontalSpacer_2, profileList, and retranslateUi().

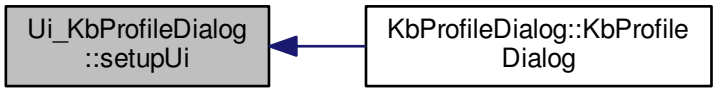
Referenced by KbProfileDialog::KbProfileDialog().

```
35     {
36         if (KbProfileDialog->objectName().isEmpty())
37             KbProfileDialog->setObjectName(QStringLiteral("KbProfileDialog"));
38         KbProfileDialog->resize(400, 300);
39         KbProfileDialog->setFocusPolicy(Qt::ClickFocus);
40         gridLayout = new QGridLayout(KbProfileDialog);
41         gridLayout->setObjectName(QStringLiteral("gridLayout"));
42         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
       QSizePolicy::Minimum);
43
44         gridLayout->addItem(horizontalSpacer, 0, 0, 1, 1);
```

```
45
46         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
47
48         gridLayout->addItem(horizontalSpacer_2, 0, 2, 1, 1);
49
50         profileList = new RListWidget(KbProfileDialog);
51         profileList->setObjectName(QStringLiteral("profileList"));
52         profileList->setContextMenuPolicy(Qt::CustomContextMenu);
53         profileList->setFrameShadow(QFrame::Plain);
54         profileList->setIconSize(QSize(24, 24));
55
56         gridLayout->addWidget(profileList, 0, 1, 1, 1);
57
58         buttonBox = new QDialogButtonBox(KbProfileDialog);
59         buttonBox->setObjectName(QStringLiteral("buttonBox"));
60         buttonBox->setOrientation(Qt::Horizontal);
61         buttonBox->setStandardButtons(QDialogButtonBox::Ok);
62
63         gridLayout->addWidget(buttonBox, 1, 0, 1, 3);
64
65
66         retranslateUi(KbProfileDialog);
67         QObject::connect(buttonBox, SIGNAL(accepted()), KbProfileDialog, SLOT(accept()));
68         QObject::connect(buttonBox, SIGNAL(rejected()), KbProfileDialog, SLOT(reject()));
69
70         QMetaObject::connectSlotsByName(KbProfileDialog);
71     } // setupUi
```

Here is the call graph for this function:

Here is the caller graph for this function:



12.94.3 Field Documentation

12.94.3.1 QDialogButtonBox* Ui_KbProfileDialog::buttonBox

Definition at line 32 of file ui_kbprofiledialog.h.

Referenced by setupUi().

12.94.3.2 QGridLayout* Ui_KbProfileDialog::gridLayout

Definition at line 28 of file ui_kbprofiledialog.h.

Referenced by setupUi().

12.94.3.3 QSpacerItem* Ui_KbProfileDialog::horizontalSpacer

Definition at line 29 of file ui_kbprofiledialog.h.

Referenced by setupUi().

12.94.3.4 QSpacerItem* Ui_KbProfileDialog::horizontalSpacer_2

Definition at line 30 of file ui_kbprofiledialog.h.

Referenced by setupUi().

12.94.3.5 RListWidget* Ui_KbProfileDialog::profileList

Definition at line 31 of file ui_kbprofiledialog.h.

Referenced by KbProfileDialog::addNewProfileItem(), KbProfileDialog::KbProfileDialog(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_itemClicked(), KbProfileDialog::profileList_reordered(), KbProfileDialog::repopulate(), and setupUi().

The documentation for this class was generated from the following file:

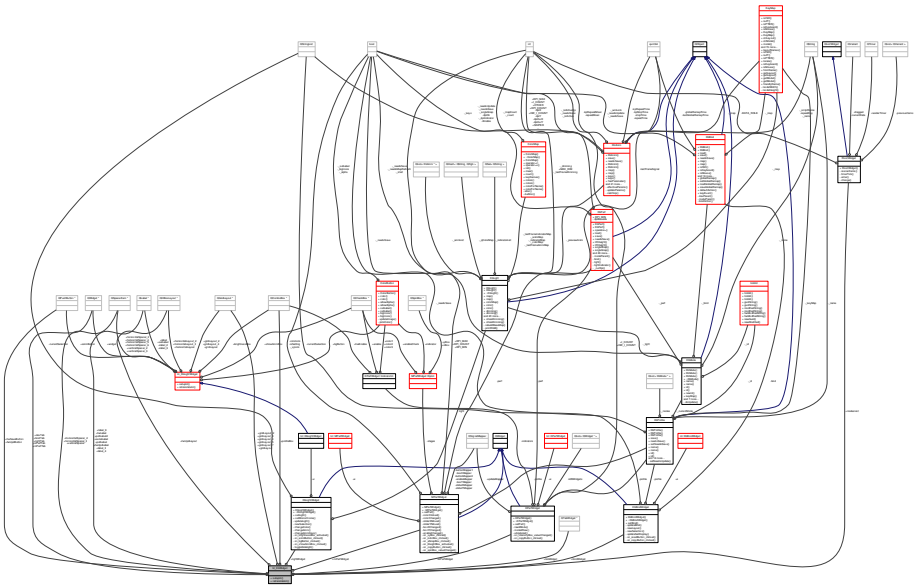
- src/ckb/ui_kbprofiledialog.h

12.95 Ui_KbWidget Class Reference

`#include <src/ckb/ui_kbwidget.h>`

Inheritance diagram for Ui_KbWidget:

Collaboration diagram for Ui_KbWidget:



Public Member Functions

- void setupUi (QWidget *KbWidget)
- void retranslateUi (QWidget *KbWidget)

Data Fields

- QGridLayout * gridLayout
- QComboBox * profileBox
- QTabWidget * tabWidget
- QWidget * lightTab
- QGridLayout * gridLayout_2
- KbLightWidget * lightWidget
- QLabel * label_3
- QWidget * bindTab
- QGridLayout * gridLayout_5
- KbBindWidget * bindWidget
- QWidget * kPerfTab
- QGridLayout * gridLayout_4
- KPerfWidget * kPerfWidget
- QWidget * mPerfTab
- QGridLayout * gridLayout_7
- MPerfWidget * mPerfWidget
- QWidget * devTab
- QGridLayout * gridLayout_3
- QLabel * fwUpdLabel
- QSpacerItem * verticalSpacer
- QLabel * fwLabel
- QLabel * label_4
- QSpacerItem * horizontalSpacer_2
- QLabel * label_9
- QLabel * serialLabel
- QHBoxLayout * fwUpdLayout
- QPushButton * fwUpdButton
- QSpacerItem * horizontalSpacer_3
- QLabel * pollLabel2
- QLabel * pollLabel
- RListWidget * modesList
- QPushButton * hwSaveButton

12.95.1 Detailed Description

Definition at line 33 of file ui_kbwidget.h.

12.95.2 Member Function Documentation

12.95.2.1 void Ui_KbWidget::retranslateUi (QWidget * *KbWidget*) [inline]

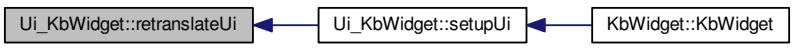
Definition at line 268 of file ui_kbwidget.h.

References bindTab, devTab, fwLabel, fwUpdButton, fwUpdLabel, hwSaveButton, kPerfTab, label_3, label_4, label_9, lightTab, mPerfTab, pollLabel, pollLabel2, serialLabel, and tabWidget.

Referenced by setupUi().

```
269     {
270         KbWidget->setWindowTitle(QApplication::translate("KbWidget", "Form", Q_NULLPTR));
271         label_3->setText(QApplication::translate("KbWidget", "Tip: Drag+drop items to reorder.
     Right-click for menu.", Q_NULLPTR));
272         tabWidget->setTabText(tabWidget->indexOf(lightTab),
    QApplication::translate("KbWidget", "Lighting", Q_NULLPTR));
273         tabWidget->setTabText(tabWidget->indexOf(bindTab), QApplication::translate
    ("KbWidget", "Binding", Q_NULLPTR));
274         tabWidget->setTabText(tabWidget->indexOf(kPerfTab),
    QApplication::translate("KbWidget", "Performance", Q_NULLPTR));
275         tabWidget->setTabText(tabWidget->indexOf(mPerfTab),
    QApplication::translate("KbWidget", "Performance", Q_NULLPTR));
276         fwUpdLabel->setText(QApplication::translate("KbWidget", "Status:", Q_NULLPTR));
277         fwLabel->setText(QApplication::translate("KbWidget", "N/A", Q_NULLPTR));
278         label_4->setText(QApplication::translate("KbWidget", "Serial Number:", Q_NULLPTR));
279         label_9->setText(QApplication::translate("KbWidget", "Firmware version:", Q_NULLPTR));
280         serialLabel->setText(QApplication::translate("KbWidget", "N/A", Q_NULLPTR));
281         fwUpdButton->setText(QApplication::translate("KbWidget", "Check for updates", Q_NULLPTR)
    );
282         pollLabel2->setText(QApplication::translate("KbWidget", "Poll rate:", Q_NULLPTR));
283         pollLabel->setText(QApplication::translate("KbWidget", "N/A", Q_NULLPTR));
284         tabWidget->setTabText(tabWidget->indexOf(devTab), QApplication::translate("
    KbWidget", "Device", Q_NULLPTR));
285         hwSaveButton->setText(QApplication::translate("KbWidget", "Save to hardware", Q_NULLPTR
    ));
286     } // retranslateUi
```

Here is the caller graph for this function:



12.95.2.2 void Ui_KbWidget::setupUi (QWidget * *KbWidget*) [inline]

Definition at line 69 of file ui_kbwidget.h.

References bindTab, bindWidget, devTab, fwLabel, fwUpdButton, fwUpdLabel, fwUpdLayout, gridLayout, gridLayout_2, gridLayout_3, gridLayout_4, gridLayout_5, gridLayout_7, horizontalSpacer_2, horizontalSpacer_3, hwSaveButton, kPerfTab, kPerfWidget, label_3, label_4, label_9, lightTab, lightWidget, modesList, mPerfTab, mPerfWidget, pollLabel, pollLabel2, profileBox, retranslateUi(), serialLabel, tabWidget, and verticalSpacer.

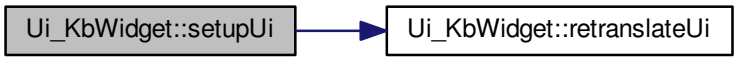
Referenced by KbWidget::KbWidget().

```
70     {
71         if (KbWidget->objectName().isEmpty())
72             KbWidget->setObjectName(QStringLiteral("KbWidget"));
73         KbWidget->resize(773, 539);
74         gridLayout = new QGridLayout(KbWidget);
75         gridLayout->setSpacing(9);
76         gridLayout->setObjectName(QStringLiteral("gridLayout"));
77         gridLayout->setContentsMargins(9, -1, 9, 9);
78         profileBox = new QComboBox(KbWidget);
79         profileBox->setObjectName(QStringLiteral("profileBox"));
80         QSizePolicy sizePolicy(QSizePolicy::Preferred, QSizePolicy::Fixed);
81         sizePolicy.setHorizontalStretch(18);
82         sizePolicy.setVerticalStretch(0);
83         sizePolicy.setHeightForWidth(profileBox->sizePolicy().hasHeightForWidth());
84         profileBox->setSizePolicy(sizePolicy);
85         profileBox->setIconSize(QSize(24, 24));
86
87         gridLayout->addWidget(profileBox, 0, 0, 1, 1);
88
89         tabWidget = new QTabWidget(KbWidget);
90         tabWidget->setObjectName(QStringLiteral("tabWidget"));
91         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Expanding);
92         sizePolicy1.setHorizontalStretch(82);
93         sizePolicy1.setVerticalStretch(0);
94         sizePolicy1.setHeightForWidth(tabWidget->sizePolicy().hasHeightForWidth());
95         tabWidget->setSizePolicy(sizePolicy1);
96         lightTab = new QWidget();
```

```
97          lightTab->setObjectName(QStringLiteral("lightTab"));
98          gridLayout_2 = new QGridLayout(lightTab);
99          gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
100          gridLayout_2->setHorizontalSpacing(0);
101          gridLayout_2->setVerticalSpacing(6);
102          gridLayout_2->setContentsMargins(6, 6, 6, 6);
103          lightWidget = new KbLightWidget(lightTab);
104          lightWidget->setObjectName(QStringLiteral("lightWidget"));
105          QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Expanding);
106          sizePolicy2.setHorizontalStretch(0);
107          sizePolicy2.setVerticalStretch(0);
108          sizePolicy2.setHeightForWidth(lightWidget->sizePolicy().hasHeightForWidth());
109          lightWidget->setSizePolicy(sizePolicy2);
110
111          gridLayout_2->addWidget(lightWidget, 0, 0, 1, 1);
112
113          label_3 = new QLabel(lightTab);
114          label_3->setObjectName(QStringLiteral("label_3"));
115          QSizePolicy sizePolicy3(QSizePolicy::Preferred, QSizePolicy::Minimum);
116          sizePolicy3.setHorizontalStretch(0);
117          sizePolicy3.setVerticalStretch(0);
118          sizePolicy3.setHeightForWidth(label_3->sizePolicy().hasHeightForWidth());
119          label_3->setSizePolicy(sizePolicy3);
120
121          gridLayout_2->addWidget(label_3, 1, 0, 1, 1);
122
123          tabWidget->addTab(lightTab, QString());
124          bindTab = new QWidget();
125          bindTab->setObjectName(QStringLiteral("bindTab"));
126          gridLayout_5 = new QGridLayout(bindTab);
127          gridLayout_5->setSpacing(0);
128          gridLayout_5->setObjectName(QStringLiteral("gridLayout_5"));
129          gridLayout_5->setContentsMargins(6, 6, 6, 6);
130          bindWidget = new KbBindWidget(bindTab);
131          bindWidget->setObjectName(QStringLiteral("bindWidget"));
132          sizePolicy2.setHeightForWidth(bindWidget->sizePolicy().hasHeightForWidth());
133          bindWidget->setSizePolicy(sizePolicy2);
134
135          gridLayout_5->addWidget(bindWidget, 0, 0, 1, 1);
136
137          tabWidget->addTab(bindTab, QString());
138          kPerfTab = new QWidget();
139          kPerfTab->setObjectName(QStringLiteral("kPerfTab"));
140          gridLayout_4 = new QGridLayout(kPerfTab);
141          gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
142          gridLayout_4->setContentsMargins(6, 6, 6, 6);
143          kPerfWidget = new KPerfWidget(kPerfTab);
144          kPerfWidget->setObjectName(QStringLiteral("kPerfWidget"));
145
146          gridLayout_4->addWidget(kPerfWidget, 0, 0, 1, 1);
147
148          tabWidget->addTab(kPerfTab, QString());
149          mPerfTab = new QWidget();
150          mPerfTab->setObjectName(QStringLiteral("mPerfTab"));
151          gridLayout_7 = new QGridLayout(mPerfTab);
152          gridLayout_7->setObjectName(QStringLiteral("gridLayout_7"));
153          gridLayout_7->setContentsMargins(6, 6, 6, 6);
154          mPerfWidget = new MPerfWidget(mPerfTab);
155          mPerfWidget->setObjectName(QStringLiteral("mPerfWidget"));
156
157          gridLayout_7->addWidget(mPerfWidget, 0, 0, 1, 1);
158
159          tabWidget->addTab(mPerfTab, QString());
160          devTab = new QWidget();
161          devTab->setObjectName(QStringLiteral("devTab"));
162          gridLayout_3 = new QGridLayout(devTab);
163          gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
164          gridLayout_3->setContentsMargins(6, 6, 6, 6);
165          fwUpdLabel = new QLabel(devTab);
166          fwUpdLabel->setObjectName(QStringLiteral("fwUpdLabel"));
167          fwUpdLabel->setMinimumSize(QSize(0, 34));
168
169          gridLayout_3->addWidget(fwUpdLabel, 4, 0, 1, 1);
170
171          verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
172
173          gridLayout_3->addItem(verticalSpacer, 6, 2, 1, 1);
174
175          fwLabel = new QLabel(devTab);
176          fwLabel->setObjectName(QStringLiteral("fwLabel"));
177          fwLabel->setCursor(QCursor(Qt::IBeamCursor));
178          fwLabel->setTextInteractionFlags(Qt::LinksAccessibleByMouse|Qt::TextSelectableByMouse);
179
180          gridLayout_3->addWidget(fwLabel, 2, 2, 1, 1);
181
182          label_4 = new QLabel(devTab);
```

```
183         label_4->setObjectName(QStringLiteral("label_4"));
184         label_4->setMinimumSize(QSize(0, 34));
185 
186         gridLayout_3->addWidget(label_4, 0, 0, 1, 1);
187 
188         horizontalSpacer_2 = new QSpacerItem(40, 1, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
189 
190         gridLayout_3->addItem(horizontalSpacer_2, 2, 1, 1, 1);
191 
192         label_9 = new QLabel(devTab);
193         label_9->setObjectName(QStringLiteral("label_9"));
194         label_9->setMinimumSize(QSize(0, 34));
195 
196         gridLayout_3->addWidget(label_9, 2, 0, 1, 1);
197 
198         serialLabel = new QLabel(devTab);
199         serialLabel->setObjectName(QStringLiteral("serialLabel"));
200         serialLabel->setCursor(QCursor(Qt::IBeamCursor));
201         serialLabel->setTextInteractionFlags(Qt::LinksAccessibleByMouse|
    Qt::TextSelectableByMouse);
202 
203         gridLayout_3->addWidget(serialLabel, 0, 2, 1, 1);
204 
205         fwUpdLayout = new QHBoxLayout();
206         fwUpdLayout->setObjectName(QStringLiteral("fwUpdLayout"));
207         fwUpdButton = new QPushButton(devTab);
208         fwUpdButton->setObjectName(QStringLiteral("fwUpdButton"));
209 
210         fwUpdLayout->addWidget(fwUpdButton);
211 
212         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
213 
214         fwUpdLayout->addItem(horizontalSpacer_3);
215 
216 
217         gridLayout_3->addLayout(fwUpdLayout, 4, 2, 1, 1);
218 
219         pollLabel2 = new QLabel(devTab);
220         pollLabel2->setObjectName(QStringLiteral("pollLabel2"));
221         pollLabel2->setMinimumSize(QSize(0, 34));
222 
223         gridLayout_3->addWidget(pollLabel2, 1, 0, 1, 1);
224 
225         pollLabel = new QLabel(devTab);
226         pollLabel->setObjectName(QStringLiteral("pollLabel"));
227 
228         gridLayout_3->addWidget(pollLabel, 1, 2, 1, 1);
229 
230         tabWidget->addTab(devTab, QString());
231 
232         gridLayout->addWidget(tabWidget, 0, 1, 3, 1);
233 
234         modesList = new RListWidget(KbWidget);
235         modesList->setObjectName(QStringLiteral("modesList"));
236         QSizePolicy sizePolicy4(QSizePolicy::Expanding, QSizePolicy::Expanding);
237         sizePolicy4.setHorizontalStretch(18);
238         sizePolicy4.setVerticalStretch(0);
239         sizePolicy4.setHeightForWidth(modesList->sizePolicy().hasHeightForWidth());
240         modesList->setSizePolicy(sizePolicy4);
241         modesList->setContextMenuPolicy(Qt::CustomContextMenu);
242         modesList->setFrameShadow(QFrame::Plain);
243         modesList->setDragDropMode(QAbstractItemView::InternalMove);
244         modesList->setIconSize(QSize(24, 24));
245         modesList->setMovement(QListView::Snap);
246 
247         gridLayout->addWidget(modesList, 1, 0, 1, 1);
248 
249         hwSaveButton = new QPushButton(KbWidget);
250         hwSaveButton->setObjectName(QStringLiteral("hwSaveButton"));
251         QSizePolicy sizePolicy5(QSizePolicy::Minimum, QSizePolicy::Fixed);
252         sizePolicy5.setHorizontalStretch(18);
253         sizePolicy5.setVerticalStretch(0);
254         sizePolicy5.setHeightForWidth(hwSaveButton->sizePolicy().hasHeightForWidth());
255         hwSaveButton->setSizePolicy(sizePolicy5);
256 
257         gridLayout->addWidget(hwSaveButton, 2, 0, 1, 1);
258 
259 
260         retranslateUi(KbWidget);
261 
262         tabWidget->setCurrentIndex(0);
263 
264 
265         QMetaObject::connectSlotsByName(KbWidget);
266     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.95.3 Field Documentation

12.95.3.1 QWidget* Ui_KbWidget::bindTab

Definition at line 43 of file ui_kbwidget.h.

Referenced by retranslateUi(), and setupUi().

12.95.3.2 KbBindWidget* Ui_KbWidget::bindWidget

Definition at line 45 of file ui_kbwidget.h.

Referenced by KbWidget::modeChanged(), and setupUi().

12.95.3.3 QWidget* Ui_KbWidget::devTab

Definition at line 52 of file ui_kbwidget.h.

Referenced by retranslateUi(), and setupUi().

12.95.3.4 QLabel* Ui_KbWidget::fwLabel

Definition at line 56 of file ui_kbwidget.h.

Referenced by KbWidget::devUpdate(), retranslateUi(), and setupUi().

12.95.3.5 QPushButton* Ui_KbWidget::fwUpdButton

Definition at line 62 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), KbWidget::on_fwUpdButton_clicked(), retranslateUi(), setupUi(), and KbWidget::updateFwButton().

12.95.3.6 QLabel* Ui_KbWidget::fwUpdLabel

Definition at line 54 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), retranslateUi(), and setupUi().

12.95.3.7 QHBoxLayout* Ui_KbWidget::fwUpdLayout

Definition at line 61 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), and setupUi().

12.95.3.8 QGridLayout* Ui_KbWidget::gridLayout

Definition at line 36 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.9 QGridLayout* Ui_KbWidget::gridLayout_2

Definition at line 40 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.10 QGridLayout* Ui_KbWidget::gridLayout_3

Definition at line 53 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.11 QGridLayout* Ui_KbWidget::gridLayout_4

Definition at line 47 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.12 QGridLayout* Ui_KbWidget::gridLayout_5

Definition at line 44 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.13 QGridLayout* Ui_KbWidget::gridLayout_7

Definition at line 50 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.14 QSpacerItem* Ui_KbWidget::horizontalSpacer_2

Definition at line 58 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.15 QSpacerItem* Ui_KbWidget::horizontalSpacer_3

Definition at line 63 of file ui_kbwidget.h.

Referenced by setupUi().

12.95.3.16 QPushButton* Ui_KbWidget::hwSaveButton

Definition at line 67 of file ui_kbwidget.h.

Referenced by retranslateUi(), and setupUi().

12.95.3.17 QWidget* Ui_KbWidget::kPerfTab

Definition at line 46 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), retranslateUi(), and setupUi().

12.95.3.18 KPerfWidget* Ui_KbWidget::kPerfWidget

Definition at line 48 of file ui_kbwidget.h.

Referenced by KbWidget::modeChanged(), and setupUi().

12.95.3.19 QLabel* Ui_KbWidget::label_3

Definition at line 42 of file ui_kbwidget.h.

Referenced by retranslateUi(), and setupUi().

12.95.3.20 QLabel* Ui_KbWidget::label_4

Definition at line 57 of file ui_kbwidget.h.

Referenced by retranslateUi(), and setupUi().

12.95.3.21 QLabel* Ui_KbWidget::label_9

Definition at line 59 of file ui_kbwidget.h.

Referenced by retranslateUi(), and setupUi().

12.95.3.22 QWidget* Ui_KbWidget::lightTab

Definition at line 39 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), retranslateUi(), and setupUi().

12.95.3.23 KbLightWidget* Ui_KbWidget::lightWidget

Definition at line 41 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), KbWidget::modeChanged(), and setupUi().

12.95.3.24 RListWidget* Ui_KbWidget::modesList

Definition at line 66 of file ui_kbwidget.h.

Referenced by KbWidget::addNewModeItem(), KbWidget::KbWidget(), KbWidget::modeChanged(), KbWidget::modesList_reordered(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::on_modesList_itemClicked(), KbWidget::profileChanged(), and setupUi().

12.95.3.25 QWidget* Ui_KbWidget::mPerfTab

Definition at line 49 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), retranslateUi(), and setupUi().

12.95.3.26 MPerfWidget* Ui_KbWidget::mPerfWidget

Definition at line 51 of file ui_kbwidget.h.

Referenced by KbWidget::modeChanged(), and setupUi().

12.95.3.27 QLabel* Ui_KbWidget::pollLabel

Definition at line 65 of file ui_kbwidget.h.

Referenced by KbWidget::devUpdate(), KbWidget::KbWidget(), retranslateUi(), and setupUi().

12.95.3.28 QLabel* Ui_KbWidget::pollLabel2

Definition at line 64 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), retranslateUi(), and setupUi().

12.95.3.29 QComboBox* Ui_KbWidget::profileBox

Definition at line 37 of file ui_kbwidget.h.

Referenced by setupUi(), and KbWidget::updateProfileList().

12.95.3.30 QLabel* Ui_KbWidget::serialLabel

Definition at line 60 of file ui_kbwidget.h.

Referenced by KbWidget::devUpdate(), retranslateUi(), and setupUi().

12.95.3.31 QTabWidget* Ui_KbWidget::tabWidget

Definition at line 38 of file ui_kbwidget.h.

Referenced by KbWidget::KbWidget(), KbWidget::on_tabWidget_currentChanged(), retranslateUi(), setupUi(), KbWidget::showFirstTab(), and KbWidget::showLastTab().

12.95.3.32 QSpacerItem* Ui_KbWidget::verticalSpacer

Definition at line 55 of file ui_kbwidget.h.

Referenced by setupUi().

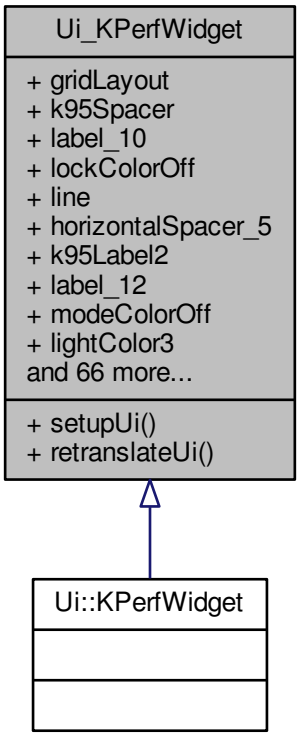
The documentation for this class was generated from the following file:

- src/ckb/ui_kbwidget.h

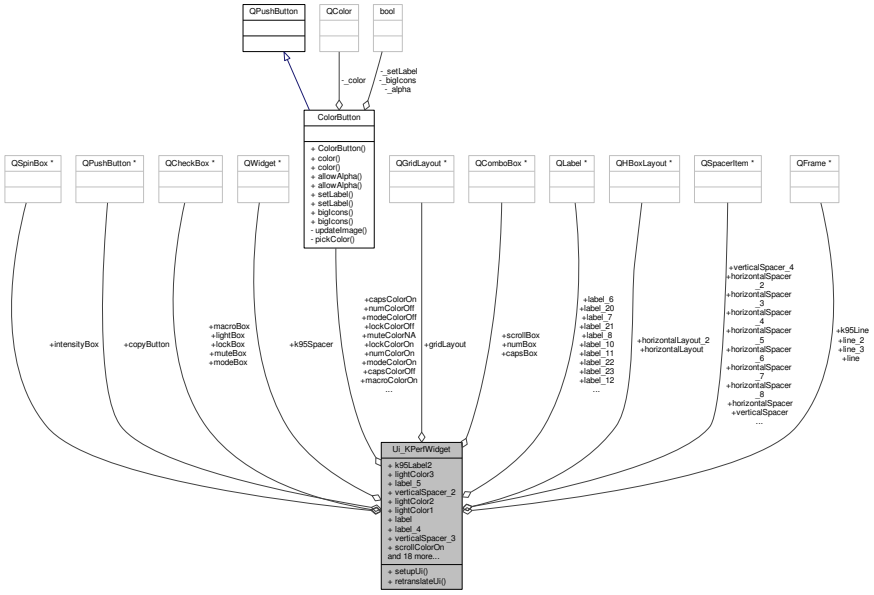
12.96 Ui_KPerfWidget Class Reference

`#include <src/ckb/ui_kperfwidget.h>`

Inheritance diagram for Ui_KPerfWidget:



Collaboration diagram for Ui_KPerfWidget:



Public Member Functions

- void setupUi (QWidget *KPerfWidget)
- void retranslateUi (QWidget *KPerfWidget)

Data Fields

- QGridLayout * gridLayout
- QWidget * k95Spacer
- QLabel * label_10
- ColorButton * lockColorOff
- QFrame * line
- QSpacerItem * horizontalSpacer_5
- QLabel * k95Label2
- QLabel * label_12
- ColorButton * modeColorOff
- ColorButton * lightColor3
- QSpacerItem * horizontalSpacer
- QCheckBox * lockBox
- ColorButton * modeColorOn
- QCheckBox * macroBox
- ColorButton * numColorOn
- QComboBox * numBox
- QLabel * label_5
- QSpacerItem * verticalSpacer_2
- ColorButton * lightColor2
- QSpacerItem * horizontalSpacer_7
- QLabel * label_8
- QCheckBox * lightBox
- ColorButton * lightColor1

- QSpacerItem * verticalSpacer_4
- QLabel * label_6
- QLabel * label
- QSpacerItem * horizontalSpacer_6
- ColorButton * lockColorOn
- QLabel * label_4
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_3
- QCheckBox * modeBox
- ColorButton * capsColorOff
- ColorButton * numColorOff
- ColorButton * capsColorOn
- QFrame * line_3
- QLabel * label_22
- QComboBox * scrollBox
- QLabel * label_7
- ColorButton * scrollColorOn
- QComboBox * capsBox
- ColorButton * scrollColorOff
- QCheckBox * muteBox
- ColorButton * muteColorOn
- ColorButton * macroColorOff
- QLabel * label_3
- ColorButton * muteColorOff
- QLabel * k95Label3
- ColorButton * macroColorOn
- ColorButton * muteColorNA
- QLabel * label_19
- QFrame * k95Line
- QLabel * label_21
- QLabel * label_24
- QLabel * label_20
- QLabel * label_16
- QLabel * label_14
- QLabel * label_18
- QSpacerItem * horizontalSpacer_2
- QLabel * label_15
- QLabel * k95Label4
- QLabel * label_17
- QLabel * label_11
- QLabel * label_23
- QLabel * k95Label1
- QHBoxLayout * horizontalLayout_2
- QPushButton * copyButton
- QSpacerItem * horizontalSpacer_8
- QHBoxLayout * horizontalLayout
- QLabel * label_2
- QSpacerItem * horizontalSpacer_4
- QSpinBox * intensityBox
- QSpacerItem * horizontalSpacer_3
- QLabel * k95Label5
- QFrame * line_2
- QLabel * k95Label6

12.96.1 Detailed Description

Definition at line 31 of file ui_kperfwidget.h.

12.96.2 Member Function Documentation

12.96.2.1 void Ui_KPerfWidget::retranslateUi (QWidget * *KPerfWidget*) [inline]

Definition at line 616 of file ui_kperfwidget.h.

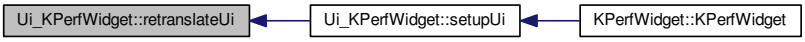
References capsBox, capsColorOff, capsColorOn, copyButton, intensityBox, k95Label1, k95Label2, k95Label3, k95Label4, k95Label5, k95Label6, label, label_10, label_11, label_12, label_14, label_15, label_16, label_17, label_18, label_19, label_2, label_20, label_21, label_22, label_23, label_24, label_3, label_4, label_5, label_6, label_7, label_8, lightBox, lightColor1, lightColor2, lightColor3, lockBox, lockColorOff, lockColorOn, macroBox, macroColorOff, macroColorOn, modeBox, modeColorOff, modeColorOn, muteBox, muteColorNA, muteColorOff, muteColorOn, numBox, numColorOff, numColorOn, scrollBox, scrollColorOff, and scrollColorOn.

Referenced by setupUi().

```
617     {
618         KPerfWidget->setWindowTitle(QApplication::translate("KPerfWidget", "Form", Q_NULLPTR));
619         label_10->setText(QApplication::translate("KPerfWidget", "100%", Q_NULLPTR));
620         lockColorOff->setText(QString());
621         k95Label2->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
622         label_12->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
623         modeColorOff->setText(QString());
624         lightColor3->setText(QString());
625         lockBox->setText(QApplication::translate("KPerfWidget", "Indicate Windows Lock:", Q_NULLPTR)
    );
626         modeColorOn->setText(QString());
627         macroBox->setText(QApplication::translate("KPerfWidget", "Indicate macro recording:",
    Q_NULLPTR));
628         numColorOn->setText(QString());
629         numBox->clear();
630         numBox->insertItems(0, QStringList()
631          << QApplication::translate("KPerfWidget", "Normal", Q_NULLPTR)
632          << QApplication::translate("KPerfWidget", "Always on", Q_NULLPTR)
633          << QApplication::translate("KPerfWidget", "Always off", Q_NULLPTR)
634          << QApplication::translate("KPerfWidget", "RGB indicator", Q_NULLPTR)
635          << QApplication::translate("KPerfWidget", "RGB + normal", Q_NULLPTR)
636         );
637         label_5->setText(QApplication::translate("KPerfWidget", "Num lock:", Q_NULLPTR));
638         lightColor2->setText(QString());
639         label_8->setText(QApplication::translate("KPerfWidget", "Note: many desktop environments do
     not use scroll lock. The indicator may not turn on even when enabled.", Q_NULLPTR));
640         lightBox->setText(QApplication::translate("KPerfWidget", "Indicate brightness:", Q_NULLPTR)
    );
641         lightColor1->setText(QString());
642         label_6->setText(QApplication::translate("KPerfWidget", "Caps lock:", Q_NULLPTR));
643         label->setText(QApplication::translate("KPerfWidget", "Miscellaneous", Q_NULLPTR));
644         lockColorOn->setText(QString());
645         label_4->setText(QApplication::translate("KPerfWidget", "Lock lights", Q_NULLPTR));
646         modeBox->setText(QApplication::translate("KPerfWidget", "Indicate current mode:", Q_NULLPTR)
    );
647         capsColorOff->setText(QString());
648         numColorOff->setText(QString());
649         capsColorOn->setText(QString());
650         label_22->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
651         scrollBox->clear();
652         scrollBox->insertItems(0, QStringList()
653          << QApplication::translate("KPerfWidget", "Normal", Q_NULLPTR)
654          << QApplication::translate("KPerfWidget", "Always on", Q_NULLPTR)
655          << QApplication::translate("KPerfWidget", "Always off", Q_NULLPTR)
656          << QApplication::translate("KPerfWidget", "RGB indicator", Q_NULLPTR)
657          << QApplication::translate("KPerfWidget", "RGB + normal", Q_NULLPTR)
658         );
659         label_7->setText(QApplication::translate("KPerfWidget", "Scroll lock:", Q_NULLPTR));
660         scrollColorOn->setText(QString());
661         capsBox->clear();
662         capsBox->insertItems(0, QStringList()
663          << QApplication::translate("KPerfWidget", "Normal", Q_NULLPTR)
664          << QApplication::translate("KPerfWidget", "Always on", Q_NULLPTR)
665          << QApplication::translate("KPerfWidget", "Always off", Q_NULLPTR)
666          << QApplication::translate("KPerfWidget", "RGB indicator", Q_NULLPTR)
667          << QApplication::translate("KPerfWidget", "RGB + normal", Q_NULLPTR)
668         );
669         scrollColorOff->setText(QString());
670         muteBox->setText(QApplication::translate("KPerfWidget", "Indicate mute:", Q_NULLPTR));
```

```
671         muteColorOn->setText(QString());
672         macroColorOff->setText(QString());
673         label_3->setText(QApplication::translate("KPerfWidget", "Top row", Q_NULLPTR));
674         muteColorOff->setText(QString());
675         k95Label3->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
676         macroColorOn->setText(QString());
677 #ifndef QT_NO_TOOLTIP
678         muteColorNA->setToolTip(QApplication::translate("KPerfWidget", "If the sound device does
    not support muting, or if mute status is unknown, this color will be displayed.", Q_NULLPTR));
679 #endif // QT_NO_TOOLTIP
680         muteColorNA->setText(QString());
681         label_19->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
682         label_21->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
683         label_24->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
684         label_20->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
685         label_16->setText(QApplication::translate("KPerfWidget", "67%", Q_NULLPTR));
686         label_14->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
687         label_18->setText(QApplication::translate("KPerfWidget", "33%", Q_NULLPTR));
688         label_15->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
689         k95Label4->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
690         label_17->setText(QApplication::translate("KPerfWidget", "Unknown", Q_NULLPTR));
691         label_11->setText(QApplication::translate("KPerfWidget", "On", Q_NULLPTR));
692         label_23->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
693         k95Label1->setText(QApplication::translate("KPerfWidget", "K95 buttons", Q_NULLPTR));
694         copyButton->setText(QApplication::translate("KPerfWidget", "Copy to mode...", Q_NULLPTR))
    ;
695         label_2->setText(QApplication::translate("KPerfWidget", "Indicator intensity:", Q_NULLPTR));
696         intensityBox->setSuffix(QApplication::translate("KPerfWidget", "%", Q_NULLPTR));
697         k95Label5->setText(QApplication::translate("KPerfWidget", "Off", Q_NULLPTR));
698         k95Label6->setText(QApplication::translate("KPerfWidget", "Note: macro recording not yet
    implemented. Indicator will always display \"off\".", Q_NULLPTR));
699     } // retranslateUi
```

Here is the caller graph for this function:



12.96.2.2 void Ui_KPerfWidget::setupUi (QWidget * *KPerfWidget*) [inline]

Definition at line 111 of file ui_kperfwidget.h.

References capsBox, capsColorOff, capsColorOn, copyButton, gridLayout, horizontalLayout, horizontalLayout_2, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, horizontalSpacer_4, horizontalSpacer_5, horizontalSpacer_6, horizontalSpacer_7, horizontalSpacer_8, intensityBox, k95Label1, k95Label2, k95Label3, k95Label4, k95Label5, k95Label6, k95Line, k95Spacer, label, label_10, label_11, label_12, label_14, label_15, label_16, label_17, label_18, label_19, label_2, label_20, label_21, label_22, label_23, label_24, label_3, label_4, label_5, label_6, label_7, label_8, lightBox, lightColor1, lightColor2, lightColor3, line, line_2, line_3, lockBox, lockColorOff, lockColorOn, macroBox, macroColorOff, macroColorOn, modeBox, modeColorOff, modeColorOn, muteBox, muteColorNA, muteColorOff, muteColorOn, numBox, numColorOff, numColorOn, retranslateUi(), scrollBox, scrollColorOff, scrollColorOn, verticalSpacer, verticalSpacer_2, verticalSpacer_3, and verticalSpacer_4.

Referenced by KPerfWidget::KPerfWidget().

```
112     {
113         if (KPerfWidget->objectName().isEmpty())
114             KPerfWidget->setObjectName(QStringLiteral("KPerfWidget"));
115         KPerfWidget->resize(840, 679);
116         gridLayout = new QGridLayout(KPerfWidget);
117         gridLayout->setObjectName(QStringLiteral("gridLayout"));
118         k95Spacer = new QWidget(KPerfWidget);
119         k95Spacer->setObjectName(QStringLiteral("k95Spacer"));
120         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
121         sizePolicy.setHorizontalStretch(0);
122         sizePolicy.setVerticalStretch(0);
123         sizePolicy.setHeightForWidth(k95Spacer->sizePolicy().hasHeightForWidth());
124         k95Spacer->setSizePolicy(sizePolicy);
125         k95Spacer->setMinimumSize(QSize(15, 10));
```

```
126         k95Spacer->setMaximumSize(QSize(15, 10));
127 
128         gridLayout->addWidget(k95Spacer, 12, 4, 1, 1);
129 
130         label_10 = new QLabel(KPerfWidget);
131         label_10->setObjectName(QStringLiteral("label_10"));
132 
133         gridLayout->addWidget(label_10, 15, 5, 1, 1);
134 
135         lockColorOff = new ColorButton(KPerfWidget);
136         lockColorOff->setObjectName(QStringLiteral("lockColorOff"));
137         sizePolicy.setHeightForWidth(lockColorOff->sizePolicy().hasHeightForWidth());
138         lockColorOff->setSizePolicy(sizePolicy);
139         lockColorOff->setMinimumSize(QSize(0, 40));
140         lockColorOff->setMaximumSize(QSize(16777215, 40));
141 
142         gridLayout->addWidget(lockColorOff, 16, 7, 1, 1);
143 
144         line = new QFrame(KPerfWidget);
145         line->setObjectName(QStringLiteral("line"));
146         line->setFrameShape(QFrame::HLine);
147         line->setFrameShadow(QFrame::Sunken);
148 
149         gridLayout->addWidget(line, 2, 0, 1, 13);
150 
151         horizontalSpacer_5 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
152 
153         gridLayout->addItem(horizontalSpacer_5, 17, 6, 1, 1);
154 
155         k95Label2 = new QLabel(KPerfWidget);
156         k95Label2->setObjectName(QStringLiteral("k95Label2"));
157 
158         gridLayout->addWidget(k95Label2, 9, 5, 1, 1);
159 
160         label_12 = new QLabel(KPerfWidget);
161         label_12->setObjectName(QStringLiteral("label_12"));
162 
163         gridLayout->addWidget(label_12, 17, 5, 1, 1);
164 
165         modeColorOff = new ColorButton(KPerfWidget);
166         modeColorOff->setObjectName(QStringLiteral("modeColorOff"));
167         sizePolicy.setHeightForWidth(modeColorOff->sizePolicy().hasHeightForWidth());
168         modeColorOff->setSizePolicy(sizePolicy);
169         modeColorOff->setMinimumSize(QSize(0, 40));
170         modeColorOff->setMaximumSize(QSize(16777215, 40));
171 
172         gridLayout->addWidget(modeColorOff, 9, 7, 1, 1);
173 
174         lightColor3 = new ColorButton(KPerfWidget);
175         lightColor3->setObjectName(QStringLiteral("lightColor3"));
176         sizePolicy.setHeightForWidth(lightColor3->sizePolicy().hasHeightForWidth());
177         lightColor3->setSizePolicy(sizePolicy);
178         lightColor3->setMinimumSize(QSize(0, 40));
179         lightColor3->setMaximumSize(QSize(16777215, 40));
180 
181         gridLayout->addWidget(lightColor3, 15, 4, 1, 1);
182 
183         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
184 
185         gridLayout->addItem(horizontalSpacer, 15, 12, 1, 1);
186 
187         lockBox = new QCheckBox(KPerfWidget);
188         lockBox->setObjectName(QStringLiteral("lockBox"));
189 
190         gridLayout->addWidget(lockBox, 16, 1, 1, 2);
191 
192         modeColorOn = new ColorButton(KPerfWidget);
193         modeColorOn->setObjectName(QStringLiteral("modeColorOn"));
194         sizePolicy.setHeightForWidth(modeColorOn->sizePolicy().hasHeightForWidth());
195         modeColorOn->setSizePolicy(sizePolicy);
196         modeColorOn->setMinimumSize(QSize(0, 40));
197         modeColorOn->setMaximumSize(QSize(16777215, 40));
198 
199         gridLayout->addWidget(modeColorOn, 9, 4, 1, 1);
200 
201         macroBox = new QCheckBox(KPerfWidget);
202         macroBox->setObjectName(QStringLiteral("macroBox"));
203 
204         gridLayout->addWidget(macroBox, 10, 1, 1, 2);
205 
206         numColorOn = new ColorButton(KPerfWidget);
207         numColorOn->setObjectName(QStringLiteral("numColorOn"));
208         sizePolicy.setHeightForWidth(numColorOn->sizePolicy().hasHeightForWidth());
209         numColorOn->setSizePolicy(sizePolicy);
210         numColorOn->setMinimumSize(QSize(0, 40));
```

```
211         numColorOn->setMaximumSize(QSize(16777215, 40));
212
213         gridLayout->addWidget(numColorOn, 21, 4, 1, 1);
214
215         numBox = new QComboBox(KPerfWidget);
216         numBox->setObjectName(QStringLiteral("numBox"));
217
218         gridLayout->addWidget(numBox, 21, 2, 1, 1);
219
220         label_5 = new QLabel(KPerfWidget);
221         label_5->setObjectName(QStringLiteral("label_5"));
222
223         gridLayout->addWidget(label_5, 21, 1, 1, 1);
224
225         verticalSpacer_2 = new QSpacerItem(20, 20, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
226
227         gridLayout->addItem(verticalSpacer_2, 4, 1, 1, 1);
228
229         lightColor2 = new ColorButton(KPerfWidget);
230         lightColor2->setObjectName(QStringLiteral("lightColor2"));
231         sizePolicy.setHeightForWidth(lightColor2->sizePolicy().hasHeightForWidth());
232         lightColor2->setSizePolicy(sizePolicy);
233         lightColor2->setMinimumSize(QSize(0, 40));
234         lightColor2->setMaximumSize(QSize(16777215, 40));
235
236         gridLayout->addWidget(lightColor2, 15, 7, 1, 1);
237
238         horizontalSpacer_7 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
239
240         gridLayout->addItem(horizontalSpacer_7, 17, 3, 1, 1);
241
242         label_8 = new QLabel(KPerfWidget);
243         label_8->setObjectName(QStringLiteral("label_8"));
244         QFont font;
245         font.setPointSize(8);
246         font.setItalic(true);
247         label_8->setFont(font);
248
249         gridLayout->addWidget(label_8, 25, 1, 1, 12);
250
251         lightBox = new QCheckBox(KPerfWidget);
252         lightBox->setObjectName(QStringLiteral("lightBox"));
253
254         gridLayout->addWidget(lightBox, 15, 1, 1, 2);
255
256         lightColor1 = new ColorButton(KPerfWidget);
257         lightColor1->setObjectName(QStringLiteral("lightColor1"));
258         sizePolicy.setHeightForWidth(lightColor1->sizePolicy().hasHeightForWidth());
259         lightColor1->setSizePolicy(sizePolicy);
260         lightColor1->setMinimumSize(QSize(0, 40));
261         lightColor1->setMaximumSize(QSize(16777215, 40));
262
263         gridLayout->addWidget(lightColor1, 15, 10, 1, 1);
264
265         verticalSpacer_4 = new QSpacerItem(0, 5, QSizePolicy::Minimum, QSizePolicy::Fixed);
266
267         gridLayout->addItem(verticalSpacer_4, 24, 4, 1, 1);
268
269         label_6 = new QLabel(KPerfWidget);
270         label_6->setObjectName(QStringLiteral("label_6"));
271
272         gridLayout->addWidget(label_6, 22, 1, 1, 1);
273
274         label = new QLabel(KPerfWidget);
275         label->setObjectName(QStringLiteral("label"));
276         QFont font1;
277         font1.setBold(true);
278         font1.setWeight(75);
279         label->setFont(font1);
280
281         gridLayout->addWidget(label, 1, 0, 1, 13);
282
283         horizontalSpacer_6 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
284
285         gridLayout->addItem(horizontalSpacer_6, 17, 9, 1, 1);
286
287         lockColorOn = new ColorButton(KPerfWidget);
288         lockColorOn->setObjectName(QStringLiteral("lockColorOn"));
289         sizePolicy.setHeightForWidth(lockColorOn->sizePolicy().hasHeightForWidth());
290         lockColorOn->setSizePolicy(sizePolicy);
291         lockColorOn->setMinimumSize(QSize(0, 40));
292         lockColorOn->setMaximumSize(QSize(16777215, 40));
293
294         gridLayout->addWidget(lockColorOn, 16, 4, 1, 1);
```

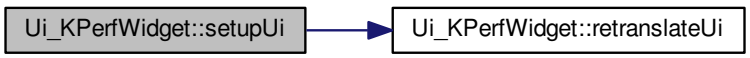
```
295
296         label_4 = new QLabel(KPerfWidget);
297         label_4->setObjectName(QStringLiteral("label_4"));
298         label_4->setFont(font1);
299
300         gridLayout->addWidget(label_4, 19, 0, 1, 13);
301
302         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
303
304         gridLayout->addItem(verticalSpacer, 26, 2, 1, 1);
305
306         verticalSpacer_3 = new QSpacerItem(20, 20, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
307
308         gridLayout->addItem(verticalSpacer_3, 18, 4, 1, 1);
309
310         modeBox = new QCheckBox(KPerfWidget);
311         modeBox->setObjectName(QStringLiteral("modeBox"));
312
313         gridLayout->addWidget(modeBox, 9, 1, 1, 2);
314
315         capsColorOff = new ColorButton(KPerfWidget);
316         capsColorOff->setObjectName(QStringLiteral("capsColorOff"));
317         sizePolicy.setHeightForWidth(capsColorOff->sizePolicy().hasHeightForWidth());
318         capsColorOff->setSizePolicy(sizePolicy);
319         capsColorOff->setMinimumSize(QSize(0, 40));
320         capsColorOff->setMaximumSize(QSize(16777215, 40));
321
322         gridLayout->addWidget(capsColorOff, 22, 7, 1, 1);
323
324         numColorOff = new ColorButton(KPerfWidget);
325         numColorOff->setObjectName(QStringLiteral("numColorOff"));
326         QSizePolicy sizePolicy1(QSizePolicy::Fixed, QSizePolicy::Fixed);
327         sizePolicy1.setHorizontalStretch(0);
328         sizePolicy1.setVerticalStretch(40);
329         sizePolicy1.setHeightForWidth(numColorOff->sizePolicy().hasHeightForWidth());
330         numColorOff->setSizePolicy(sizePolicy1);
331         numColorOff->setMaximumSize(QSize(16777215, 40));
332
333         gridLayout->addWidget(numColorOff, 21, 7, 1, 1);
334
335         capsColorOn = new ColorButton(KPerfWidget);
336         capsColorOn->setObjectName(QStringLiteral("capsColorOn"));
337         sizePolicy.setHeightForWidth(capsColorOn->sizePolicy().hasHeightForWidth());
338         capsColorOn->setSizePolicy(sizePolicy);
339         capsColorOn->setMinimumSize(QSize(0, 40));
340         capsColorOn->setMaximumSize(QSize(16777215, 40));
341
342         gridLayout->addWidget(capsColorOn, 22, 4, 1, 1);
343
344         line_3 = new QFrame(KPerfWidget);
345         line_3->setObjectName(QStringLiteral("line_3"));
346         line_3->setFrameShape(QFrame::HLine);
347         line_3->setFrameShadow(QFrame::Sunken);
348
349         gridLayout->addWidget(line_3, 20, 0, 1, 13);
350
351         label_22 = new QLabel(KPerfWidget);
352         label_22->setObjectName(QStringLiteral("label_22"));
353
354         gridLayout->addWidget(label_22, 21, 8, 1, 1);
355
356         scrollBox = new QComboBox(KPerfWidget);
357         scrollBox->setObjectName(QStringLiteral("scrollBox"));
358
359         gridLayout->addWidget(scrollBox, 23, 2, 1, 1);
360
361         label_7 = new QLabel(KPerfWidget);
362         label_7->setObjectName(QStringLiteral("label_7"));
363
364         gridLayout->addWidget(label_7, 23, 1, 1, 1);
365
366         scrollColorOn = new ColorButton(KPerfWidget);
367         scrollColorOn->setObjectName(QStringLiteral("scrollColorOn"));
368         sizePolicy.setHeightForWidth(scrollColorOn->sizePolicy().hasHeightForWidth());
369         scrollColorOn->setSizePolicy(sizePolicy);
370         scrollColorOn->setMinimumSize(QSize(0, 40));
371         scrollColorOn->setMaximumSize(QSize(16777215, 40));
372
373         gridLayout->addWidget(scrollColorOn, 23, 4, 1, 1);
374
375         capsBox = new QComboBox(KPerfWidget);
376         capsBox->setObjectName(QStringLiteral("capsBox"));
377
378         gridLayout->addWidget(capsBox, 22, 2, 1, 1);
379
```

```
380         scrollColorOff = new ColorButton(KPerfWidget);
381         scrollColorOff->setObjectName(QStringLiteral("scrollColorOff"));
382         sizePolicy.setHeightForWidth(scrollColorOff->sizePolicy().hasHeightForWidth());
383         scrollColorOff->setSizePolicy(sizePolicy);
384         scrollColorOff->setMinimumSize(QSize(0, 40));
385         scrollColorOff->setMaximumSize(QSize(16777215, 40));
386 
387         gridLayout->addWidget(scrollColorOff, 23, 7, 1, 1);
388 
389         muteBox = new QCheckBox(KPerfWidget);
390         muteBox->setObjectName(QStringLiteral("muteBox"));
391 
392         gridLayout->addWidget(muteBox, 17, 1, 1, 2);
393 
394         muteColorOn = new ColorButton(KPerfWidget);
395         muteColorOn->setObjectName(QStringLiteral("muteColorOn"));
396         sizePolicy.setHeightForWidth(muteColorOn->sizePolicy().hasHeightForWidth());
397         muteColorOn->setSizePolicy(sizePolicy);
398         muteColorOn->setMinimumSize(QSize(0, 40));
399         muteColorOn->setMaximumSize(QSize(16777215, 40));
400 
401         gridLayout->addWidget(muteColorOn, 17, 4, 1, 1);
402 
403         macroColorOff = new ColorButton(KPerfWidget);
404         macroColorOff->setObjectName(QStringLiteral("macroColorOff"));
405         QSizePolicy sizePolicy2(QSizePolicy::Minimum, QSizePolicy::Fixed);
406         sizePolicy2.setHorizontalStretch(0);
407         sizePolicy2.setVerticalStretch(40);
408         sizePolicy2.setHeightForWidth(macroColorOff->sizePolicy().hasHeightForWidth());
409         macroColorOff->setSizePolicy(sizePolicy2);
410         macroColorOff->setMinimumSize(QSize(0, 40));
411 
412         gridLayout->addWidget(macroColorOff, 10, 7, 1, 1);
413 
414         label_3 = new QLabel(KPerfWidget);
415         label_3->setObjectName(QStringLiteral("label_3"));
416         label_3->setFont(font1);
417 
418         gridLayout->addWidget(label_3, 13, 0, 1, 3);
419 
420         muteColorOff = new ColorButton(KPerfWidget);
421         muteColorOff->setObjectName(QStringLiteral("muteColorOff"));
422         sizePolicy.setHeightForWidth(muteColorOff->sizePolicy().hasHeightForWidth());
423         muteColorOff->setSizePolicy(sizePolicy);
424         muteColorOff->setMinimumSize(QSize(0, 40));
425         muteColorOff->setMaximumSize(QSize(16777215, 40));
426 
427         gridLayout->addWidget(muteColorOff, 17, 7, 1, 1);
428 
429         k95Label3 = new QLabel(KPerfWidget);
430         k95Label3->setObjectName(QStringLiteral("k95Label3"));
431 
432         gridLayout->addWidget(k95Label3, 9, 8, 1, 1);
433 
434         macroColorOn = new ColorButton(KPerfWidget);
435         macroColorOn->setObjectName(QStringLiteral("macroColorOn"));
436         sizePolicy2.setHeightForWidth(macroColorOn->sizePolicy().hasHeightForWidth());
437         macroColorOn->setSizePolicy(sizePolicy2);
438         macroColorOn->setMinimumSize(QSize(0, 40));
439 
440         gridLayout->addWidget(macroColorOn, 10, 4, 1, 1);
441 
442         muteColorNA = new ColorButton(KPerfWidget);
443         muteColorNA->setObjectName(QStringLiteral("muteColorNA"));
444         muteColorNA->setMinimumSize(QSize(0, 40));
445         muteColorNA->setMaximumSize(QSize(16777215, 40));
446 
447         gridLayout->addWidget(muteColorNA, 17, 10, 1, 1);
448 
449         label_19 = new QLabel(KPerfWidget);
450         label_19->setObjectName(QStringLiteral("label_19"));
451 
452         gridLayout->addWidget(label_19, 21, 5, 1, 1);
453 
454         k95Line = new QFrame(KPerfWidget);
455         k95Line->setObjectName(QStringLiteral("k95Line"));
456         k95Line->setFrameShape(QFrame::HLine);
457         k95Line->setFrameShadow(QFrame::Sunken);
458 
459         gridLayout->addWidget(k95Line, 8, 0, 1, 13);
460 
461         label_21 = new QLabel(KPerfWidget);
462         label_21->setObjectName(QStringLiteral("label_21"));
463 
464         gridLayout->addWidget(label_21, 23, 5, 1, 1);
465 
466         label_24 = new QLabel(KPerfWidget);
```

```
467         label_24->setObjectName(QStringLiteral("label_24"));
468
469         gridLayout->addWidget(label_24, 23, 8, 1, 1);
470
471         label_20 = new QLabel(KPerfWidget);
472         label_20->setObjectName(QStringLiteral("label_20"));
473
474         gridLayout->addWidget(label_20, 22, 5, 1, 1);
475
476         label_16 = new QLabel(KPerfWidget);
477         label_16->setObjectName(QStringLiteral("label_16"));
478
479         gridLayout->addWidget(label_16, 15, 8, 1, 1);
480
481         label_14 = new QLabel(KPerfWidget);
482         label_14->setObjectName(QStringLiteral("label_14"));
483
484         gridLayout->addWidget(label_14, 17, 8, 1, 1);
485
486         label_18 = new QLabel(KPerfWidget);
487         label_18->setObjectName(QStringLiteral("label_18"));
488
489         gridLayout->addWidget(label_18, 15, 11, 1, 1);
490
491         horizontalSpacer_2 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
492
493         gridLayout->addItem(horizontalSpacer_2, 15, 0, 1, 1);
494
495         label_15 = new QLabel(KPerfWidget);
496         label_15->setObjectName(QStringLiteral("label_15"));
497
498         gridLayout->addWidget(label_15, 16, 8, 1, 1);
499
500         k95Label4 = new QLabel(KPerfWidget);
501         k95Label4->setObjectName(QStringLiteral("k95Label4"));
502
503         gridLayout->addWidget(k95Label4, 10, 5, 1, 1);
504
505         label_17 = new QLabel(KPerfWidget);
506         label_17->setObjectName(QStringLiteral("label_17"));
507
508         gridLayout->addWidget(label_17, 17, 11, 1, 1);
509
510         label_11 = new QLabel(KPerfWidget);
511         label_11->setObjectName(QStringLiteral("label_11"));
512
513         gridLayout->addWidget(label_11, 16, 5, 1, 1);
514
515         label_23 = new QLabel(KPerfWidget);
516         label_23->setObjectName(QStringLiteral("label_23"));
517
518         gridLayout->addWidget(label_23, 22, 8, 1, 1);
519
520         k95Label1 = new QLabel(KPerfWidget);
521         k95Label1->setObjectName(QStringLiteral("k95Label1"));
522         k95Label1->setFont(font1);
523
524         gridLayout->addWidget(k95Label1, 7, 0, 1, 5);
525
526         horizontalLayout_2 = new QHBoxLayout();
527         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
528         copyButton = new QPushButton(KPerfWidget);
529         copyButton->setObjectName(QStringLiteral("copyButton"));
530
531         horizontalLayout_2->addWidget(copyButton);
532
533         horizontalSpacer_8 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
534
535         horizontalLayout_2->addItem(horizontalSpacer_8);
536
537
538         gridLayout->addLayout(horizontalLayout_2, 3, 10, 1, 3);
539
540         horizontalLayout = new QHBoxLayout();
541         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
542         label_2 = new QLabel(KPerfWidget);
543         label_2->setObjectName(QStringLiteral("label_2"));
544
545         horizontalLayout->addWidget(label_2);
546
547         horizontalSpacer_4 = new QSpacerItem(10, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
548
549         horizontalLayout->addItem(horizontalSpacer_4);
550
```

```
551         intensityBox = new QSpinBox(KPerfWidget);
552         intensityBox->setObjectName(QStringLiteral("intensityBox"));
553         intensityBox->setMaximum(100);
554         intensityBox->setValue(100);
555 
556         horizontalLayout->addWidget(intensityBox);
557 
558         horizontalSpacer_3 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
559 
560         horizontalLayout->addItem(horizontalSpacer_3);
561 
562 
563         gridLayout->addLayout(horizontalLayout, 3, 1, 1, 9);
564 
565         k95Label5 = new QLabel(KPerfWidget);
566         k95Label5->setObjectName(QStringLiteral("k95Label5"));
567 
568         gridLayout->addWidget(k95Label5, 10, 8, 1, 1);
569 
570         line_2 = new QFrame(KPerfWidget);
571         line_2->setObjectName(QStringLiteral("line_2"));
572         line_2->setFrameShape(QFrame::HLine);
573         line_2->setFrameShadow(QFrame::Sunken);
574 
575         gridLayout->addWidget(line_2, 14, 0, 1, 13);
576 
577         k95Label6 = new QLabel(KPerfWidget);
578         k95Label6->setObjectName(QStringLiteral("k95Label6"));
579         k95Label6->setFont(font);
580 
581         gridLayout->addWidget(k95Label6, 11, 1, 1, 10);
582 
583         QWidget::setTabOrder(intensityBox, copyButton);
584         QWidget::setTabOrder(copyButton, modeBox);
585         QWidget::setTabOrder(modeBox, modeColorOn);
586         QWidget::setTabOrder(modeColorOn, modeColorOff);
587         QWidget::setTabOrder(modeColorOff, macroBox);
588         QWidget::setTabOrder(macroBox, macroColorOn);
589         QWidget::setTabOrder(macroColorOn, macroColorOff);
590         QWidget::setTabOrder(macroColorOff, lightBox);
591         QWidget::setTabOrder(lightBox, lightColor3);
592         QWidget::setTabOrder(lightColor3, lightColor2);
593         QWidget::setTabOrder(lightColor2, lightColor1);
594         QWidget::setTabOrder(lightColor1, lockBox);
595         QWidget::setTabOrder(lockBox, lockColorOn);
596         QWidget::setTabOrder(lockColorOn, lockColorOff);
597         QWidget::setTabOrder(lockColorOff, muteBox);
598         QWidget::setTabOrder(muteBox, muteColorOn);
599         QWidget::setTabOrder(muteColorOn, muteColorOff);
600         QWidget::setTabOrder(muteColorOff, muteColorNA);
601         QWidget::setTabOrder(muteColorNA, numBox);
602         QWidget::setTabOrder(numBox, numColorOn);
603         QWidget::setTabOrder(numColorOn, numColorOff);
604         QWidget::setTabOrder(numColorOff, capsBox);
605         QWidget::setTabOrder(capsBox, capsColorOn);
606         QWidget::setTabOrder(capsColorOn, capsColorOff);
607         QWidget::setTabOrder(capsColorOff, scrollBox);
608         QWidget::setTabOrder(scrollBox, scrollColorOn);
609         QWidget::setTabOrder(scrollColorOn, scrollColorOff);
610 
611         retranslateUi(KPerfWidget);
612 
613         QMetaObject::connectSlotsByName(KPerfWidget);
614     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.96.3 Field Documentation

12.96.3.1 QComboBox* Ui_KPerfWidget::capsBox

Definition at line 74 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.2 ColorButton* Ui_KPerfWidget::capsColorOff

Definition at line 66 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.3 ColorButton* Ui_KPerfWidget::capsColorOn

Definition at line 68 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.4 QPushButton* Ui_KPerfWidget::copyButton

Definition at line 100 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.5 QGridLayout* Ui_KPerfWidget::gridLayout

Definition at line 34 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.6 QHBoxLayout* Ui_KPerfWidget::horizontalLayout

Definition at line 102 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.7 QHBoxLayout* Ui_KPerfWidget::horizontalLayout_2

Definition at line 99 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.8 QSpacerItem* Ui_KPerfWidget::horizontalSpacer

Definition at line 44 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.9 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_2

Definition at line 92 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.10 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_3

Definition at line 106 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.11 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_4

Definition at line 104 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.12 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_5

Definition at line 39 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.13 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_6

Definition at line 60 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.14 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_7

Definition at line 53 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.15 QSpacerItem* Ui_KPerfWidget::horizontalSpacer_8

Definition at line 101 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.16 QSpinBox* Ui_KPerfWidget::intensityBox

Definition at line 105 of file ui_kperfwidget.h.

Referenced by retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.17 QLabel* Ui_KPerfWidget::k95Label1

Definition at line 98 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.18 QLabel* Ui_KPerfWidget::k95Label2

Definition at line 40 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.19 QLabel* Ui_KPerfWidget::k95Label3

Definition at line 81 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.20 QLabel* Ui_KPerfWidget::k95Label4

Definition at line 94 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.21 QLabel* Ui_KPerfWidget::k95Label5

Definition at line 107 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.22 QLabel* Ui_KPerfWidget::k95Label6

Definition at line 109 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.23 QFrame* Ui_KPerfWidget::k95Line

Definition at line 85 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), and setupUi().

12.96.3.24 QWidget* Ui_KPerfWidget::k95Spacer

Definition at line 35 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), and setupUi().

12.96.3.25 QLabel* Ui_KPerfWidget::label

Definition at line 59 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.26 QLabel* Ui_KPerfWidget::label_10

Definition at line 36 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.27 QLabel* Ui_KPerfWidget::label_11

Definition at line 96 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.28 QLabel* Ui_KPerfWidget::label_12

Definition at line 41 of file ui_kperfwidget.h.

Referenced by retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.29 QLabel* Ui_KPerfWidget::label_14

Definition at line 90 of file ui_kperfwidget.h.

Referenced by retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.30 QLabel* Ui_KPerfWidget::label_15

Definition at line 93 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.31 QLabel* Ui_KPerfWidget::label_16

Definition at line 89 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.32 QLabel* Ui_KPerfWidget::label_17

Definition at line 95 of file ui_kperfwidget.h.

Referenced by retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.33 QLabel* Ui_KPerfWidget::label_18

Definition at line 91 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.34 QLabel* Ui_KPerfWidget::label_19

Definition at line 84 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.35 QLabel* Ui_KPerfWidget::label_2

Definition at line 103 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.36 QLabel* Ui_KPerfWidget::label_20

Definition at line 88 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.37 QLabel* Ui_KPerfWidget::label_21

Definition at line 86 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.38 QLabel* Ui_KPerfWidget::label_22

Definition at line 70 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.39 QLabel* Ui_KPerfWidget::label_23

Definition at line 97 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.40 QLabel* Ui_KPerfWidget::label_24

Definition at line 87 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.41 QLabel* Ui_KPerfWidget::label_3

Definition at line 79 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.42 QLabel* Ui_KPerfWidget::label_4

Definition at line 62 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.43 QLabel* Ui_KPerfWidget::label_5

Definition at line 50 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.44 QLabel* Ui_KPerfWidget::label_6

Definition at line 58 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.45 QLabel* Ui_KPerfWidget::label_7

Definition at line 72 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.46 QLabel* Ui_KPerfWidget::label_8

Definition at line 54 of file ui_kperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.96.3.47 QCheckBox* Ui_KPerfWidget::lightBox

Definition at line 55 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.48 ColorButton* Ui_KPerfWidget::lightColor1

Definition at line 56 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.49 ColorButton* Ui_KPerfWidget::lightColor2

Definition at line 52 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.50 ColorButton* Ui_KPerfWidget::lightColor3

Definition at line 43 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.51 QFrame* Ui_KPerfWidget::line

Definition at line 38 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.52 QFrame* Ui_KPerfWidget::line_2

Definition at line 108 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.53 QFrame* Ui_KPerfWidget::line_3

Definition at line 69 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.54 QCheckBox* Ui_KPerfWidget::lockBox

Definition at line 45 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.55 ColorButton* Ui_KPerfWidget::lockColorOff

Definition at line 37 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.56 ColorButton* Ui_KPerfWidget::lockColorOn

Definition at line 61 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.57 QCheckBox* Ui_KPerfWidget::macroBox

Definition at line 47 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.58 ColorButton* Ui_KPerfWidget::macroColorOff

Definition at line 78 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.59 ColorButton* Ui_KPerfWidget::macroColorOn

Definition at line 82 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.60 QCheckBox* Ui_KPerfWidget::modeBox

Definition at line 65 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.61 ColorButton* Ui_KPerfWidget::modeColorOff

Definition at line 42 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.62 ColorButton* Ui_KPerfWidget::modeColorOn

Definition at line 46 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.63 QCheckBox* Ui_KPerfWidget::muteBox

Definition at line 76 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.64 ColorButton* Ui_KPerfWidget::muteColorNA

Definition at line 83 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.65 ColorButton* Ui_KPerfWidget::muteColorOff

Definition at line 80 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.66 ColorButton* Ui_KPerfWidget::muteColorOn

Definition at line 77 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), KPerfWidget::setPerf(), and setupUi().

12.96.3.67 QComboBox* Ui_KPerfWidget::numBox

Definition at line 49 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.68 ColorButton* Ui_KPerfWidget::numColorOff

Definition at line 67 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.69 ColorButton* Ui_KPerfWidget::numColorOn

Definition at line 48 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.70 QComboBox* Ui_KPerfWidget::scrollBox

Definition at line 71 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.71 ColorButton* Ui_KPerfWidget::scrollColorOff

Definition at line 75 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.72 ColorButton* Ui_KPerfWidget::scrollColorOn

Definition at line 73 of file ui_kperfwidget.h.

Referenced by KPerfWidget::KPerfWidget(), retranslateUi(), and setupUi().

12.96.3.73 QSpacerItem* Ui_KPerfWidget::verticalSpacer

Definition at line 63 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.74 QSpacerItem* Ui_KPerfWidget::verticalSpacer_2

Definition at line 51 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.75 QSpacerItem* Ui_KPerfWidget::verticalSpacer_3

Definition at line 64 of file ui_kperfwidget.h.

Referenced by setupUi().

12.96.3.76 QSpacerItem* Ui_KPerfWidget::verticalSpacer_4

Definition at line 57 of file ui_kperfwidget.h.

Referenced by setupUi().

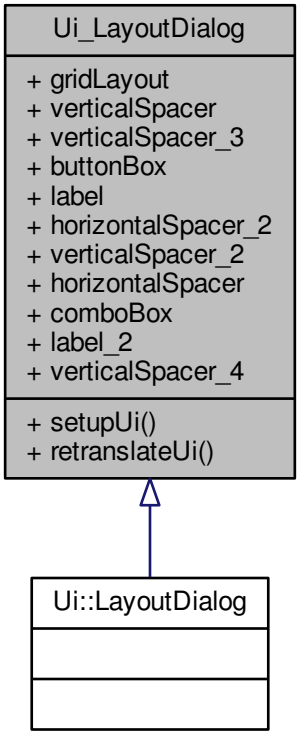
The documentation for this class was generated from the following file:

- src/ckb/ui_kperfwidget.h

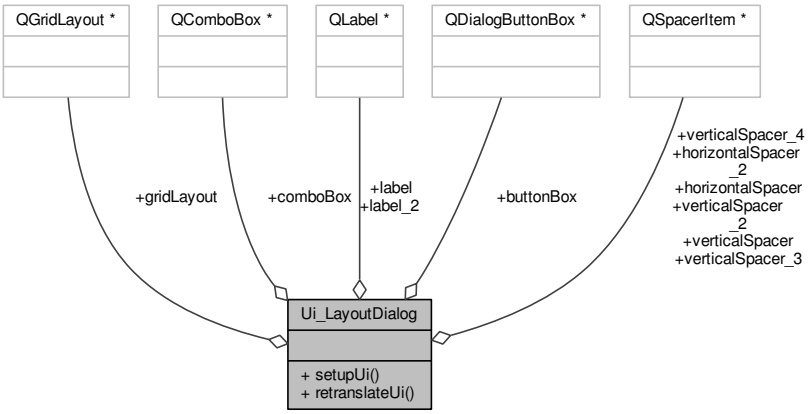
12.97 Ui_LayoutDialog Class Reference

`#include <src/ckb/ui_layoutdialog.h>`

Inheritance diagram for Ui_LayoutDialog:



Collaboration diagram for Ui_LayoutDialog:



Public Member Functions

- void setupUi (QDialog *LayoutDialog)

- void retranslateUi (QDialog *LayoutDialog)

Data Fields

- QGridLayout * gridLayout
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_3
- QDialogButtonBox * buttonBox
- QLabel * label
- QSpacerItem * horizontalSpacer_2
- QSpacerItem * verticalSpacer_2
- QSpacerItem * horizontalSpacer
- QComboBox * comboBox
- QLabel * label_2
- QSpacerItem * verticalSpacer_4

12.97.1 Detailed Description

Definition at line 26 of file ui_layoutdialog.h.

12.97.2 Member Function Documentation

12.97.2.1 void Ui_LayoutDialog::retranslateUi (QDialog * *LayoutDialog*) [inline]

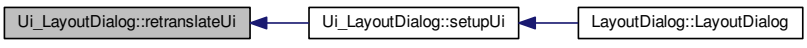
Definition at line 109 of file ui_layoutdialog.h.

References label, and label_2.

Referenced by setupUi().

```
110     {
111         LayoutDialog->setWindowTitle(QApplication::translate("LayoutDialog", "Keyboard Layout", Q_NULLPTR))
    ;
112         label->setText(QApplication::translate("LayoutDialog", "Please select your keyboard layout:",
    Q_NULLPTR));
113         label_2->setText(QApplication::translate("LayoutDialog", "Note: this only controls the
     layout within the ckb application. If your keys are typing the wrong characters, check your system preferences.",
     Q_NULLPTR));
114     } // retranslateUi
```

Here is the caller graph for this function:



12.97.2.2 void Ui_LayoutDialog::setupUi (QDialog * *LayoutDialog*) [inline]

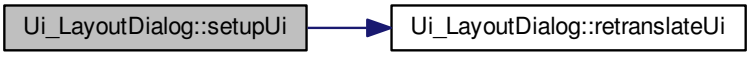
Definition at line 41 of file ui_layoutdialog.h.

References buttonBox, comboBox, gridLayout, horizontalSpacer, horizontalSpacer_2, label, label_2, retranslateUi(), verticalSpacer, verticalSpacer_2, verticalSpacer_3, and verticalSpacer_4.

Referenced by LayoutDialog::LayoutDialog().

```
42      {
43          if (LayoutDialog->objectName().isEmpty())
44              LayoutDialog->setObjectName(QStringLiteral("LayoutDialog"));
45          LayoutDialog->resize(400, 240);
46          gridLayout = new QGridLayout(LayoutDialog);
47          gridLayout->setObjectName(QStringLiteral("gridLayout"));
48          verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
49
50          gridLayout->addItem(verticalSpacer, 6, 0, 1, 3);
51
52          verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
53
54          gridLayout->addItem(verticalSpacer_3, 0, 1, 1, 1);
55
56          buttonBox = new QDialogButtonBox(LayoutDialog);
57          buttonBox->setObjectName(QStringLiteral("buttonBox"));
58          buttonBox->setOrientation(Qt::Horizontal);
59          buttonBox->setStandardButtons(QDialogButtonBox::Ok);
60
61          gridLayout->addWidget(buttonBox, 7, 0, 1, 3);
62
63          label = new QLabel(LayoutDialog);
64          label->setObjectName(QStringLiteral("label"));
65          label->setAlignment(Qt::AlignCenter);
66
67          gridLayout->addWidget(label, 1, 0, 1, 3);
68
69          horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
70
71          gridLayout->addItem(horizontalSpacer_2, 3, 2, 1, 1);
72
73          verticalSpacer_2 = new QSpacerItem(20, 15, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
74
75          gridLayout->addItem(verticalSpacer_2, 4, 1, 1, 1);
76
77          horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
78
79          gridLayout->addItem(horizontalSpacer, 3, 0, 1, 1);
80
81          comboBox = new QComboBox(LayoutDialog);
82          comboBox->setObjectName(QStringLiteral("comboBox"));
83
84          gridLayout->addWidget(comboBox, 3, 1, 1, 1);
85
86          label_2 = new QLabel(LayoutDialog);
87          label_2->setObjectName(QStringLiteral("label_2"));
88          QFont font;
89          font.setPointSize(8);
90          font.setItalic(true);
91          label_2->setFont(font);
92          label_2->setAlignment(Qt::AlignCenter);
93          label_2->setWordWrap(true);
94
95          gridLayout->addWidget(label_2, 5, 0, 1, 3);
96
97          verticalSpacer_4 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
98
99          gridLayout->addItem(verticalSpacer_4, 2, 1, 1, 1);
100
101
102          retranslateUi(LayoutDialog);
103          QObject::connect(buttonBox, SIGNAL(accepted()), LayoutDialog, SLOT(accept()));
104          QObject::connect(buttonBox, SIGNAL(rejected()), LayoutDialog, SLOT(reject()));
105
106          QMetaObject::connectSlotsByName(LayoutDialog);
107      } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.97.3 Field Documentation

12.97.3.1 QDialogButtonBox* Ui_LayoutDialog::buttonBox

Definition at line 32 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.2 QComboBox* Ui_LayoutDialog::comboBox

Definition at line 37 of file ui_layoutdialog.h.

Referenced by LayoutDialog::LayoutDialog(), LayoutDialog::selected(), and setupUi().

12.97.3.3 QGridLayout* Ui_LayoutDialog::gridLayout

Definition at line 29 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.4 QSpacerItem* Ui_LayoutDialog::horizontalSpacer

Definition at line 36 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.5 QSpacerItem* Ui_LayoutDialog::horizontalSpacer_2

Definition at line 34 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.6 QLabel* Ui_LayoutDialog::label

Definition at line 33 of file ui_layoutdialog.h.

Referenced by retranslateUi(), and setupUi().

12.97.3.7 QLabel* Ui_LayoutDialog::label_2

Definition at line 38 of file ui_layoutdialog.h.

Referenced by retranslateUi(), and setupUi().

12.97.3.8 QSpacerItem* Ui_LayoutDialog::verticalSpacer

Definition at line 30 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.9 QSpacerItem* Ui_LayoutDialog::verticalSpacer_2

Definition at line 35 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.10 QSpacerItem* Ui_LayoutDialog::verticalSpacer_3

Definition at line 31 of file ui_layoutdialog.h.

Referenced by setupUi().

12.97.3.11 QSpacerItem* Ui_LayoutDialog::verticalSpacer_4

Definition at line 39 of file ui_layoutdialog.h.

Referenced by setupUi().

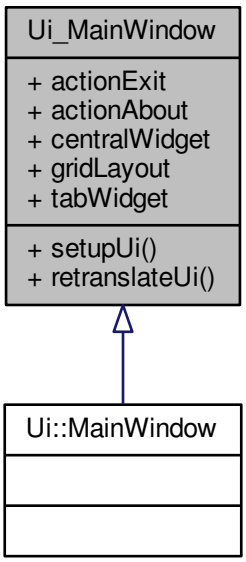
The documentation for this class was generated from the following file:

- src/ckb/ui_layoutdialog.h

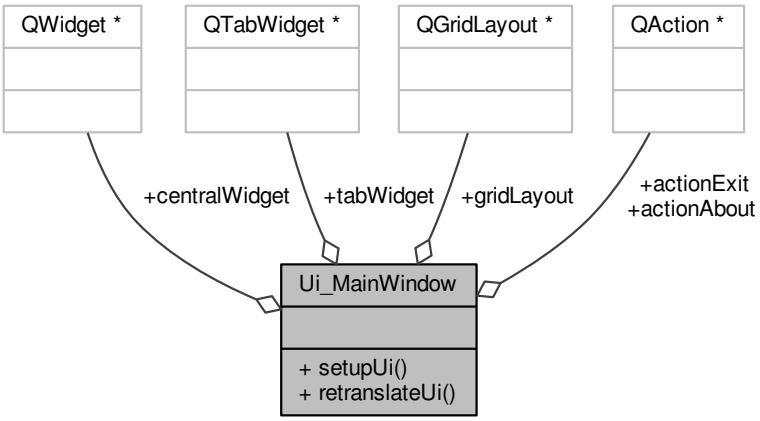
12.98 Ui_MainWindow Class Reference

`#include <src/ckb/ui_mainwindow.h>`

Inheritance diagram for Ui_MainWindow:



Collaboration diagram for Ui_MainWindow:



Public Member Functions

- void setupUi (QMainWindow *MainWindow)
- void retranslateUi (QMainWindow *MainWindow)

Data Fields

- QAction * actionExit
- QAction * actionAbout
- QWidget * centralWidget
- QGridLayout * gridLayout
- QTabWidget * tabWidget

12.98.1 Detailed Description

Definition at line 24 of file ui_mainwindow.h.

12.98.2 Member Function Documentation

12.98.2.1 void Ui_MainWindow::retranslateUi (QMainWindow * *MainWindow*) [inline]

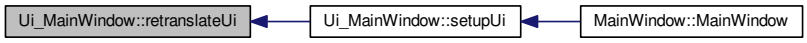
Definition at line 68 of file ui_mainwindow.h.

References actionAbout, and actionExit.

Referenced by setupUi().

```
69     {
70         MainWindow->setWindowTitle(QApplication::translate("MainWindow", "ckb-next", Q_NULLPTR));
71         actionExit->setText(QApplication::translate("MainWindow", "Exit", Q_NULLPTR));
72         actionExit->setShortcut(QApplication::translate("MainWindow", "Ctrl+Q", Q_NULLPTR));
73         actionAbout->setText(QApplication::translate("MainWindow", "About", Q_NULLPTR));
74     } // retranslateUi
```

Here is the caller graph for this function:



12.98.2.2 void Ui_MainWindow::setupUi (QMainWindow * *MainWindow*) [inline]

Definition at line 33 of file ui_mainwindow.h.

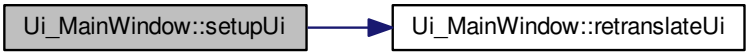
References actionAbout, actionExit, centralWidget, gridLayout, retranslateUi(), and tabWidget.

Referenced by MainWindow::MainWindow().

```
34     {
35         if (MainWindow->objectName().isEmpty())
36             MainWindow->setObjectName(QStringLiteral("MainWindow"));
37         MainWindow->resize(1000, 750);
38         MainWindow->setMinimumSize(QSize(1000, 750));
39         QIcon icon;
40         icon.addFile(QStringLiteral(":/img/ckb-logo.png"), QSize(), QIcon::Normal, QIcon::Off);
41         MainWindow->setWindowIcon(icon);
42         MainWindow->setUnifiedTitleAndToolBarOnMac(true);
43         actionExit = new QAction(MainWindow);
44         actionExit->setObjectName(QStringLiteral("actionExit"));
45         actionAbout = new QAction(MainWindow);
46         actionAbout->setObjectName(QStringLiteral("actionAbout"));
47         centralWidget = new QWidget(MainWindow);
48         centralWidget->setObjectName(QStringLiteral("centralWidget"));
49         gridLayout = new QGridLayout(centralWidget);
50         gridLayout->setSpacing(6);
```

```
51          gridLayout->setContentsMargins(11, 11, 11, 11);
52          gridLayout->setObjectName(QStringLiteral("gridLayout"));
53          tabWidget = new QTabWidget(centralWidget);
54          tabWidget->setObjectName(QStringLiteral("tabWidget"));
55
56          gridLayout->addWidget(tabWidget, 0, 0, 1, 1);
57
58          MainWindow->setCentralWidget(centralWidget);
59
60          retranslateUi(MainWindow);
61
62          tabWidget->setCurrentIndex(-1);
63
64
65          QMetaObject::connectSlotsByName(MainWindow);
66      } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.98.3 Field Documentation

12.98.3.1 QAction* Ui_MainWindow::actionAbout

Definition at line 28 of file ui_mainwindow.h.

Referenced by retranslateUi(), and setupUi().

12.98.3.2 QAction* Ui_MainWindow::actionExit

Definition at line 27 of file ui_mainwindow.h.

Referenced by MainWindow::MainWindow(), retranslateUi(), and setupUi().

12.98.3.3 QWidget* Ui_MainWindow::centralWidget

Definition at line 29 of file ui_mainwindow.h.

Referenced by setupUi().

12.98.3.4 QGridLayout* Ui_MainWindow::gridLayout

Definition at line 30 of file ui_mainwindow.h.

Referenced by setupUi().

12.98.3.5 QTabWidget* Ui_MainWindow::tabWidget

Definition at line 31 of file ui_mainwindow.h.

Referenced by MainWindow::addDevice(), MainWindow::MainWindow(), MainWindow::removeDevice(), setupUi(), and MainWindow::showFwUpdateNotification().

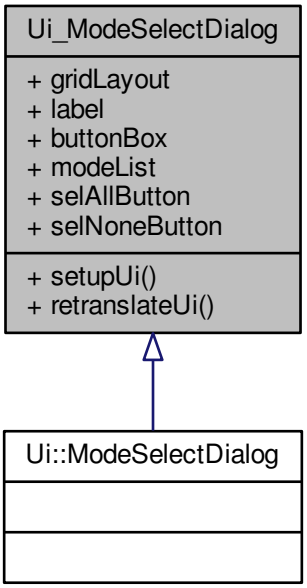
The documentation for this class was generated from the following file:

- src/ckb/ui_mainwindow.h

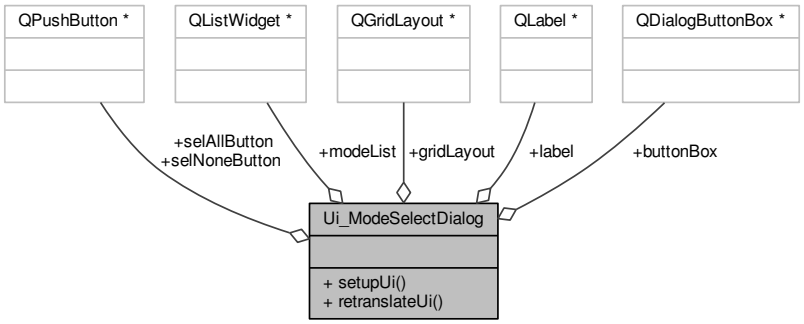
12.99 Ui_ModeSelectDialog Class Reference

`#include <src/ckb/ui_modeselectdialog.h>`

Inheritance diagram for Ui_ModeSelectDialog:



Collaboration diagram for Ui_ModeSelectDialog:



Public Member Functions

- void setupUi (QDialog *ModeSelectDialog)
- void retranslateUi (QDialog *ModeSelectDialog)

Data Fields

- QGridLayout * gridLayout
- QLabel * label
- QDialogButtonBox * buttonBox
- QListWidget * modeList
- QPushButton * selAllButton
- QPushButton * selNoneButton

12.99.1 Detailed Description

Definition at line 26 of file ui_modeselectdialog.h.

12.99.2 Member Function Documentation

12.99.2.1 void Ui_ModeSelectDialog::retranslateUi (QDialog * *ModeSelectDialog*) [inline]

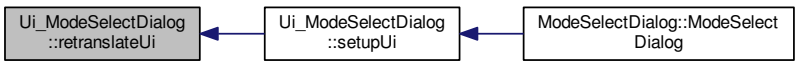
Definition at line 84 of file ui_modeselectdialog.h.

References label, selAllButton, and selNoneButton.

Referenced by setupUi().

```
85     {
86         ModeSelectDialog->setWindowTitle(QApplication::translate("ModeSelectDialog", "Select Modes",
      Q_NULLPTR));
87         label->setText(QApplication::translate("ModeSelectDialog", "TextLabel", Q_NULLPTR));
88         selAllButton->setText(QApplication::translate("ModeSelectDialog", "Select All",
      Q_NULLPTR));
89         selNoneButton->setText(QApplication::translate("ModeSelectDialog", "Select None",
      Q_NULLPTR));
90     } // retranslateUi
```

Here is the caller graph for this function:



12.99.2.2 void Ui_ModeSelectDialog::setupUi (QDialog * *ModeSelectDialog*) [inline]

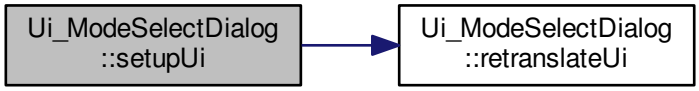
Definition at line 36 of file ui_modeselectdialog.h.

References buttonBox, gridLayout, label, modeList, retranslateUi(), selAllButton, and selNoneButton.

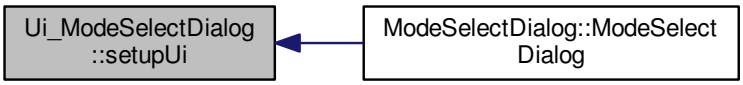
Referenced by ModeSelectDialog::ModeSelectDialog().

```
37     {
38         if (ModeSelectDialog->objectName().isEmpty())
39             ModeSelectDialog->setObjectName(QStringLiteral("ModeSelectDialog"));
40         ModeSelectDialog->resize(320, 320);
41         gridLayout = new QGridLayout(ModeSelectDialog);
42         gridLayout->setObjectName(QStringLiteral("gridLayout"));
43         label = new QLabel(ModeSelectDialog);
44         label->setObjectName(QStringLiteral("label"));
45         label->setAlignment(Qt::AlignLeading|Qt::AlignLeft|Qt::AlignTop);
46 
47         gridLayout->addWidget(label, 0, 0, 1, 2);
48 
49         buttonBox = new QDialogButtonBox(ModeSelectDialog);
50         buttonBox->setObjectName(QStringLiteral("buttonBox"));
51         buttonBox->setOrientation(Qt::Vertical);
52         buttonBox->setStandardButtons(QDialogButtonBox::Cancel|QDialogButtonBox::Ok);
53         buttonBox->setCenterButtons(false);
54 
55         gridLayout->addWidget(buttonBox, 0, 2, 3, 1);
56 
57         modeList = new QListWidget(ModeSelectDialog);
58         modeList->setObjectName(QStringLiteral("modeList"));
59         modeList->setFrameShadow(QFrame::Plain);
60         modeList->setSelectionMode(QAbstractItemView::ExtendedSelection);
61 
62         gridLayout->addWidget(modeList, 1, 0, 1, 2);
63 
64         selAllButton = new QPushButton(ModeSelectDialog);
65         selAllButton->setObjectName(QStringLiteral("selAllButton"));
66 
67         gridLayout->addWidget(selAllButton, 2, 0, 1, 1);
68 
69         selNoneButton = new QPushButton(ModeSelectDialog);
70         selNoneButton->setObjectName(QStringLiteral("selNoneButton"));
71 
72         gridLayout->addWidget(selNoneButton, 2, 1, 1, 1);
73 
74         QWidget::setTabOrder(modeList, selAllButton);
75         QWidget::setTabOrder(selAllButton, selNoneButton);
76 
77         retranslateUi(ModeSelectDialog);
78         QObject::connect(buttonBox, SIGNAL(accepted()), ModeSelectDialog, SLOT(accept()));
79         QObject::connect(buttonBox, SIGNAL(rejected()), ModeSelectDialog, SLOT(reject()));
80 
81         QMetaObject::connectSlotsByName(ModeSelectDialog);
82     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.99.3 Field Documentation

12.99.3.1 QDialogButtonBox* Ui_ModeSelectDialog::buttonBox

Definition at line 31 of file ui_modeselectdialog.h.

Referenced by setupUi().

12.99.3.2 QGridLayout* Ui_ModeSelectDialog::gridLayout

Definition at line 29 of file ui_modeselectdialog.h.

Referenced by setupUi().

12.99.3.3 QLabel* Ui_ModeSelectDialog::label

Definition at line 30 of file ui_modeselectdialog.h.

Referenced by ModeSelectDialog::ModeSelectDialog(), retranslateUi(), and setupUi().

12.99.3.4 QListWidget* Ui_ModeSelectDialog::modeList

Definition at line 32 of file ui_modeselectdialog.h.

Referenced by ModeSelectDialog::ModeSelectDialog(), ModeSelectDialog::on_selAllButton_clicked(), ModeSelectDialog::on_selNoneButton_clicked(), ModeSelectDialog::selection(), and setupUi().

12.99.3.5 QPushButton* Ui_ModeSelectDialog::selAllButton

Definition at line 33 of file ui_modeselectdialog.h.

Referenced by retranslateUi(), and setupUi().

12.99.3.6 QPushButton* Ui_ModeSelectDialog::selNoneButton

Definition at line 34 of file ui_modeselectdialog.h.

Referenced by retranslateUi(), and setupUi().

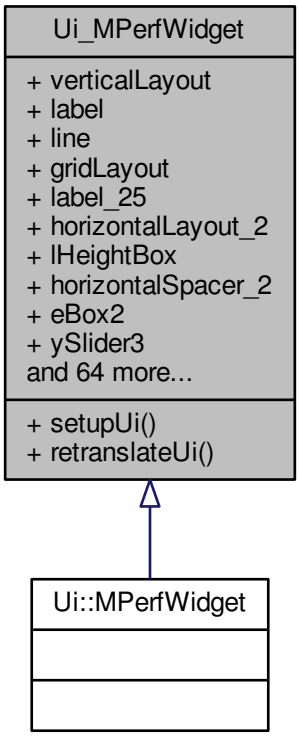
The documentation for this class was generated from the following file:

- src/ckb/ui_modeselectdialog.h

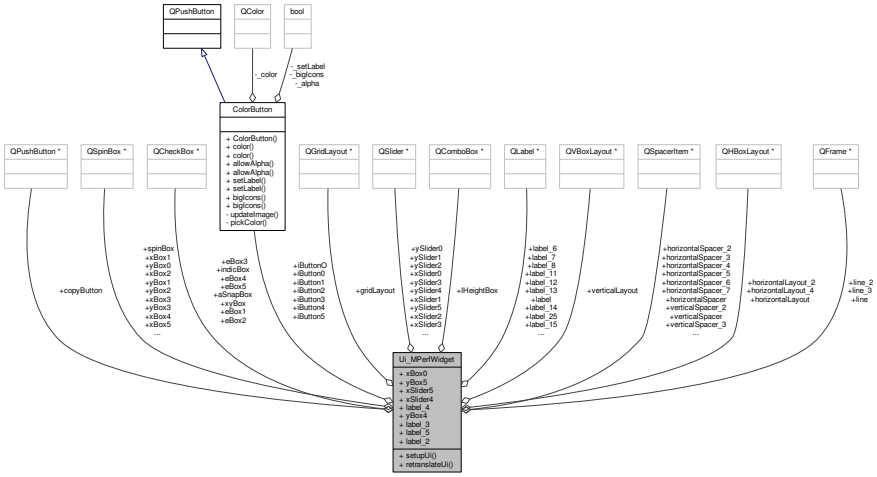
12.100 Ui_MPerfWidget Class Reference

`#include <src/ckb/ui_mperfwidget.h>`

Inheritance diagram for Ui_MPerfWidget:



Collaboration diagram for Ui_MPerfWidget:



Public Member Functions

- void setupUi (QWidget *MPerfWidget)
- void retranslateUi (QWidget *MPerfWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QLabel * label
- QFrame * line
- QGridLayout * gridLayout
- QLabel * label_25
- QHBoxLayout * horizontalLayout_2
- QComboBox * lHeightBox
- QSpacerItem * horizontalSpacer_2
- QCheckBox * eBox2
- QSlider * ySlider3
- QSlider * xSlider2
- QSpinBox * xBox3
- QSpinBox * yBox0
- QSlider * ySlider0
- ColorButton * iButton4
- ColorButton * iButton5
- QSpacerItem * horizontalSpacer_4
- QSpacerItem * verticalSpacer_3
- QSpacerItem * horizontalSpacer_5
- ColorButton * iButton3
- QLabel * label_12
- QSpinBox * xBox4
- QSpinBox * xBox0
- QSpinBox * yBox5
- QCheckBox * eBox5
- QSlider * xSlider5
- QSpinBox * xBox1

- QSpinBox * yBox2
- QSpinBox * yBox3
- QSlider * xSlider4
- QLabel * label_4
- QSlider * xSlider0
- ColorButton * iButton2
- QFrame * line_3
- ColorButton * iButtonO
- QSpinBox * yBox1
- QSlider * ySlider2
- QSpinBox * xBox2
- QSlider * xSlider1
- QSpacerItem * horizontalSpacer
- QSpinBox * yBox4
- ColorButton * iButton1
- QSlider * ySlider4
- QSlider * ySlider1
- ColorButton * iButton0
- QSlider * xSlider3
- QSpinBox * xBox5
- QLabel * label_3
- QLabel * label_6
- QSlider * ySlider5
- QLabel * label_7
- QLabel * label_8
- QLabel * label_5
- QLabel * label_15
- QLabel * label_13
- QLabel * label_14
- QCheckBox * eBox4
- QLabel * label_11
- QCheckBox * eBox3
- QCheckBox * eBox1
- QLabel * label_2
- QSpacerItem * verticalSpacer
- QCheckBox * aSnapBox
- QFrame * line_2
- QHBoxLayout * horizontalLayout_4
- QCheckBox * indicBox
- QSpacerItem * horizontalSpacer_3
- QSpinBox * spinBox
- QSpacerItem * horizontalSpacer_7
- QCheckBox * xyBox
- QSpacerItem * verticalSpacer_2
- QHBoxLayout * horizontalLayout
- QPushButton * copyButton
- QSpacerItem * horizontalSpacer_6

12.100.1 Detailed Description

Definition at line 33 of file ui_mperfwidget.h.

12.100.2 Member Function Documentation

12.100.2.1 void Ui_MPerfWidget::retranslateUi (QWidget * *MPerfWidget*) [inline]

Definition at line 682 of file ui_mperfwidget.h.

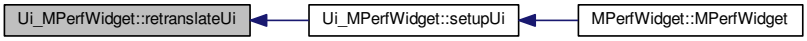
References aSnapBox, copyButton, eBox1, eBox2, eBox3, eBox4, eBox5, iButton0, iButton1, iButton2, iButton3, iButton4, iButton5, iButtonO, indicBox, label, label_11, label_12, label_13, label_14, label_15, label_2, label_25, label_3, label_4, label_5, label_6, label_7, label_8, lHeightBox, spinBox, and xyBox.

Referenced by setupUi().

```
683     {
684         MPerfWidget->setWindowTitle(QApplication::translate("MPerfWidget", "Form", Q_NULLPTR));
685         label->setText(QApplication::translate("MPerfWidget", "DPI", Q_NULLPTR));
686         label_25->setText(QApplication::translate("MPerfWidget", "Lift height:", Q_NULLPTR));
687         lHeightBox->clear();
688         lHeightBox->insertItems(0, QStringList()
689          << QApplication::translate("MPerfWidget", "Low", Q_NULLPTR)
690          << QApplication::translate("MPerfWidget", "Medium-low", Q_NULLPTR)
691          << QApplication::translate("MPerfWidget", "Medium", Q_NULLPTR)
692          << QApplication::translate("MPerfWidget", "Medium-high", Q_NULLPTR)
693          << QApplication::translate("MPerfWidget", "High", Q_NULLPTR)
694         );
695 #ifndef QT_NO_TOOLTIP
696         eBox2->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
       while moving up or down.", Q_NULLPTR));
697 #endif // QT_NO_TOOLTIP
698         eBox2->setText(QString());
699 #ifndef QT_NO_TOOLTIP
700         iButton4->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
701 #endif // QT_NO_TOOLTIP
702         iButton4->setText(QString());
703 #ifndef QT_NO_TOOLTIP
704         iButton5->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
705 #endif // QT_NO_TOOLTIP
706         iButton5->setText(QString());
707 #ifndef QT_NO_TOOLTIP
708         iButton3->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
709 #endif // QT_NO_TOOLTIP
710         iButton3->setText(QString());
711         label_12->setText(QApplication::translate("MPerfWidget", "X Value:", Q_NULLPTR));
712 #ifndef QT_NO_TOOLTIP
713         eBox5->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
       while moving up or down.", Q_NULLPTR));
714 #endif // QT_NO_TOOLTIP
715         eBox5->setText(QString());
716         label_4->setText(QApplication::translate("MPerfWidget", "Y Value:", Q_NULLPTR));
717 #ifndef QT_NO_TOOLTIP
718         iButton2->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
719 #endif // QT_NO_TOOLTIP
720         iButton2->setText(QString());
721 #ifndef QT_NO_TOOLTIP
722         iButtonO->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
723 #endif // QT_NO_TOOLTIP
724         iButtonO->setText(QString());
725 #ifndef QT_NO_TOOLTIP
726         iButton1->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
727 #endif // QT_NO_TOOLTIP
728         iButton1->setText(QString());
729 #ifndef QT_NO_TOOLTIP
730         iButton0->setToolTip(QApplication::translate("MPerfWidget", "Tip: alt+click to set all
       colors at once", Q_NULLPTR));
731 #endif // QT_NO_TOOLTIP
732         iButton0->setText(QString());
733         label_3->setText(QApplication::translate("MPerfWidget", "You can assign other DPIs in the
       Binding tab.", Q_NULLPTR));
734         label_6->setText(QApplication::translate("MPerfWidget", "2:", Q_NULLPTR));
735         label_7->setText(QApplication::translate("MPerfWidget", "3:", Q_NULLPTR));
736         label_8->setText(QApplication::translate("MPerfWidget", "4:", Q_NULLPTR));
737         label_5->setText(QApplication::translate("MPerfWidget", "1:", Q_NULLPTR));
738         label_15->setText(QApplication::translate("MPerfWidget", "Other:", Q_NULLPTR));
739         label_13->setText(QApplication::translate("MPerfWidget", "5:", Q_NULLPTR));
740         label_14->setText(QApplication::translate("MPerfWidget", "Sniper:", Q_NULLPTR));
741 #ifndef QT_NO_TOOLTIP
742         eBox4->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
       while moving up or down.", Q_NULLPTR));
```

```
743 #endif // QT_NO_TOOLTIP
744         eBox4->setText(QString());
745         label_11->setText(QApplication::translate("MPerfWidget", "DPI Stage:", Q_NULLPTR));
746 #ifndef QT_NO_TOOLTIP
747         eBox3->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
      while moving up or down.", Q_NULLPTR));
748 #endif // QT_NO_TOOLTIP
749         eBox3->setText(QString());
750 #ifndef QT_NO_TOOLTIP
751         eBox1->setToolTip(QApplication::translate("MPerfWidget", "Uncheck this box to skip the stage
      while moving up or down.", Q_NULLPTR));
752 #endif // QT_NO_TOOLTIP
753         eBox1->setText(QString());
754         label_2->setText(QApplication::translate("MPerfWidget", "Movement", Q_NULLPTR));
755         aSnapBox->setText(QApplication::translate("MPerfWidget", "Angle snap", Q_NULLPTR));
756         indicBox->setText(QApplication::translate("MPerfWidget", "Use DPI indicator:", Q_NULLPTR));
757 #ifndef QT_NO_TOOLTIP
758         spinBox->setToolTip(QApplication::translate("MPerfWidget", "Indicator intensity", Q_NULLPTR)
    );
759 #endif // QT_NO_TOOLTIP
760         spinBox->setSuffix(QApplication::translate("MPerfWidget", "%", Q_NULLPTR));
761         xyBox->setText(QApplication::translate("MPerfWidget", "Independent X/Y states", Q_NULLPTR));
762         copyButton->setText(QApplication::translate("MPerfWidget", "Copy to mode...", Q_NULLPTR))
    ;
763     } // retranslateUi
```

Here is the caller graph for this function:



12.100.2.2 void Ui_MPerfWidget::setupUi (QWidget * *MPerfWidget*) [inline]

Definition at line 111 of file ui_mperfwidget.h.

References aSnapBox, copyButton, eBox1, eBox2, eBox3, eBox4, eBox5, gridLayout, horizontalLayout, horizontalLayout_2, horizontalLayout_4, horizontalSpacer, horizontalSpacer_2, horizontalSpacer_3, horizontalSpacer_4, horizontalSpacer_5, horizontalSpacer_6, horizontalSpacer_7, iButton0, iButton1, iButton2, iButton3, iButton4, iButton5, iButtonO, indicBox, label, label_11, label_12, label_13, label_14, label_15, label_2, label_25, label_3, label_4, label_5, label_6, label_7, label_8, lHeightBox, line, line_2, line_3, retranslateUi(), spinBox, verticalLayout, verticalSpacer, verticalSpacer_2, verticalSpacer_3, xBox0, xBox1, xBox2, xBox3, xBox4, xBox5, xSlider0, xSlider1, xSlider2, xSlider3, xSlider4, xSlider5, xyBox, yBox0, yBox1, yBox2, yBox3, yBox4, yBox5, ySlider0, ySlider1, ySlider2, ySlider3, ySlider4, and ySlider5.

Referenced by MPerfWidget::MPerfWidget().

```
112     {
113         if (MPerfWidget->objectName().isEmpty())
114             MPerfWidget->setObjectName(QStringLiteral("MPerfWidget"));
115         MPerfWidget->resize(932, 637);
116         verticalLayout = new QVBoxLayout(MPerfWidget);
117         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
118         label = new QLabel(MPerfWidget);
119         label->setObjectName(QStringLiteral("label"));
120         QFont font;
121         font.setBold(true);
122         font.setWeight(75);
123         label->setFont(font);
124
125         verticalLayout->addWidget(label);
126
127         line = new QFrame(MPerfWidget);
128         line->setObjectName(QStringLiteral("line"));
129         line->setFrameShape(QFrame::HLine);
130         line->setFrameShadow(QFrame::Sunken);
131
132         verticalLayout->addWidget(line);
133
134         gridLayout = new QGridLayout();
```

```
135         gridLayout->setObjectName(QStringLiteral("gridLayout"));
136         label_25 = new QLabel(MPerfWidget);
137         label_25->setObjectName(QStringLiteral("label_25"));
138         label_25->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
139 
140         gridLayout->addWidget(label_25, 16, 5, 1, 2);
141 
142         horizontalLayout_2 = new QHBoxLayout();
143         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
144         lHeightBox = new QComboBox(MPerfWidget);
145         lHeightBox->setObjectName(QStringLiteral("lHeightBox"));
146 
147         horizontalLayout_2->addWidget(lHeightBox);
148 
149         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
150 
151         horizontalLayout_2->addItem(horizontalSpacer_2);
152 
153 
154         gridLayout->addLayout(horizontalLayout_2, 16, 8, 1, 3);
155 
156         eBox2 = new QCheckBox(MPerfWidget);
157         eBox2->setObjectName(QStringLiteral("eBox2"));
158 
159         gridLayout->addWidget(eBox2, 7, 1, 1, 1);
160 
161         ySlider3 = new QSlider(MPerfWidget);
162         ySlider3->setObjectName(QStringLiteral("ySlider3"));
163         ySlider3->setMinimum(0);
164         ySlider3->setMaximum(1000);
165         ySlider3->setSingleStep(10);
166         ySlider3->setPageStep(50);
167         ySlider3->setValue(0);
168         ySlider3->setOrientation(Qt::Horizontal);
169 
170         gridLayout->addWidget(ySlider3, 8, 8, 1, 1);
171 
172         xSlider2 = new QSlider(MPerfWidget);
173         xSlider2->setObjectName(QStringLiteral("xSlider2"));
174         xSlider2->setMinimum(0);
175         xSlider2->setMaximum(1000);
176         xSlider2->setSingleStep(10);
177         xSlider2->setPageStep(50);
178         xSlider2->setValue(0);
179         xSlider2->setOrientation(Qt::Horizontal);
180 
181         gridLayout->addWidget(xSlider2, 7, 5, 1, 1);
182 
183         xBox3 = new QSpinBox(MPerfWidget);
184         xBox3->setObjectName(QStringLiteral("xBox3"));
185         xBox3->setMinimum(100);
186         xBox3->setMaximum(12000);
187         xBox3->setSingleStep(50);
188 
189         gridLayout->addWidget(xBox3, 8, 6, 1, 1);
190 
191         yBox0 = new QSpinBox(MPerfWidget);
192         yBox0->setObjectName(QStringLiteral("yBox0"));
193         yBox0->setMinimum(100);
194         yBox0->setMaximum(12000);
195         yBox0->setSingleStep(50);
196 
197         gridLayout->addWidget(yBox0, 11, 9, 1, 1);
198 
199         ySlider0 = new QSlider(MPerfWidget);
200         ySlider0->setObjectName(QStringLiteral("ySlider0"));
201         ySlider0->setMinimum(0);
202         ySlider0->setMaximum(1000);
203         ySlider0->setSingleStep(10);
204         ySlider0->setPageStep(50);
205         ySlider0->setValue(0);
206         ySlider0->setOrientation(Qt::Horizontal);
207 
208         gridLayout->addWidget(ySlider0, 11, 8, 1, 1);
209 
210         iButton4 = new ColorButton(MPerfWidget);
211         iButton4->setObjectName(QStringLiteral("iButton4"));
212         QSizePolicy sizePolicy(QSizePolicy::Fixed, QSizePolicy::Fixed);
213         sizePolicy.setHorizontalStretch(0);
214         sizePolicy.setVerticalStretch(0);
215         sizePolicy.setHeightForWidth(iButton4->sizePolicy().hasHeightForWidth());
216         iButton4->setSizePolicy(sizePolicy);
217         iButton4->setMinimumSize(QSize(0, 40));
218 
219         gridLayout->addWidget(iButton4, 9, 2, 1, 1);
220 
```

```
221         iButton5 = new ColorButton(MPerfWidget);
222         iButton5->setObjectName(QStringLiteral("iButton5"));
223         sizePolicy.setHeightForWidth(iButton5->sizePolicy().hasHeightForWidth());
224         iButton5->setSizePolicy(sizePolicy);
225         iButton5->setMinimumSize(QSize(0, 40));
226 
227         gridLayout->addWidget(iButton5, 10, 2, 1, 1);
228 
229         horizontalSpacer_4 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
230 
231         gridLayout->addItem(horizontalSpacer_4, 4, 4, 1, 1);
232 
233         verticalSpacer_3 = new QSpacerItem(20, 15, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
234 
235         gridLayout->addItem(verticalSpacer_3, 3, 2, 1, 1);
236 
237         horizontalSpacer_5 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
238 
239         gridLayout->addItem(horizontalSpacer_5, 4, 7, 1, 1);
240 
241         iButton3 = new ColorButton(MPerfWidget);
242         iButton3->setObjectName(QStringLiteral("iButton3"));
243         sizePolicy.setHeightForWidth(iButton3->sizePolicy().hasHeightForWidth());
244         iButton3->setSizePolicy(sizePolicy);
245         iButton3->setMinimumSize(QSize(0, 40));
246 
247         gridLayout->addWidget(iButton3, 8, 2, 1, 1);
248 
249         label_12 = new QLabel(MPerfWidget);
250         label_12->setObjectName(QStringLiteral("label_12"));
251         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Preferred);
252         sizePolicy1.setHorizontalStretch(0);
253         sizePolicy1.setVerticalStretch(0);
254         sizePolicy1.setHeightForWidth(label_12->sizePolicy().hasHeightForWidth());
255         label_12->setSizePolicy(sizePolicy1);
256 
257         gridLayout->addWidget(label_12, 4, 5, 1, 1);
258 
259         xBox4 = new QSpinBox(MPerfWidget);
260         xBox4->setObjectName(QStringLiteral("xBox4"));
261         xBox4->setMinimum(100);
262         xBox4->setMaximum(12000);
263         xBox4->setSingleStep(50);
264 
265         gridLayout->addWidget(xBox4, 9, 6, 1, 1);
266 
267         xBox0 = new QSpinBox(MPerfWidget);
268         xBox0->setObjectName(QStringLiteral("xBox0"));
269         xBox0->setMinimum(100);
270         xBox0->setMaximum(12000);
271         xBox0->setSingleStep(50);
272 
273         gridLayout->addWidget(xBox0, 11, 6, 1, 1);
274 
275         yBox5 = new QSpinBox(MPerfWidget);
276         yBox5->setObjectName(QStringLiteral("yBox5"));
277         yBox5->setMinimum(100);
278         yBox5->setMaximum(12000);
279         yBox5->setSingleStep(50);
280 
281         gridLayout->addWidget(yBox5, 10, 9, 1, 1);
282 
283         eBox5 = new QCheckBox(MPerfWidget);
284         eBox5->setObjectName(QStringLiteral("eBox5"));
285 
286         gridLayout->addWidget(eBox5, 10, 1, 1, 1);
287 
288         xSlider5 = new QSlider(MPerfWidget);
289         xSlider5->setObjectName(QStringLiteral("xSlider5"));
290         xSlider5->setMinimum(0);
291         xSlider5->setMaximum(1000);
292         xSlider5->setSingleStep(10);
293         xSlider5->setPageStep(50);
294         xSlider5->setValue(0);
295         xSlider5->setOrientation(Qt::Horizontal);
296 
297         gridLayout->addWidget(xSlider5, 10, 5, 1, 1);
298 
299         xBox1 = new QSpinBox(MPerfWidget);
300         xBox1->setObjectName(QStringLiteral("xBox1"));
301         sizePolicy.setHeightForWidth(xBox1->sizePolicy().hasHeightForWidth());
302         xBox1->setSizePolicy(sizePolicy);
303         xBox1->setMinimum(100);
304         xBox1->setMaximum(12000);
```

```
305         xBox1->setSingleStep(50);
306
307         gridLayout->addWidget(xBox1, 6, 6, 1, 1);
308
309         yBox2 = new QSpinBox(MPerfWidget);
310         yBox2->setObjectName(QStringLiteral("yBox2"));
311         yBox2->setMinimum(100);
312         yBox2->setMaximum(12000);
313         yBox2->setSingleStep(50);
314
315         gridLayout->addWidget(yBox2, 7, 9, 1, 1);
316
317         yBox3 = new QSpinBox(MPerfWidget);
318         yBox3->setObjectName(QStringLiteral("yBox3"));
319         yBox3->setMinimum(100);
320         yBox3->setMaximum(12000);
321         yBox3->setSingleStep(50);
322
323         gridLayout->addWidget(yBox3, 8, 9, 1, 1);
324
325         xSlider4 = new QSlider(MPerfWidget);
326         xSlider4->setObjectName(QStringLiteral("xSlider4"));
327         xSlider4->setMinimum(0);
328         xSlider4->setMaximum(1000);
329         xSlider4->setSingleStep(10);
330         xSlider4->setPageStep(50);
331         xSlider4->setValue(0);
332         xSlider4->setOrientation(Qt::Horizontal);
333
334         gridLayout->addWidget(xSlider4, 9, 5, 1, 1);
335
336         label_4 = new QLabel(MPerfWidget);
337         label_4->setObjectName(QStringLiteral("label_4"));
338         sizePolicy1.setHeightForWidth(label_4->sizePolicy().hasHeightForWidth());
339         label_4->setSizePolicy(sizePolicy1);
340
341         gridLayout->addWidget(label_4, 4, 8, 1, 1);
342
343         xSlider0 = new QSlider(MPerfWidget);
344         xSlider0->setObjectName(QStringLiteral("xSlider0"));
345         xSlider0->setMinimum(0);
346         xSlider0->setMaximum(1000);
347         xSlider0->setSingleStep(10);
348         xSlider0->setPageStep(50);
349         xSlider0->setValue(0);
350         xSlider0->setOrientation(Qt::Horizontal);
351
352         gridLayout->addWidget(xSlider0, 11, 5, 1, 1);
353
354         iButton2 = new ColorButton(MPerfWidget);
355         iButton2->setObjectName(QStringLiteral("iButton2"));
356         sizePolicy.setHeightForWidth(iButton2->sizePolicy().hasHeightForWidth());
357         iButton2->setSizePolicy(sizePolicy);
358         iButton2->setMinimumSize(QSize(0, 40));
359
360         gridLayout->addWidget(iButton2, 7, 2, 1, 1);
361
362         line_3 = new QFrame(MPerfWidget);
363         line_3->setObjectName(QStringLiteral("line_3"));
364         line_3->setFrameShape(QFrame::HLine);
365         line_3->setFrameShadow(QFrame::Sunken);
366
367         gridLayout->addWidget(line_3, 5, 0, 1, 11);
368
369         iButton0 = new ColorButton(MPerfWidget);
370         iButton0->setObjectName(QStringLiteral("iButton0"));
371         sizePolicy.setHeightForWidth(iButton0->sizePolicy().hasHeightForWidth());
372         iButton0->setSizePolicy(sizePolicy);
373         iButton0->setMinimumSize(QSize(0, 40));
374
375         gridLayout->addWidget(iButton0, 12, 2, 1, 1);
376
377         yBox1 = new QSpinBox(MPerfWidget);
378         yBox1->setObjectName(QStringLiteral("yBox1"));
379         sizePolicy.setHeightForWidth(yBox1->sizePolicy().hasHeightForWidth());
380         yBox1->setSizePolicy(sizePolicy);
381         yBox1->setMinimum(100);
382         yBox1->setMaximum(12000);
383         yBox1->setSingleStep(50);
384
385         gridLayout->addWidget(yBox1, 6, 9, 1, 1);
386
387         ySlider2 = new QSlider(MPerfWidget);
388         ySlider2->setObjectName(QStringLiteral("ySlider2"));
389         ySlider2->setMinimum(0);
390         ySlider2->setMaximum(1000);
391         ySlider2->setSingleStep(10);
```

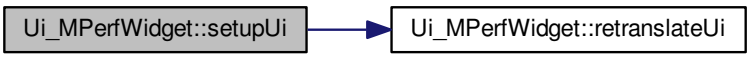
```
392         ySlider2->setPageStep(50);
393         ySlider2->setValue(0);
394         ySlider2->setOrientation(Qt::Horizontal);
395 
396         gridLayout->addWidget(ySlider2, 7, 8, 1, 1);
397 
398         xBox2 = new QSpinBox(MPerfWidget);
399         xBox2->setObjectName(QStringLiteral("xBox2"));
400         xBox2->setMinimum(100);
401         xBox2->setMaximum(12000);
402         xBox2->setSingleStep(50);
403 
404         gridLayout->addWidget(xBox2, 7, 6, 1, 1);
405 
406         xSlider1 = new QSlider(MPerfWidget);
407         xSlider1->setObjectName(QStringLiteral("xSlider1"));
408         QSizePolicy sizePolicy2(QSizePolicy::Expanding, QSizePolicy::Fixed);
409         sizePolicy2.setHorizontalStretch(0);
410         sizePolicy2.setVerticalStretch(0);
411         sizePolicy2.setHeightForWidth(xSlider1->sizePolicy().hasHeightForWidth());
412         xSlider1->setSizePolicy(sizePolicy2);
413         xSlider1->setMinimum(0);
414         xSlider1->setMaximum(1000);
415         xSlider1->setSingleStep(10);
416         xSlider1->setPageStep(50);
417         xSlider1->setValue(0);
418         xSlider1->setOrientation(Qt::Horizontal);
419 
420         gridLayout->addWidget(xSlider1, 6, 5, 1, 1);
421 
422         horizontalSpacer = new QSpacerItem(5, 20, QSizePolicy::Fixed, QSizePolicy::Minimum)
    ;
423 
424         gridLayout->addItem(horizontalSpacer, 0, 0, 1, 1);
425 
426         yBox4 = new QSpinBox(MPerfWidget);
427         yBox4->setObjectName(QStringLiteral("yBox4"));
428         yBox4->setMinimum(100);
429         yBox4->setMaximum(12000);
430         yBox4->setSingleStep(50);
431 
432         gridLayout->addWidget(yBox4, 9, 9, 1, 1);
433 
434         iButton1 = new ColorButton(MPerfWidget);
435         iButton1->setObjectName(QStringLiteral("iButton1"));
436         sizePolicy.setHeightForWidth(iButton1->sizePolicy().hasHeightForWidth());
437         iButton1->setSizePolicy(sizePolicy);
438         iButton1->setMinimumSize(QSize(0, 40));
439 
440         gridLayout->addWidget(iButton1, 6, 2, 1, 1);
441 
442         ySlider4 = new QSlider(MPerfWidget);
443         ySlider4->setObjectName(QStringLiteral("ySlider4"));
444         ySlider4->setMinimum(0);
445         ySlider4->setMaximum(1000);
446         ySlider4->setSingleStep(10);
447         ySlider4->setPageStep(50);
448         ySlider4->setValue(0);
449         ySlider4->setOrientation(Qt::Horizontal);
450 
451         gridLayout->addWidget(ySlider4, 9, 8, 1, 1);
452 
453         ySlider1 = new QSlider(MPerfWidget);
454         ySlider1->setObjectName(QStringLiteral("ySlider1"));
455         ySlider1->setMinimum(0);
456         ySlider1->setMaximum(1000);
457         ySlider1->setSingleStep(10);
458         ySlider1->setPageStep(50);
459         ySlider1->setValue(0);
460         ySlider1->setOrientation(Qt::Horizontal);
461 
462         gridLayout->addWidget(ySlider1, 6, 8, 1, 1);
463 
464         iButton0 = new ColorButton(MPerfWidget);
465         iButton0->setObjectName(QStringLiteral("iButton0"));
466         sizePolicy.setHeightForWidth(iButton0->sizePolicy().hasHeightForWidth());
467         iButton0->setSizePolicy(sizePolicy);
468         iButton0->setMinimumSize(QSize(0, 40));
469 
470         gridLayout->addWidget(iButton0, 11, 2, 1, 1);
471 
472         xSlider3 = new QSlider(MPerfWidget);
473         xSlider3->setObjectName(QStringLiteral("xSlider3"));
474         xSlider3->setMinimum(0);
475         xSlider3->setMaximum(1000);
476         xSlider3->setSingleStep(10);
477         xSlider3->setPageStep(50);
```

```
478         xSlider3->setValue(0);
479         xSlider3->setOrientation(Qt::Horizontal);
480 
481         gridLayout->addWidget(xSlider3, 8, 5, 1, 1);
482 
483         xBox5 = new QSpinBox(MPerfWidget);
484         xBox5->setObjectName(QStringLiteral("xBox5"));
485         xBox5->setMinimum(100);
486         xBox5->setMaximum(12000);
487         xBox5->setSingleStep(50);
488 
489         gridLayout->addWidget(xBox5, 10, 6, 1, 1);
490 
491         label_3 = new QLabel(MPerfWidget);
492         label_3->setObjectName(QStringLiteral("label_3"));
493 
494         gridLayout->addWidget(label_3, 12, 5, 1, 5);
495 
496         label_6 = new QLabel(MPerfWidget);
497         label_6->setObjectName(QStringLiteral("label_6"));
498 
499         gridLayout->addWidget(label_6, 7, 3, 1, 1);
500 
501         ySlider5 = new QSlider(MPerfWidget);
502         ySlider5->setObjectName(QStringLiteral("ySlider5"));
503         ySlider5->setMinimum(0);
504         ySlider5->setMaximum(1000);
505         ySlider5->setSingleStep(10);
506         ySlider5->setPageStep(50);
507         ySlider5->setValue(0);
508         ySlider5->setOrientation(Qt::Horizontal);
509 
510         gridLayout->addWidget(ySlider5, 10, 8, 1, 1);
511 
512         label_7 = new QLabel(MPerfWidget);
513         label_7->setObjectName(QStringLiteral("label_7"));
514 
515         gridLayout->addWidget(label_7, 8, 3, 1, 1);
516 
517         label_8 = new QLabel(MPerfWidget);
518         label_8->setObjectName(QStringLiteral("label_8"));
519 
520         gridLayout->addWidget(label_8, 9, 3, 1, 1);
521 
522         label_5 = new QLabel(MPerfWidget);
523         label_5->setObjectName(QStringLiteral("label_5"));
524 
525         gridLayout->addWidget(label_5, 6, 3, 1, 1);
526 
527         label_15 = new QLabel(MPerfWidget);
528         label_15->setObjectName(QStringLiteral("label_15"));
529 
530         gridLayout->addWidget(label_15, 12, 3, 1, 1);
531 
532         label_13 = new QLabel(MPerfWidget);
533         label_13->setObjectName(QStringLiteral("label_13"));
534 
535         gridLayout->addWidget(label_13, 10, 3, 1, 1);
536 
537         label_14 = new QLabel(MPerfWidget);
538         label_14->setObjectName(QStringLiteral("label_14"));
539 
540         gridLayout->addWidget(label_14, 11, 3, 1, 1);
541 
542         eBox4 = new QCheckBox(MPerfWidget);
543         eBox4->setObjectName(QStringLiteral("eBox4"));
544 
545         gridLayout->addWidget(eBox4, 9, 1, 1, 1);
546 
547         label_11 = new QLabel(MPerfWidget);
548         label_11->setObjectName(QStringLiteral("label_11"));
549 
550         gridLayout->addWidget(label_11, 4, 2, 1, 2);
551 
552         eBox3 = new QCheckBox(MPerfWidget);
553         eBox3->setObjectName(QStringLiteral("eBox3"));
554 
555         gridLayout->addWidget(eBox3, 8, 1, 1, 1);
556 
557         eBox1 = new QCheckBox(MPerfWidget);
558         eBox1->setObjectName(QStringLiteral("eBox1"));
559 
560         gridLayout->addWidget(eBox1, 6, 1, 1, 1);
561 
562         label_2 = new QLabel(MPerfWidget);
563         label_2->setObjectName(QStringLiteral("label_2"));
564         label_2->setFont(font);
```

```
565
566         gridLayout->addWidget(label_2, 14, 0, 1, 10);
567
568         verticalSpacer = new QSpacerItem(20, 30, QSizePolicy::Minimum, QSizePolicy::Fixed);
569
570         gridLayout->addItem(verticalSpacer, 13, 2, 1, 1);
571
572         aSnapBox = new QCheckBox(MPerfWidget);
573         aSnapBox->setObjectName(QStringLiteral("aSnapBox"));
574
575         gridLayout->addWidget(aSnapBox, 16, 1, 1, 4);
576
577         line_2 = new QFrame(MPerfWidget);
578         line_2->setObjectName(QStringLiteral("line_2"));
579         line_2->setFrameShape(QFrame::HLine);
580         line_2->setFrameShadow(QFrame::Sunken);
581
582         gridLayout->addWidget(line_2, 15, 0, 1, 10);
583
584         horizontalLayout_4 = new QHBoxLayout();
585         horizontalLayout_4->setObjectName(QStringLiteral("horizontalLayout_4"));
586         indicBox = new QCheckBox(MPerfWidget);
587         indicBox->setObjectName(QStringLiteral("indicBox"));
588
589         horizontalLayout_4->addWidget(indicBox);
590
591         horizontalSpacer_3 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
592
593         horizontalLayout_4->addItem(horizontalSpacer_3);
594
595         spinBox = new QSpinBox(MPerfWidget);
596         spinBox->setObjectName(QStringLiteral("spinBox"));
597         spinBox->setMaximum(100);
598         spinBox->setValue(100);
599
600         horizontalLayout_4->addWidget(spinBox);
601
602         horizontalSpacer_7 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
603
604         horizontalLayout_4->addItem(horizontalSpacer_7);
605
606
607         gridLayout->addLayout(horizontalLayout_4, 0, 1, 1, 9);
608
609         xyBox = new QCheckBox(MPerfWidget);
610         xyBox->setObjectName(QStringLiteral("xyBox"));
611
612         gridLayout->addWidget(xyBox, 2, 1, 1, 9);
613
614
615         verticalLayout->addLayout(gridLayout);
616
617         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
618
619         verticalLayout->addItem(verticalSpacer_2);
620
621         horizontalLayout = new QHBoxLayout();
622         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
623         copyButton = new QPushButton(MPerfWidget);
624         copyButton->setObjectName(QStringLiteral("copyButton"));
625
626         horizontalLayout->addWidget(copyButton);
627
628         horizontalSpacer_6 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
629
630         horizontalLayout->addItem(horizontalSpacer_6);
631
632
633         verticalLayout->addLayout(horizontalLayout);
634
635         QWidget::setTabOrder(indicBox, spinBox);
636         QWidget::setTabOrder(spinBox, xyBox);
637         QWidget::setTabOrder(xyBox, eBox1);
638         QWidget::setTabOrder(eBox1, eBox2);
639         QWidget::setTabOrder(eBox2, eBox3);
640         QWidget::setTabOrder(eBox3, eBox4);
641         QWidget::setTabOrder(eBox4, eBox5);
642         QWidget::setTabOrder(eBox5, iButton1);
643         QWidget::setTabOrder(iButton1, iButton2);
644         QWidget::setTabOrder(iButton2, iButton3);
645         QWidget::setTabOrder(iButton3, iButton4);
646         QWidget::setTabOrder(iButton4, iButton5);
647         QWidget::setTabOrder(iButton5, iButton0);
```

```
648         QWidget::setTabOrder(iButton0, iButtonO);
649         QWidget::setTabOrder(iButtonO, xSlider1);
650         QWidget::setTabOrder(xSlider1, xSlider2);
651         QWidget::setTabOrder(xSlider2, xSlider3);
652         QWidget::setTabOrder(xSlider3, xSlider4);
653         QWidget::setTabOrder(xSlider4, xSlider5);
654         QWidget::setTabOrder(xSlider5, xSlider0);
655         QWidget::setTabOrder(xSlider0, xBox1);
656         QWidget::setTabOrder(xBox1, xBox2);
657         QWidget::setTabOrder(xBox2, xBox3);
658         QWidget::setTabOrder(xBox3, xBox4);
659         QWidget::setTabOrder(xBox4, xBox5);
660         QWidget::setTabOrder(xBox5, xBox0);
661         QWidget::setTabOrder(xBox0, ySlider1);
662         QWidget::setTabOrder(ySlider1, ySlider2);
663         QWidget::setTabOrder(ySlider2, ySlider3);
664         QWidget::setTabOrder(ySlider3, ySlider4);
665         QWidget::setTabOrder(ySlider4, ySlider5);
666         QWidget::setTabOrder(ySlider5, ySlider0);
667         QWidget::setTabOrder(ySlider0, yBox1);
668         QWidget::setTabOrder(yBox1, yBox2);
669         QWidget::setTabOrder(yBox2, yBox3);
670         QWidget::setTabOrder(yBox3, yBox4);
671         QWidget::setTabOrder(yBox4, yBox5);
672         QWidget::setTabOrder(yBox5, yBox0);
673         QWidget::setTabOrder(yBox0, aSnapBox);
674         QWidget::setTabOrder(aSnapBox, lHeightBox);
675         QWidget::setTabOrder(lHeightBox, copyButton);
676 
677         retranslateUi(MPerfWidget);
678 
679         QMetaObject::connectSlotsByName(MPerfWidget);
680     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.100.3 Field Documentation

12.100.3.1 QCheckBox* Ui_MPerfWidget::aSnapBox

Definition at line 98 of file ui_mperfwidget.h.

Referenced by retranslateUi(), MPerfWidget::setPerf(), and setupUi().

12.100.3.2 QPushButton* Ui_MPerfWidget::copyButton

Definition at line 108 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.3 QCheckBox* Ui_MPerfWidget::eBox1

Definition at line 95 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.4 QCheckBox* Ui_MPerfWidget::eBox2

Definition at line 44 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.5 QCheckBox* Ui_MPerfWidget::eBox3

Definition at line 94 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.6 QCheckBox* Ui_MPerfWidget::eBox4

Definition at line 92 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.7 QCheckBox* Ui_MPerfWidget::eBox5

Definition at line 60 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.8 QGridLayout* Ui_MPerfWidget::gridLayout

Definition at line 39 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.9 QHBoxLayout* Ui_MPerfWidget::horizontalLayout

Definition at line 107 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.10 QHBoxLayout* Ui_MPerfWidget::horizontalLayout_2

Definition at line 41 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.11 QHBoxLayout* Ui_MPerfWidget::horizontalLayout_4

Definition at line 100 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.12 QSpacerItem* Ui_MPerfWidget::horizontalSpacer

Definition at line 75 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.13 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_2

Definition at line 43 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.14 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_3

Definition at line 102 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.15 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_4

Definition at line 52 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.16 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_5

Definition at line 54 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.17 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_6

Definition at line 109 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.18 QSpacerItem* Ui_MPerfWidget::horizontalSpacer_7

Definition at line 104 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.19 ColorButton* Ui_MPerfWidget::iButton0

Definition at line 80 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.20 ColorButton* Ui_MPerfWidget::iButton1

Definition at line 77 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.21 ColorButton* Ui_MPerfWidget::iButton2

Definition at line 68 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.22 ColorButton* Ui_MPerfWidget::iButton3

Definition at line 55 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.23 ColorButton* Ui_MPerfWidget::iButton4

Definition at line 50 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.24 ColorButton* Ui_MPerfWidget::iButton5

Definition at line 51 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), and setupUi().

12.100.3.25 ColorButton* Ui_MPerfWidget::iButtonO

Definition at line 70 of file ui_mperfwidget.h.

Referenced by MPerfWidget::colorChanged(), MPerfWidget::MPerfWidget(), retranslateUi(), MPerfWidget::setPerf(), and setupUi().

12.100.3.26 QCheckBox* Ui_MPerfWidget::indicBox

Definition at line 101 of file ui_mperfwidget.h.

Referenced by retranslateUi(), MPerfWidget::setPerf(), and setupUi().

12.100.3.27 QLabel* Ui_MPerfWidget::label

Definition at line 37 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.28 QLabel* Ui_MPerfWidget::label_11

Definition at line 93 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.29 QLabel* Ui_MPerfWidget::label_12

Definition at line 56 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.30 QLabel* Ui_MPerfWidget::label_13

Definition at line 90 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.31 QLabel* Ui_MPerfWidget::label_14

Definition at line 91 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.32 QLabel* Ui_MPerfWidget::label_15

Definition at line 89 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.33 QLabel* Ui_MPerfWidget::label_2

Definition at line 96 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.34 QLabel* Ui_MPerfWidget::label_25

Definition at line 40 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.35 QLabel* Ui_MPerfWidget::label_3

Definition at line 83 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.36 QLabel* Ui_MPerfWidget::label_4

Definition at line 66 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.37 QLabel* Ui_MPerfWidget::label_5

Definition at line 88 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.38 QLabel* Ui_MPerfWidget::label_6

Definition at line 84 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.39 QLabel* Ui_MPerfWidget::label_7

Definition at line 86 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.40 QLabel* Ui_MPerfWidget::label_8

Definition at line 87 of file ui_mperfwidget.h.

Referenced by retranslateUi(), and setupUi().

12.100.3.41 QComboBox* Ui_MPerfWidget::lHeightBox

Definition at line 42 of file ui_mperfwidget.h.

Referenced by retranslateUi(), MPerfWidget::setPerf(), and setupUi().

12.100.3.42 QFrame* Ui_MPerfWidget::line

Definition at line 38 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.43 QFrame* Ui_MPerfWidget::line_2

Definition at line 99 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.44 QFrame* Ui_MPerfWidget::line_3

Definition at line 69 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.45 QSpinBox* Ui_MPerfWidget::spinBox

Definition at line 103 of file ui_mperfwidget.h.

Referenced by retranslateUi(), MPerfWidget::setPerf(), and setupUi().

12.100.3.46 QVBoxLayout* Ui_MPerfWidget::verticalLayout

Definition at line 36 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.47 QSpacerItem* Ui_MPerfWidget::verticalSpacer

Definition at line 97 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.48 QSpacerItem* Ui_MPerfWidget::verticalSpacer_2

Definition at line 106 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.49 QSpacerItem* Ui_MPerfWidget::verticalSpacer_3

Definition at line 53 of file ui_mperfwidget.h.

Referenced by setupUi().

12.100.3.50 QSpinBox* Ui_MPerfWidget::xBox0

Definition at line 58 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.51 QSpinBox* Ui_MPerfWidget::xBox1

Definition at line 62 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.52 QSpinBox* Ui_MPerfWidget::xBox2

Definition at line 73 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.53 QSpinBox* Ui_MPerfWidget::xBox3

Definition at line 47 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.54 QSpinBox* Ui_MPerfWidget::xBox4

Definition at line 57 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.55 QSpinBox* Ui_MPerfWidget::xBox5

Definition at line 82 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.56 QSlider* Ui_MPerfWidget::xSlider0

Definition at line 67 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.57 QSlider* Ui_MPerfWidget::xSlider1

Definition at line 74 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.58 QSlider* Ui_MPerfWidget::xSlider2

Definition at line 46 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.59 QSlider* Ui_MPerfWidget::xSlider3

Definition at line 81 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.60 QSlider* Ui_MPerfWidget::xSlider4

Definition at line 65 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.61 QSlider* Ui_MPerfWidget::xSlider5

Definition at line 61 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.62 QCheckBox* Ui_MPerfWidget::xyBox

Definition at line 105 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), retranslateUi(), setupUi(), and MPerfWidget::sliderYMoved().

12.100.3.63 QSpinBox* Ui_MPerfWidget::yBox0

Definition at line 48 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.64 QSpinBox* Ui_MPerfWidget::yBox1

Definition at line 71 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.65 QSpinBox* Ui_MPerfWidget::yBox2

Definition at line 63 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.66 QSpinBox* Ui_MPerfWidget::yBox3

Definition at line 64 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.67 QSpinBox* Ui_MPerfWidget::yBox4

Definition at line 76 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.68 QSpinBox* Ui_MPerfWidget::yBox5

Definition at line 59 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.69 QSlider* Ui_MPerfWidget::ySlider0

Definition at line 49 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.70 QSlider* Ui_MPerfWidget::ySlider1

Definition at line 79 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.71 QSlider* Ui_MPerfWidget::ySlider2

Definition at line 72 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.72 QSlider* Ui_MPerfWidget::ySlider3

Definition at line 45 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.73 QSlider* Ui_MPerfWidget::ySlider4

Definition at line 78 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

12.100.3.74 QSlider* Ui_MPerfWidget::ySlider5

Definition at line 85 of file ui_mperfwidget.h.

Referenced by MPerfWidget::MPerfWidget(), and setupUi().

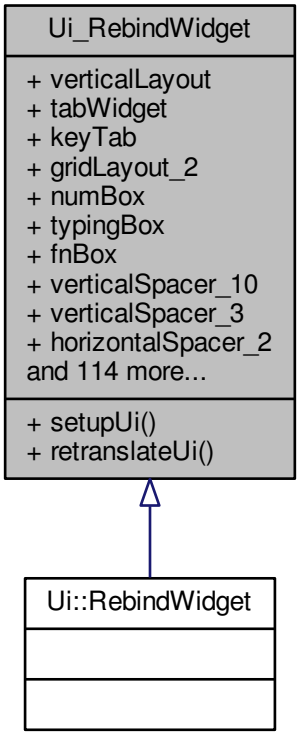
The documentation for this class was generated from the following file:

- src/ckb/ui_mperfwidget.h

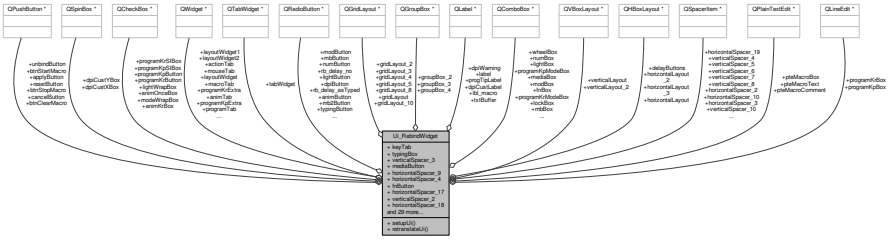
12.101 Ui_RebindWidget Class Reference

`#include <src/ckb/ui_rebindwidget.h>`

Inheritance diagram for Ui_RebindWidget:



Collaboration diagram for Ui_RebindWidget:



Public Member Functions

- void setupUi (QWidget ∗RebindWidget)
- void retranslateUi (QWidget ∗RebindWidget)

Data Fields

- QVBoxLayout * verticalLayout
- QTabWidget * tabWidget
- QWidget * keyTab
- QGridLayout * gridLayout_2
- QComboBox * numBox
- QComboBox * typingBox
- QComboBox * fnBox
- QSpacerItem * verticalSpacer_10
- QSpacerItem * verticalSpacer_3
- QSpacerItem * horizontalSpacer_2
- QRadioButton * typingButton
- QComboBox * modBox
- QSpacerItem * horizontalSpacer_3
- QRadioButton * mediaButton
- QSpacerItem * verticalSpacer_4
- QSpacerItem * horizontalSpacer_9
- QSpacerItem * horizontalSpacer_4
- QRadioButton * modButton
- QRadioButton * fnButton
- QComboBox * mediaBox
- QRadioButton * numButton
- QSpacerItem * horizontalSpacer_17
- QWidget * mouseTab
- QGridLayout * gridLayout
- QRadioButton * dpiButton
- QSpacerItem * verticalSpacer_2
- QSpacerItem * verticalSpacer_8
- QSpacerItem * horizontalSpacer_18
- QSpacerItem * horizontalSpacer_20
- QComboBox * mb2Box
- QComboBox * dpiBox
- QSpinBox * dpiCustYBox
- QComboBox * mbBox
- QComboBox * wheelBox
- QRadioButton * mbButton
- QSpinBox * dpiCustXBox
- QLabel * dpiWarning
- QRadioButton * wheelButton
- QSpacerItem * horizontalSpacer_22
- QRadioButton * mb2Button
- QLabel * dpiCustLabel
- QSpacerItem * horizontalSpacer_19
- QSpacerItem * horizontalSpacer_21
- QWidget * animTab
- QGridLayout * gridLayout_5
- QSpacerItem * verticalSpacer_11
- QComboBox * animBox
- QSpacerItem * horizontalSpacer_24
- QSpacerItem * horizontalSpacer_25
- QCheckBox * animKrBox
- QCheckBox * animOnceBox
- QSpacerItem * horizontalSpacer_26
- QRadioButton * animButton

- QSpacerItem * horizontalSpacer_23
- QWidget * actionTab
- QGridLayout * gridLayout_3
- QSpacerItem * horizontalSpacer_5
- QRadioButton * modeButton
- QCheckBox * modeWrapBox
- QCheckBox * lightWrapBox
- QSpacerItem * verticalSpacer_7
- QSpacerItem * verticalSpacer_6
- QRadioButton * lockButton
- QSpacerItem * verticalSpacer_5
- QComboBox * lightBox
- QComboBox * lockBox
- QRadioButton * lightButton
- QComboBox * modeBox
- QSpacerItem * horizontalSpacer_8
- QSpacerItem * horizontalSpacer_10
- QSpacerItem * horizontalSpacer_7
- QSpacerItem * horizontalSpacer_6
- QWidget * programTab
- QGridLayout * gridLayout_4
- QSpacerItem * verticalSpacer
- QSpacerItem * horizontalSpacer
- QCheckBox * programKpButton
- QLineEdit * programKpBox
- QSpacerItem * horizontalSpacer_11
- QSpacerItem * horizontalSpacer_13
- QLineEdit * programKrBox
- QCheckBox * programKrButton
- QSpacerItem * horizontalSpacer_12
- QWidget * programKpExtra
- QHBoxLayout * horizontalLayout_2
- QCheckBox * programKpSIBox
- QComboBox * programKpModeBox
- QSpacerItem * horizontalSpacer_15
- QWidget * programKrExtra
- QHBoxLayout * horizontalLayout_3
- QCheckBox * programKrSIBox
- QComboBox * programKrModeBox
- QSpacerItem * horizontalSpacer_16
- QLabel * progTipLabel
- QWidget * macroTab
- QWidget * layoutWidget1
- QGridLayout * gridLayout_8
- QGroupBox * groupBox_2
- QPlainTextEdit * pteMacroBox
- QPlainTextEdit * pteMacroComment
- QPlainTextEdit * pteMacroText
- QWidget * layoutWidget
- QVBoxLayout * verticalLayout_2
- QGroupBox * groupBox_3
- QGroupBox * groupBox_4
- QWidget * layoutWidget_2
- QGridLayout * gridLayout_10
- QPushButton * btnStopMacro

- QPushButton * btnClearMacro
- QPushButton * btnStartMacro
- QLabel * label
- QLabel * lbl_macro
- QWidget * layoutWidget2
- QHBoxLayout * delayButtons
- QRadioButton * rb_delay_no
- QRadioButton * rb_delay_default
- QRadioButton * rb_delay_asTyped
- QLabel * txtBuffer
- QHBoxLayout * horizontalLayout
- QSpacerItem * horizontalSpacer_14
- QPushButton * unbindButton
- QPushButton * resetButton
- QPushButton * cancelButton
- QPushButton * applyButton

12.101.1 Detailed Description

Definition at line 35 of file ui_rebindwidget.h.

12.101.2 Member Function Documentation

12.101.2.1 void Ui_RebindWidget::retranslateUi (QWidget * *RebindWidget*) [inline]

Definition at line 901 of file ui_rebindwidget.h.

References actionTab, animBox, animButton, animKrBox, animOnceBox, animTab, applyButton, btnClearMacro, btnStartMacro, btnStopMacro, cancelButton, dpiBox, dpiButton, dpiCustLabel, dpiWarning, fnBox, fnButton, groupBox_2, groupBox_3, groupBox_4, keyTab, label, lbl_macro, lightBox, lightButton, lightWrapBox, lockBox, lockButton, macroTab, mb2Box, mb2Button, mbBox, mbButton, mediaBox, mediaButton, modBox, modButton, modeBox, modeButton, modeWrapBox, mouseTab, numBox, numButton, programKpButton, programKpModeBox, programKpSlBox, programKrButton, programKrModeBox, programKrSlBox, programTab, progTipLabel, pteMacroComment, pteMacroText, rb_delay_asTyped, rb_delay_default, rb_delay_no, resetButton, tabWidget, txtBuffer, typingBox, typingButton, unbindButton, wheelBox, and wheelButton.

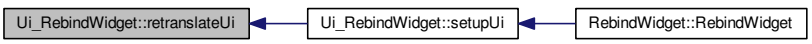
Referenced by setupUi().

```
902     {
903         numBox->clear();
904         numBox->insertItems(0, QStringList()
905          << QString()
906         );
907         typingBox->clear();
908         typingBox->insertItems(0, QStringList()
909          << QString()
910         );
911         fnBox->clear();
912         fnBox->insertItems(0, QStringList()
913          << QString()
914         );
915         typingButton->setText(QApplication::translate("RebindWidget", "Typing:", Q_NULLPTR));
916         modBox->clear();
917         modBox->insertItems(0, QStringList()
918          << QString()
919         );
920         mediaButton->setText(QApplication::translate("RebindWidget", "Media:", Q_NULLPTR));
921         modButton->setText(QApplication::translate("RebindWidget", "Modifier:", Q_NULLPTR));
922         fnButton->setText(QApplication::translate("RebindWidget", "Miscellaneous:", Q_NULLPTR));
923         mediaBox->clear();
924         mediaBox->insertItems(0, QStringList()
925          << QString()
926         );
927         numButton->setText(QApplication::translate("RebindWidget", "Number pad:", Q_NULLPTR));
928         tabWidget->setTabText(tabWidget->indexOf(keyTab), QApplication::translate("
```

```
      RebindWidget", "Keyboard", Q_NULLPTR));
929         dpiButton->setText(QApplication::translate("RebindWidget", "Change DPI:", Q_NULLPTR));
930         mb2Box->clear();
931         mb2Box->insertItems(0, QStringList()
932          << QString()
933          << QApplication::translate("RebindWidget", "4 (Back)", Q_NULLPTR)
934          << QApplication::translate("RebindWidget", "5 (Forward)", Q_NULLPTR)
935          << QApplication::translate("RebindWidget", "6", Q_NULLPTR)
936          << QApplication::translate("RebindWidget", "7", Q_NULLPTR)
937          << QApplication::translate("RebindWidget", "8", Q_NULLPTR)
938         );
939         dpiBox->clear();
940         dpiBox->insertItems(0, QStringList()
941          << QString()
942          << QApplication::translate("RebindWidget", "(Cycle Up)", Q_NULLPTR)
943          << QApplication::translate("RebindWidget", "(Cycle Down)", Q_NULLPTR)
944          << QApplication::translate("RebindWidget", "(Up)", Q_NULLPTR)
945          << QApplication::translate("RebindWidget", "(Down)", Q_NULLPTR)
946          << QApplication::translate("RebindWidget", "Sniper", Q_NULLPTR)
947          << QApplication::translate("RebindWidget", "1:", Q_NULLPTR)
948          << QApplication::translate("RebindWidget", "2:", Q_NULLPTR)
949          << QApplication::translate("RebindWidget", "3:", Q_NULLPTR)
950          << QApplication::translate("RebindWidget", "4:", Q_NULLPTR)
951          << QApplication::translate("RebindWidget", "5:", Q_NULLPTR)
952          << QApplication::translate("RebindWidget", "Custom:", Q_NULLPTR)
953         );
954         mbBox->clear();
955         mbBox->insertItems(0, QStringList()
956          << QString()
957          << QApplication::translate("RebindWidget", "Left", Q_NULLPTR)
958          << QApplication::translate("RebindWidget", "Right", Q_NULLPTR)
959          << QApplication::translate("RebindWidget", "Middle", Q_NULLPTR)
960         );
961         wheelBox->clear();
962         wheelBox->insertItems(0, QStringList()
963          << QString()
964          << QApplication::translate("RebindWidget", "Up", Q_NULLPTR)
965          << QApplication::translate("RebindWidget", "Down", Q_NULLPTR)
966         );
967         mbButton->setText(QApplication::translate("RebindWidget", "Standard button:", Q_NULLPTR));
968         dpiWarning->setText(QApplication::translate("RebindWidget", "DPI is only available for
       mice.", Q_NULLPTR));
969         wheelButton->setText(QApplication::translate("RebindWidget", "Wheel:", Q_NULLPTR));
970         mb2Button->setText(QApplication::translate("RebindWidget", "Special button:", Q_NULLPTR));
971         dpiCustLabel->setText(QApplication::translate("RebindWidget", "x", Q_NULLPTR));
972         tabWidget->setTabText(tabWidget->indexOf(mouseTab),
      QApplication::translate("RebindWidget", "Mouse", Q_NULLPTR));
973         animBox->clear();
974         animBox->insertItems(0, QStringList()
975          << QString()
976         );
977         animKrBox->setText(QApplication::translate("RebindWidget", "Stop on key release",
      Q_NULLPTR));
978         animOnceBox->setText(QApplication::translate("RebindWidget", "Restart every time",
      Q_NULLPTR));
979         animButton->setText(QApplication::translate("RebindWidget", "Start animation on keypress:
      ", Q_NULLPTR));
980         tabWidget->setTabText(tabWidget->indexOf(animTab), QApplication::translate
      ("RebindWidget", "Animation", Q_NULLPTR));
981         modeButton->setText(QApplication::translate("RebindWidget", "Switch to mode:", Q_NULLPTR)
      );
982         modeWrapBox->setText(QApplication::translate("RebindWidget", "Wrap-around", Q_NULLPTR));
983         lightWrapBox->setText(QApplication::translate("RebindWidget", "Wrap-around", Q_NULLPTR)
      );
984         lockButton->setText(QApplication::translate("RebindWidget", "Windows lock:", Q_NULLPTR));
985         lightBox->clear();
986         lightBox->insertItems(0, QStringList()
987          << QString()
988          << QApplication::translate("RebindWidget", "Up", Q_NULLPTR)
989          << QApplication::translate("RebindWidget", "Down", Q_NULLPTR)
990         );
991         lockBox->clear();
992         lockBox->insertItems(0, QStringList()
993          << QString()
994          << QApplication::translate("RebindWidget", "Toggle", Q_NULLPTR)
995          << QApplication::translate("RebindWidget", "On", Q_NULLPTR)
996          << QApplication::translate("RebindWidget", "Off", Q_NULLPTR)
997         );
998         lightButton->setText(QApplication::translate("RebindWidget", "Brightness:", Q_NULLPTR));
999         modeBox->clear();
1000         modeBox->insertItems(0, QStringList()
1001          << QString()
1002          << QApplication::translate("RebindWidget", "(Previous)", Q_NULLPTR)
1003          << QApplication::translate("RebindWidget", "(Next)", Q_NULLPTR)
1004         );
1005         tabWidget->setTabText(tabWidget->indexOf(actionTab),
      QApplication::translate("RebindWidget", "Special", Q_NULLPTR));
```

```
1006         programKpButton->setText(QApplication::translate("RebindWidget", "Launch program on
      key press:", Q_NULLPTR));
1007         programKrButton->setText(QApplication::translate("RebindWidget", "Launch program on
      key release:", Q_NULLPTR));
1008         programKpSIBox->setText(QApplication::translate("RebindWidget", "Single instance",
     Q_NULLPTR));
1009         programKpModeBox->clear();
1010         programKpModeBox->insertItems(0, QStringList()
1011          << QApplication::translate("RebindWidget", "Run indefinitely", Q_NULLPTR)
1012          << QApplication::translate("RebindWidget", "Stop on release", Q_NULLPTR)
1013          << QApplication::translate("RebindWidget", "Stop on re-press", Q_NULLPTR)
1014         );
1015         programKrSIBox->setText(QApplication::translate("RebindWidget", "Single instance",
     Q_NULLPTR));
1016         programKrModeBox->clear();
1017         programKrModeBox->insertItems(0, QStringList()
1018          << QApplication::translate("RebindWidget", "Run indefinitely", Q_NULLPTR)
1019          << QApplication::translate("RebindWidget", "Stop on re-press", Q_NULLPTR)
1020         );
1021         progTipLabel->setText(QApplication::translate("RebindWidget", "Tip", Q_NULLPTR));
1022         tabWidget->setTabText(tabWidget->indexOf(programTab),
     QApplication::translate("RebindWidget", "Program", Q_NULLPTR));
1023         groupBox_2->setTitle(QApplication::translate("RebindWidget", "Macro Key Actions",
     Q_NULLPTR));
1024 #ifndef QT_NO_TOOLTIP
1025         pteMacroComment->setToolTip(QApplication::translate("RebindWidget", "What is this
      macro for?", Q_NULLPTR));
1026 #endif // QT_NO_TOOLTIP
1027         pteMacroComment->setDocumentTitle(QString());
1028         pteMacroComment->setPlainText(QString());
1029         pteMacroText->setDocumentTitle(QString());
1030         pteMacroText->setPlainText(QString());
1031         groupBox_3->setTitle(QApplication::translate("RebindWidget", "Macro Comment", Q_NULLPTR))
     ;
1032         groupBox_4->setTitle(QApplication::translate("RebindWidget", "Macro Text", Q_NULLPTR));
1033         btnStopMacro->setText(QApplication::translate("RebindWidget", "Stop", Q_NULLPTR));
1034         btnClearMacro->setText(QApplication::translate("RebindWidget", "Clear", Q_NULLPTR));
1035         btnStartMacro->setText(QApplication::translate("RebindWidget", "Start", Q_NULLPTR));
1036         label->setText(QApplication::translate("RebindWidget", "Record macro ", Q_NULLPTR));
1037         lbl_macro->setText(QApplication::translate("RebindWidget", "Comment label for help",
     Q_NULLPTR));
1038 #ifndef QT_NO_WHATSTHIS
1039         rb_delay_no->setWhatsThis(QApplication::translate("RebindWidget", "configure delays
      between keystrokes: Disable delay", Q_NULLPTR));
1040 #endif // QT_NO_WHATSTHIS
1041         rb_delay_no->setText(QApplication::translate("RebindWidget", "No delay", Q_NULLPTR));
1042 #ifndef QT_NO_WHATSTHIS
1043         rb_delay_default->setWhatsThis(QApplication::translate("RebindWidget", "configure
      delays between keystrokes: Set delay to default values: 20us up to 15 chars, 200us above", Q_NULLPTR));
1044 #endif // QT_NO_WHATSTHIS
1045         rb_delay_default->setText(QApplication::translate("RebindWidget", "default",
     Q_NULLPTR));
1046 #ifndef QT_NO_WHATSTHIS
1047         rb_delay_asTyped->setWhatsThis(QApplication::translate("RebindWidget", "configure
      delays between keystrokes: Remember delays as they had been typed while the macro definition process",
     Q_NULLPTR));
1048 #endif // QT_NO_WHATSTHIS
1049         rb_delay_asTyped->setText(QApplication::translate("RebindWidget", "as typed",
     Q_NULLPTR));
1050         txtBuffer->setText(QString());
1051         tabWidget->setTabText(tabWidget->indexOf(macroTab),
     QApplication::translate("RebindWidget", "Macro", Q_NULLPTR));
1052         unbindButton->setText(QApplication::translate("RebindWidget", "Unbind", Q_NULLPTR));
1053         resetButton->setText(QApplication::translate("RebindWidget", "Reset to Default",
     Q_NULLPTR));
1054         cancelButton->setText(QApplication::translate("RebindWidget", "Cancel", Q_NULLPTR));
1055         applyButton->setText(QApplication::translate("RebindWidget", "Apply", Q_NULLPTR));
1056         Q_UNUSED(RebindWidget);
1057     } // retranslateUi
```

Here is the caller graph for this function:



12.101.2.2 void Ui_RebindWidget::setupUi (QWidget * *RebindWidget*) [inline]

Definition at line 163 of file ui_rebindwidget.h.

References actionTab, animBox, animButton, animKrBox, animOnceBox, animTab, applyButton, btnClearMacro, btnStartMacro, btnStopMacro, cancelButton, delayButtons, dpiBox, dpiButton, dpiCustLabel, dpiCustXBox, dpiCustYBox, dpiWarning, fnBox, fnButton, gridLayout, gridLayout_10, gridLayout_2, gridLayout_3, gridLayout_4, gridLayout_5, gridLayout_8, groupBox_2, groupBox_3, groupBox_4, horizontalLayout, horizontalLayout_2, horizontalLayout_3, horizontalSpacer, horizontalSpacer_10, horizontalSpacer_11, horizontalSpacer_12, horizontalSpacer_13, horizontalSpacer_14, horizontalSpacer_15, horizontalSpacer_16, horizontalSpacer_17, horizontalSpacer_18, horizontalSpacer_19, horizontalSpacer_2, horizontalSpacer_20, horizontalSpacer_21, horizontalSpacer_22, horizontalSpacer_23, horizontalSpacer_24, horizontalSpacer_25, horizontalSpacer_26, horizontalSpacer_3, horizontalSpacer_4, horizontalSpacer_5, horizontalSpacer_6, horizontalSpacer_7, horizontalSpacer_8, horizontalSpacer_9, keyTab, label, layoutWidget, layoutWidget1, layoutWidget2, layoutWidget_2, lbl_macro, lightBox, lightButton, lightWrapBox, lockBox, lockButton, macroTab, mb2Box, mb2Button, mbBox, mbButton, mediaBox, mediaButton, modBox, modButton, modeBox, modeButton, modeWrapBox, mouseTab, numBox, numButton, programKpBox, programKpButton, programKpExtra, programKpModeBox, programKpSIBox, programKrBox, programKrButton, programKrExtra, programKrModeBox, programKrSIBox, programTab, progTipLabel, pteMacroBox, pteMacroComment, pteMacroText, rb_delay_asTyped, rb_delay_default, rb_delay_no, resetButton, retranslateUi(), tabWidget, txtBuffer, typingBox, typingButton, unbindButton, verticalLayout, verticalLayout_2, verticalSpacer, verticalSpacer_10, verticalSpacer_11, verticalSpacer_2, verticalSpacer_3, verticalSpacer_4, verticalSpacer_5, verticalSpacer_6, verticalSpacer_7, verticalSpacer_8, wheelBox, and wheelButton.

Referenced by RebindWidget::RebindWidget().

```
164     {
165         if (RebindWidget->objectName().isEmpty())
166             RebindWidget->setObjectName(QStringLiteral("RebindWidget"));
167         RebindWidget->resize(687, 342);
168         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
169         sizePolicy.setHorizontalStretch(0);
170         sizePolicy.setVerticalStretch(0);
171         sizePolicy.setHeightForWidth(RebindWidget->sizePolicy().hasHeightForWidth());
172         RebindWidget->setSizePolicy(sizePolicy);
173         RebindWidget->setStyleSheet(QStringLiteral("font-size: 12px ;"));
174         verticalLayout = new QVBoxLayout(RebindWidget);
175         verticalLayout->setObjectName(QStringLiteral("verticalLayout"));
176         tabWidget = new QTabWidget(RebindWidget);
177         tabWidget->setObjectName(QStringLiteral("tabWidget"));
178         tabWidget->setStyleSheet(QStringLiteral(""));
179         tabWidget->setUsesScrollButtons(false);
180         tabWidget->setDocumentMode(false);
181         keyTab = new QWidget();
182         keyTab->setObjectName(QStringLiteral("keyTab"));
183         sizePolicy.setHeightForWidth(keyTab->sizePolicy().hasHeightForWidth());
184         keyTab->setSizePolicy(sizePolicy);
185         gridLayout_2 = new QGridLayout(keyTab);
186         gridLayout_2->setObjectName(QStringLiteral("gridLayout_2"));
187         gridLayout_2->setContentsMargins(-1, -1, -1, 0);
188         numBox = new QComboBox(keyTab);
189         numBox->setObjectName(QStringLiteral("numBox"));
190         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Fixed);
191         sizePolicy1.setHorizontalStretch(0);
192         sizePolicy1.setVerticalStretch(0);
193         sizePolicy1.setHeightForWidth(numBox->sizePolicy().hasHeightForWidth());
194         numBox->setSizePolicy(sizePolicy1);
195 
196         gridLayout_2->addWidget(numBox, 6, 4, 1, 1);
197 
198         typingBox = new QComboBox(keyTab);
199         typingBox->setObjectName(QStringLiteral("typingBox"));
200         sizePolicy1.setHeightForWidth(typingBox->sizePolicy().hasHeightForWidth());
201         typingBox->setSizePolicy(sizePolicy1);
202 
203         gridLayout_2->addWidget(typingBox, 1, 1, 1, 1);
204 
205         fnBox = new QComboBox(keyTab);
206         fnBox->setObjectName(QStringLiteral("fnBox"));
207         sizePolicy1.setHeightForWidth(fnBox->sizePolicy().hasHeightForWidth());
208         fnBox->setSizePolicy(sizePolicy1);
209 
210         gridLayout_2->addWidget(fnBox, 6, 1, 1, 1);
211 
212         verticalSpacer_10 = new QSpacerItem(0, 5, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
213 
214         gridLayout_2->addItem(verticalSpacer_10, 9, 1, 1, 1);
```

```
215
216         verticalSpacer_3 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
217
218         gridLayout_2->addItem(verticalSpacer_3, 15, 2, 1, 1);
219
220         horizontalSpacer_2 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
221
222         gridLayout_2->addItem(horizontalSpacer_2, 1, 3, 1, 1);
223
224         typingButton = new QRadioButton(keyTab);
225         typingButton->setObjectName(QStringLiteral("typingButton"));
226         typingButton->setAutoExclusive(false);
227
228         gridLayout_2->addWidget(typingButton, 0, 0, 1, 2);
229
230         modBox = new QComboBox(keyTab);
231         modBox->setObjectName(QStringLiteral("modBox"));
232         sizePolicy1.setHeightForWidth(modBox->sizePolicy().hasHeightForWidth());
233         modBox->setSizePolicy(sizePolicy1);
234
235         gridLayout_2->addWidget(modBox, 1, 4, 1, 1);
236
237         horizontalSpacer_3 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
238
239         gridLayout_2->addItem(horizontalSpacer_3, 1, 0, 1, 1);
240
241         mediaButton = new QRadioButton(keyTab);
242         mediaButton->setObjectName(QStringLiteral("mediaButton"));
243         mediaButton->setAutoExclusive(false);
244
245         gridLayout_2->addWidget(mediaButton, 10, 0, 1, 2);
246
247         verticalSpacer_4 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
248
249         gridLayout_2->addItem(verticalSpacer_4, 2, 1, 1, 1);
250
251         horizontalSpacer_9 = new QSpacerItem(0, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
252
253         gridLayout_2->addItem(horizontalSpacer_9, 1, 5, 1, 2);
254
255         horizontalSpacer_4 = new QSpacerItem(0, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
256
257         gridLayout_2->addItem(horizontalSpacer_4, 1, 2, 1, 1);
258
259         modButton = new QRadioButton(keyTab);
260         modButton->setObjectName(QStringLiteral("modButton"));
261
262         gridLayout_2->addWidget(modButton, 0, 3, 1, 2);
263
264         fnButton = new QRadioButton(keyTab);
265         fnButton->setObjectName(QStringLiteral("fnButton"));
266         fnButton->setAutoExclusive(false);
267
268         gridLayout_2->addWidget(fnButton, 5, 0, 1, 2);
269
270         mediaBox = new QComboBox(keyTab);
271         mediaBox->setObjectName(QStringLiteral("mediaBox"));
272         sizePolicy1.setHeightForWidth(mediaBox->sizePolicy().hasHeightForWidth());
273         mediaBox->setSizePolicy(sizePolicy1);
274
275         gridLayout_2->addWidget(mediaBox, 11, 1, 1, 1);
276
277         numButton = new QRadioButton(keyTab);
278         numButton->setObjectName(QStringLiteral("numButton"));
279         numButton->setAutoExclusive(false);
280
281         gridLayout_2->addWidget(numButton, 5, 3, 1, 2);
282
283         horizontalSpacer_17 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
284
285         gridLayout_2->addItem(horizontalSpacer_17, 1, 7, 1, 1);
286
287         tabWidget->addTab(keyTab, QString());
288         mouseTab = new QWidget();
289         mouseTab->setObjectName(QStringLiteral("mouseTab"));
290         gridLayout = new QGridLayout(mouseTab);
291         gridLayout->setObjectName(QStringLiteral("gridLayout"));
292         dpiButton = new QRadioButton(mouseTab);
293         dpiButton->setObjectName(QStringLiteral("dpiButton"));
294         dpiButton->setAutoExclusive(false);
```

```
295
296         gridLayout->addWidget(dpiButton, 3, 3, 1, 5);
297
298         verticalSpacer_2 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
299
300         gridLayout->addItem(verticalSpacer_2, 5, 1, 1, 1);
301
302         verticalSpacer_8 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
303
304         gridLayout->addItem(verticalSpacer_8, 2, 1, 1, 1);
305
306         horizontalSpacer_18 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
307
308         gridLayout->addItem(horizontalSpacer_18, 1, 0, 1, 1);
309
310         horizontalSpacer_20 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
311
312         gridLayout->addItem(horizontalSpacer_20, 1, 3, 1, 1);
313
314         mb2Box = new QComboBox(mouseTab);
315         mb2Box->setObjectName(QStringLiteral("mb2Box"));
316         sizePolicy1.setHeightForWidth(mb2Box->sizePolicy().hasHeightForWidth());
317         mb2Box->setSizePolicy(sizePolicy1);
318
319         gridLayout->addWidget(mb2Box, 1, 4, 1, 4);
320
321         dpiBox = new QComboBox(mouseTab);
322         dpiBox->setObjectName(QStringLiteral("dpiBox"));
323
324         gridLayout->addWidget(dpiBox, 4, 4, 1, 1);
325
326         dpiCustYBox = new QSpinBox(mouseTab);
327         dpiCustYBox->setObjectName(QStringLiteral("dpiCustYBox"));
328         dpiCustYBox->setMinimum(400);
329         dpiCustYBox->setMaximum(8500);
330         dpiCustYBox->setSingleStep(50);
331
332         gridLayout->addWidget(dpiCustYBox, 4, 7, 1, 1);
333
334         mbBox = new QComboBox(mouseTab);
335         mbBox->setObjectName(QStringLiteral("mbBox"));
336         sizePolicy1.setHeightForWidth(mbBox->sizePolicy().hasHeightForWidth());
337         mbBox->setSizePolicy(sizePolicy1);
338
339         gridLayout->addWidget(mbBox, 1, 1, 1, 1);
340
341         wheelBox = new QComboBox(mouseTab);
342         wheelBox->setObjectName(QStringLiteral("wheelBox"));
343
344         gridLayout->addWidget(wheelBox, 4, 1, 1, 1);
345
346         mbButton = new QRadioButton(mouseTab);
347         mbButton->setObjectName(QStringLiteral("mbButton"));
348         mbButton->setAutoExclusive(false);
349
350         gridLayout->addWidget(mbButton, 0, 0, 1, 2);
351
352         dpiCustXBox = new QSpinBox(mouseTab);
353         dpiCustXBox->setObjectName(QStringLiteral("dpiCustXBox"));
354         dpiCustXBox->setMinimum(400);
355         dpiCustXBox->setMaximum(8500);
356         dpiCustXBox->setSingleStep(50);
357
358         gridLayout->addWidget(dpiCustXBox, 4, 5, 1, 1);
359
360         dpiWarning = new QLabel(mouseTab);
361         dpiWarning->setObjectName(QStringLiteral("dpiWarning"));
362         QSizePolicy sizePolicy2(QSizePolicy::Preferred, QSizePolicy::Fixed);
363         sizePolicy2.setHorizontalStretch(0);
364         sizePolicy2.setVerticalStretch(0);
365         sizePolicy2.setHeightForWidth(dpiWarning->sizePolicy().hasHeightForWidth());
366         dpiWarning->setSizePolicy(sizePolicy2);
367
368         gridLayout->addWidget(dpiWarning, 6, 0, 1, 10);
369
370         wheelButton = new QRadioButton(mouseTab);
371         wheelButton->setObjectName(QStringLiteral("wheelButton"));
372
373         gridLayout->addWidget(wheelButton, 3, 0, 1, 2);
374
375         horizontalSpacer_22 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
376
```

```
377         gridLayout->addItem(horizontalSpacer_22, 1, 8, 1, 1);
378
379         mb2Button = new QRadioButton(mouseTab);
380         mb2Button->setObjectName(QStringLiteral("mb2Button"));
381         mb2Button->setAutoExclusive(false);
382
383         gridLayout->addWidget(mb2Button, 0, 3, 1, 5);
384
385         dpiCustLabel = new QLabel(mouseTab);
386         dpiCustLabel->setObjectName(QStringLiteral("dpiCustLabel"));
387
388         gridLayout->addWidget(dpiCustLabel, 4, 6, 1, 1);
389
390         horizontalSpacer_19 = new QSpacerItem(0, 0, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
391
392         gridLayout->addItem(horizontalSpacer_19, 1, 2, 1, 1);
393
394         horizontalSpacer_21 = new QSpacerItem(0, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
395
396         gridLayout->addItem(horizontalSpacer_21, 1, 9, 1, 1);
397
398         tabWidget->addTab(mouseTab, QString());
399         animTab = new QWidget();
400         animTab->setObjectName(QStringLiteral("animTab"));
401         gridLayout_5 = new QGridLayout(animTab);
402         gridLayout_5->setObjectName(QStringLiteral("gridLayout_5"));
403         verticalSpacer_11 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
404
405         gridLayout_5->addItem(verticalSpacer_11, 6, 2, 1, 1);
406
407         animBox = new QComboBox(animTab);
408         animBox->setObjectName(QStringLiteral("animBox"));
409         sizePolicy1.setHeightForWidth(animBox->sizePolicy().hasHeightForWidth());
410         animBox->setSizePolicy(sizePolicy1);
411
412         gridLayout_5->addWidget(animBox, 4, 2, 1, 1);
413
414         horizontalSpacer_24 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
415
416         gridLayout_5->addItem(horizontalSpacer_24, 4, 3, 1, 1);
417
418         horizontalSpacer_25 = new QSpacerItem(438, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
419
420         gridLayout_5->addItem(horizontalSpacer_25, 4, 7, 1, 1);
421
422         animKrBox = new QCheckBox(animTab);
423         animKrBox->setObjectName(QStringLiteral("animKrBox"));
424
425         gridLayout_5->addWidget(animKrBox, 4, 6, 1, 1);
426
427         animOnceBox = new QCheckBox(animTab);
428         animOnceBox->setObjectName(QStringLiteral("animOnceBox"));
429
430         gridLayout_5->addWidget(animOnceBox, 4, 4, 1, 1);
431
432         horizontalSpacer_26 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
433
434         gridLayout_5->addItem(horizontalSpacer_26, 4, 5, 1, 1);
435
436         animButton = new QRadioButton(animTab);
437         animButton->setObjectName(QStringLiteral("animButton"));
438         sizePolicy1.setHeightForWidth(animButton->sizePolicy().hasHeightForWidth());
439         animButton->setSizePolicy(sizePolicy1);
440
441         gridLayout_5->addWidget(animButton, 0, 1, 1, 2);
442
443         horizontalSpacer_23 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
444
445         gridLayout_5->addItem(horizontalSpacer_23, 4, 1, 1, 1);
446
447         tabWidget->addTab(animTab, QString());
448         actionTab = new QWidget();
449         actionTab->setObjectName(QStringLiteral("actionTab"));
450         sizePolicy.setHeightForWidth(actionTab->sizePolicy().hasHeightForWidth());
451         actionTab->setSizePolicy(sizePolicy);
452         gridLayout_3 = new QGridLayout(actionTab);
453         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
454         gridLayout_3->setContentsMargins(-1, -1, -1, 0);
455         horizontalSpacer_5 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
```

```
456
457         gridLayout_3->addItem(horizontalSpacer_5, 1, 0, 1, 1);
458
459         modeButton = new QRadioButton(actionTab);
460         modeButton->setObjectName(QStringLiteral("modeButton"));
461         modeButton->setAutoExclusive(false);
462
463         gridLayout_3->addWidget(modeButton, 0, 0, 1, 2);
464
465         modeWrapBox = new QCheckBox(actionTab);
466         modeWrapBox->setObjectName(QStringLiteral("modeWrapBox"));
467         modeWrapBox->setChecked(true);
468
469         gridLayout_3->addWidget(modeWrapBox, 1, 3, 1, 1);
470
471         lightWrapBox = new QCheckBox(actionTab);
472         lightWrapBox->setObjectName(QStringLiteral("lightWrapBox"));
473         lightWrapBox->setChecked(true);
474
475         gridLayout_3->addWidget(lightWrapBox, 4, 3, 1, 1);
476
477         verticalSpacer_7 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
478
479         gridLayout_3->addItem(verticalSpacer_7, 2, 1, 1, 1);
480
481         verticalSpacer_6 = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
482
483         gridLayout_3->addItem(verticalSpacer_6, 5, 1, 1, 1);
484
485         lockButton = new QRadioButton(actionTab);
486         lockButton->setObjectName(QStringLiteral("lockButton"));
487         lockButton->setAutoExclusive(false);
488
489         gridLayout_3->addWidget(lockButton, 6, 0, 1, 2);
490
491         verticalSpacer_5 = new QSpacerItem(20, 40, QSizePolicy::Minimum,
    QSizePolicy::Expanding);
492
493         gridLayout_3->addItem(verticalSpacer_5, 8, 1, 1, 1);
494
495         lightBox = new QComboBox(actionTab);
496         lightBox->setObjectName(QStringLiteral("lightBox"));
497         sizePolicy1.setHeightForWidth(lightBox->sizePolicy().hasHeightForWidth());
498         lightBox->setSizePolicy(sizePolicy1);
499
500         gridLayout_3->addWidget(lightBox, 4, 1, 1, 1);
501
502         lockBox = new QComboBox(actionTab);
503         lockBox->setObjectName(QStringLiteral("lockBox"));
504         sizePolicy1.setHeightForWidth(lockBox->sizePolicy().hasHeightForWidth());
505         lockBox->setSizePolicy(sizePolicy1);
506
507         gridLayout_3->addWidget(lockBox, 7, 1, 1, 1);
508
509         lightButton = new QRadioButton(actionTab);
510         lightButton->setObjectName(QStringLiteral("lightButton"));
511         lightButton->setAutoExclusive(false);
512
513         gridLayout_3->addWidget(lightButton, 3, 0, 1, 2);
514
515         modeBox = new QComboBox(actionTab);
516         modeBox->setObjectName(QStringLiteral("modeBox"));
517         sizePolicy1.setHeightForWidth(modeBox->sizePolicy().hasHeightForWidth());
518         modeBox->setSizePolicy(sizePolicy1);
519
520         gridLayout_3->addWidget(modeBox, 1, 1, 1, 1);
521
522         horizontalSpacer_8 = new QSpacerItem(5, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
523
524         gridLayout_3->addItem(horizontalSpacer_8, 1, 2, 1, 1);
525
526         horizontalSpacer_10 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
527
528         gridLayout_3->addItem(horizontalSpacer_10, 7, 3, 1, 1);
529
530         horizontalSpacer_7 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
531
532         gridLayout_3->addItem(horizontalSpacer_7, 1, 4, 1, 1);
533
534         horizontalSpacer_6 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
535
```

```
536         gridLayout_3->addItem(horizontalSpacer_6, 1, 5, 1, 1);
537 
538         tabWidget->addTab(actionTab, QString());
539         programTab = new QWidget();
540         programTab->setObjectName(QStringLiteral("programTab"));
541         gridLayout_4 = new QGridLayout(programTab);
542         gridLayout_4->setObjectName(QStringLiteral("gridLayout_4"));
543         verticalSpacer = new QSpacerItem(20, 5, QSizePolicy::Minimum, QSizePolicy::Fixed);
544 
545         gridLayout_4->addItem(verticalSpacer, 2, 1, 1, 1);
546 
547         horizontalSpacer = new QSpacerItem(5, 20, QSizePolicy::Fixed, QSizePolicy::Minimum)
    ;
548 
549         gridLayout_4->addItem(horizontalSpacer, 1, 0, 1, 1);
550 
551         programKpButton = new QCheckBox(programTab);
552         programKpButton->setObjectName(QStringLiteral("programKpButton"));
553 
554         gridLayout_4->addWidget(programKpButton, 0, 0, 1, 2);
555 
556         programKpBox = new QLineEdit(programTab);
557         programKpBox->setObjectName(QStringLiteral("programKpBox"));
558 
559         gridLayout_4->addWidget(programKpBox, 1, 1, 1, 1);
560 
561         horizontalSpacer_11 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
562 
563         gridLayout_4->addItem(horizontalSpacer_11, 0, 2, 1, 2);
564 
565         horizontalSpacer_13 = new QSpacerItem(20, 20, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
566 
567         gridLayout_4->addItem(horizontalSpacer_13, 1, 2, 1, 1);
568 
569         programKrBox = new QLineEdit(programTab);
570         programKrBox->setObjectName(QStringLiteral("programKrBox"));
571 
572         gridLayout_4->addWidget(programKrBox, 4, 1, 1, 1);
573 
574         programKrButton = new QCheckBox(programTab);
575         programKrButton->setObjectName(QStringLiteral("programKrButton"));
576 
577         gridLayout_4->addWidget(programKrButton, 3, 0, 1, 2);
578 
579         horizontalSpacer_12 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
580 
581         gridLayout_4->addItem(horizontalSpacer_12, 0, 4, 1, 1);
582 
583         programKpExtra = new QWidget(programTab);
584         programKpExtra->setObjectName(QStringLiteral("programKpExtra"));
585         horizontalLayout_2 = new QHBoxLayout(programKpExtra);
586         horizontalLayout_2->setSpacing(12);
587         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
588         horizontalLayout_2->setContentsMargins(0, 0, 0, 0);
589         programKpSIBox = new QCheckBox(programKpExtra);
590         programKpSIBox->setObjectName(QStringLiteral("programKpSIBox"));
591 
592         horizontalLayout_2->addWidget(programKpSIBox);
593 
594         programKpModeBox = new QComboBox(programKpExtra);
595         programKpModeBox->setObjectName(QStringLiteral("programKpModeBox"));
596 
597         horizontalLayout_2->addWidget(programKpModeBox);
598 
599         horizontalSpacer_15 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
600 
601         horizontalLayout_2->addItem(horizontalSpacer_15);
602 
603 
604         gridLayout_4->addWidget(programKpExtra, 1, 3, 1, 2);
605 
606         programKrExtra = new QWidget(programTab);
607         programKrExtra->setObjectName(QStringLiteral("programKrExtra"));
608         horizontalLayout_3 = new QHBoxLayout(programKrExtra);
609         horizontalLayout_3->setSpacing(12);
610         horizontalLayout_3->setObjectName(QStringLiteral("horizontalLayout_3"));
611         horizontalLayout_3->setContentsMargins(0, 0, 0, 0);
612         programKrSIBox = new QCheckBox(programKrExtra);
613         programKrSIBox->setObjectName(QStringLiteral("programKrSIBox"));
614 
615         horizontalLayout_3->addWidget(programKrSIBox);
616 
617         programKrModeBox = new QComboBox(programKrExtra);
```

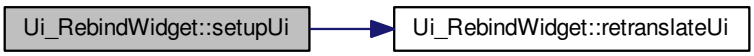
```
618         programKrModeBox->setObjectName(QStringLiteral("programKrModeBox"));
619
620         horizontalLayout_3->addWidget(programKrModeBox);
621
622         horizontalSpacer_16 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
623
624         horizontalLayout_3->addItem(horizontalSpacer_16);
625
626
627         gridLayout_4->addWidget(programKrExtra, 4, 3, 1, 2);
628
629         progTipLabel = new QLabel(programTab);
630         progTipLabel->setObjectName(QStringLiteral("progTipLabel"));
631         QSizePolicy sizePolicy3(QSizePolicy::Preferred, QSizePolicy::Expanding);
632         sizePolicy3.setHorizontalStretch(0);
633         sizePolicy3.setVerticalStretch(0);
634         sizePolicy3.setHeightForWidth(progTipLabel->sizePolicy().hasHeightForWidth());
635         progTipLabel->setSizePolicy(sizePolicy3);
636         progTipLabel->setTextInteractionFlags(Qt::LinksAccessibleByMouse|
    Qt::TextSelectableByMouse);
637
638         gridLayout_4->addWidget(progTipLabel, 5, 1, 1, 4);
639
640         tabWidget->addTab(programTab, QString());
641         macroTab = new QWidget();
642         macroTab->setObjectName(QStringLiteral("macroTab"));
643         layoutWidget1 = new QWidget(macroTab);
644         layoutWidget1->setObjectName(QStringLiteral("layoutWidget1"));
645         layoutWidget1->setGeometry(QRect(40, 12, 241, 161));
646         gridLayout_8 = new QGridLayout(layoutWidget1);
647         gridLayout_8->setObjectName(QStringLiteral("gridLayout_8"));
648         gridLayout_8->setSizeConstraint(QLayout::SetNoConstraint);
649         gridLayout_8->setContentsMargins(0, 0, 0, 0);
650         groupBox_2 = new QGroupBox(layoutWidget1);
651         groupBox_2->setObjectName(QStringLiteral("groupBox_2"));
652         pteMacroBox = new QPlainTextEdit(groupBox_2);
653         pteMacroBox->setObjectName(QStringLiteral("pteMacroBox"));
654         pteMacroBox->setGeometry(QRect(0, 20, 241, 131));
655         pteMacroBox->setFocusPolicy(Qt::WheelFocus);
656         pteMacroBox->setTabChangesFocus(true);
657
658         gridLayout_8->addWidget(groupBox_2, 0, 0, 1, 1);
659
660         pteMacroComment = new QPlainTextEdit(macroTab);
661         pteMacroComment->setObjectName(QStringLiteral("pteMacroComment"));
662         pteMacroComment->setEnabled(true);
663         pteMacroComment->setGeometry(QRect(410, 10, 311, 31));
664         pteMacroComment->setFocusPolicy(Qt::WheelFocus);
665         pteMacroComment->setContextMenuPolicy(Qt::NoContextMenu);
666         pteMacroComment->setAcceptDrops(true);
667         pteMacroComment->setToolTipDuration(1);
668         pteMacroComment->setInputMethodHints(Qt::ImhNone);
669         pteMacroComment->setVerticalScrollBarPolicy(Qt::ScrollBarAlwaysOff);
670         pteMacroComment->setTabChangesFocus(true);
671         pteMacroComment->setUndoRedoEnabled(true);
672         pteMacroComment->setReadOnly(false);
673         pteMacroComment->setTabStopWidth(1);
674         pteMacroText = new QPlainTextEdit(macroTab);
675         pteMacroText->setObjectName(QStringLiteral("pteMacroText"));
676         pteMacroText->setGeometry(QRect(410, 50, 311, 66));
677         pteMacroText->setFocusPolicy(Qt::StrongFocus);
678         pteMacroText->setContextMenuPolicy(Qt::NoContextMenu);
679         pteMacroText->setAcceptDrops(false);
680         pteMacroText->setInputMethodHints(Qt::ImhNoAutoUppercase);
681         pteMacroText->setUndoRedoEnabled(false);
682         pteMacroText->setReadOnly(false);
683         pteMacroText->setTabStopWidth(1);
684         layoutWidget = new QWidget(macroTab);
685         layoutWidget->setObjectName(QStringLiteral("layoutWidget"));
686         layoutWidget->setGeometry(QRect(300, 10, 131, 74));
687         verticalLayout_2 = new QVBoxLayout(layoutWidget);
688         verticalLayout_2->setObjectName(QStringLiteral("verticalLayout_2"));
689         verticalLayout_2->setSizeConstraint(QLayout::SetNoConstraint);
690         verticalLayout_2->setContentsMargins(0, 0, 0, 0);
691         groupBox_3 = new QGroupBox(layoutWidget);
692         groupBox_3->setObjectName(QStringLiteral("groupBox_3"));
693
694         verticalLayout_2->addWidget(groupBox_3);
695
696         groupBox_4 = new QGroupBox(layoutWidget);
697         groupBox_4->setObjectName(QStringLiteral("groupBox_4"));
698
699         verticalLayout_2->addWidget(groupBox_4);
700
701         layoutWidget_2 = new QWidget(macroTab);
702         layoutWidget_2->setObjectName(QStringLiteral("layoutWidget_2"));
```

```
703         layoutWidget_2->setGeometry(QRect(300, 160, 407, 36));
704         gridLayout_10 = new QGridLayout(layoutWidget_2);
705         gridLayout_10->setObjectName(QStringLiteral("gridLayout_10"));
706         gridLayout_10->setContentsMargins(0, 0, 0, 0);
707         btnStopMacro = new QPushButton(layoutWidget_2);
708         btnStopMacro->setObjectName(QStringLiteral("btnStopMacro"));
709         btnStopMacro->setEnabled(false);
710 
711         gridLayout_10->addWidget(btnStopMacro, 0, 2, 1, 1);
712 
713         btnClearMacro = new QPushButton(layoutWidget_2);
714         btnClearMacro->setObjectName(QStringLiteral("btnClearMacro"));
715 
716         gridLayout_10->addWidget(btnClearMacro, 0, 3, 1, 1);
717 
718         btnStartMacro = new QPushButton(layoutWidget_2);
719         btnStartMacro->setObjectName(QStringLiteral("btnStartMacro"));
720 
721         gridLayout_10->addWidget(btnStartMacro, 0, 1, 1, 1);
722 
723         label = new QLabel(layoutWidget_2);
724         label->setObjectName(QStringLiteral("label"));
725         label->setStyleSheet(QLatin1String("font: 75 10pt ; \n"
726 ""));
727 
728         gridLayout_10->addWidget(label, 0, 0, 1, 1);
729 
730         lbl_macro = new QLabel(macroTab);
731         lbl_macro->setObjectName(QStringLiteral("lbl_macro"));
732         lbl_macro->setGeometry(QRect(300, 130, 431, 27));
733         lbl_macro->setStyleSheet(QStringLiteral("font: 75 11pt ; color: darkblue;"));
734         layoutWidget2 = new QWidget(macroTab);
735         layoutWidget2->setObjectName(QStringLiteral("layoutWidget2"));
736         layoutWidget2->setGeometry(QRect(40, 170, 238, 28));
737         delayButtons = new QHBoxLayout(layoutWidget2);
738         delayButtons->setSpacing(0);
739         delayButtons->setObjectName(QStringLiteral("delayButtons"));
740         delayButtons->setSizeConstraint(QLayout::SetMinimumSize);
741         delayButtons->setContentsMargins(0, 0, 0, 0);
742         rb_delay_no = new QRadioButton(layoutWidget2);
743         rb_delay_no->setObjectName(QStringLiteral("rb_delay_no"));
744         rb_delay_no->setStyleSheet(QStringLiteral("font: 10pt ; color:  green;"));
745 
746         delayButtons->addWidget(rb_delay_no);
747 
748         rb_delay_default = new QRadioButton(layoutWidget2);
749         rb_delay_default->setObjectName(QStringLiteral("rb_delay_default"));
750         rb_delay_default->setStyleSheet(QStringLiteral("font: 10pt ; color:  green;"));
751         rb_delay_default->setChecked(false);
752         rb_delay_default->setAutoExclusive(true);
753 
754         delayButtons->addWidget(rb_delay_default);
755 
756         rb_delay_asTyped = new QRadioButton(layoutWidget2);
757         rb_delay_asTyped->setObjectName(QStringLiteral("rb_delay_asTyped"));
758         rb_delay_asTyped->setStyleSheet(QStringLiteral("font: 10pt ; color:  green;"));
759         rb_delay_asTyped->setChecked(true);
760 
761         delayButtons->addWidget(rb_delay_asTyped);
762 
763         txtBuffer = new QLabel(macroTab);
764         txtBuffer->setObjectName(QStringLiteral("txtBuffer"));
765         txtBuffer->setEnabled(false);
766         txtBuffer->setGeometry(QRect(300, 100, 0, 0));
767         QSizePolicy sizePolicy4(QSizePolicy::Fixed, QSizePolicy::Fixed);
768         sizePolicy4.setHorizontalStretch(0);
769         sizePolicy4.setVerticalStretch(0);
770         sizePolicy4.setHeightForWidth(txtBuffer->sizePolicy().hasHeightForWidth());
771         txtBuffer->setSizePolicy(sizePolicy4);
772         txtBuffer->setMaximumSize(QSize(0, 0));
773         txtBuffer->setContextMenuPolicy(Qt::NoContextMenu);
774 #ifndef QT_NO_TOOLTIP
775         txtBuffer->setToolTip(QStringLiteral(""));
776 #endif // QT_NO_TOOLTIP
777         txtBuffer->setToolTipDuration(0);
778 #ifndef QT_NO_STATUSTIP
779         txtBuffer->setStatusTip(QStringLiteral(""));
780 #endif // QT_NO_STATUSTIP
781 #ifndef QT_NO_WHATSTHIS
782         txtBuffer->setWhatsThis(QStringLiteral(""));
783 #endif // QT_NO_WHATSTHIS
784         txtBuffer->setFrameShadow(QFrame::Plain);
785         txtBuffer->setLineWidth(0);
786         txtBuffer->setTextFormat(Qt::PlainText);
787         txtBuffer->setIndent(0);
788         txtBuffer->setTextInteractionFlags(Qt::NoTextInteraction);
789         tabWidget->addTab(macroTab, QString());
```

```
790         pteMacroComment->raise();
791         layoutWidget->raise();
792         layoutWidget1->raise();
793         layoutWidget->raise();
794         pteMacroText->raise();
795         layoutWidget_2->raise();
796         lbl_macro->raise();
797         txtBuffer->raise();
798 
799         verticalLayout->addWidget(tabWidget);
800 
801         horizontalLayout = new QHBoxLayout();
802         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
803         horizontalSpacer_14 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
804 
805         horizontalLayout->addItem(horizontalSpacer_14);
806 
807         unbindButton = new QPushButton(RebindWidget);
808         unbindButton->setObjectName(QStringLiteral("unbindButton"));
809 
810         horizontalLayout->addWidget(unbindButton);
811 
812         resetButton = new QPushButton(RebindWidget);
813         resetButton->setObjectName(QStringLiteral("resetButton"));
814 
815         horizontalLayout->addWidget(resetButton);
816 
817         cancelButton = new QPushButton(RebindWidget);
818         cancelButton->setObjectName(QStringLiteral("cancelButton"));
819 
820         horizontalLayout->addWidget(cancelButton);
821 
822         applyButton = new QPushButton(RebindWidget);
823         applyButton->setObjectName(QStringLiteral("applyButton"));
824 
825         horizontalLayout->addWidget(applyButton);
826 
827 
828         verticalLayout->addLayout(horizontalLayout);
829 
830         QWidget::setTabOrder(tabWidget, typingButton);
831         QWidget::setTabOrder(typingButton, typingBox);
832         QWidget::setTabOrder(typingBox, modButton);
833         QWidget::setTabOrder(modButton, modBox);
834         QWidget::setTabOrder(modBox, fnButton);
835         QWidget::setTabOrder(fnButton, fnBox);
836         QWidget::setTabOrder(fnBox, numButton);
837         QWidget::setTabOrder(numButton, numBox);
838         QWidget::setTabOrder(numBox, mediaButton);
839         QWidget::setTabOrder(mediaButton, mediaBox);
840         QWidget::setTabOrder(mediaBox, mbButton);
841         QWidget::setTabOrder(mbButton, mbBox);
842         QWidget::setTabOrder(mbBox, mb2Button);
843         QWidget::setTabOrder(mb2Button, mb2Box);
844         QWidget::setTabOrder(mb2Box, wheelButton);
845         QWidget::setTabOrder(wheelButton, wheelBox);
846         QWidget::setTabOrder(wheelBox, dpiButton);
847         QWidget::setTabOrder(dpiButton, dpiBox);
848         QWidget::setTabOrder(dpiBox, dpiCustXBox);
849         QWidget::setTabOrder(dpiCustXBox, dpiCustYBox);
850         QWidget::setTabOrder(dpiCustYBox, animButton);
851         QWidget::setTabOrder(animButton, animBox);
852         QWidget::setTabOrder(animBox, animOnceBox);
853         QWidget::setTabOrder(animOnceBox, animKrBox);
854         QWidget::setTabOrder(animKrBox, modeButton);
855         QWidget::setTabOrder(modeButton, modeBox);
856         QWidget::setTabOrder(modeBox, modeWrapBox);
857         QWidget::setTabOrder(modeWrapBox, lightButton);
858         QWidget::setTabOrder(lightButton, lightBox);
859         QWidget::setTabOrder(lightBox, lightWrapBox);
860         QWidget::setTabOrder(lightWrapBox, lockButton);
861         QWidget::setTabOrder(lockButton, lockBox);
862         QWidget::setTabOrder(lockBox, programKpButton);
863         QWidget::setTabOrder(programKpButton, programKpBox);
864         QWidget::setTabOrder(programKpBox, programKpSIBox);
865         QWidget::setTabOrder(programKpSIBox, programKpModeBox);
866         QWidget::setTabOrder(programKpModeBox, programKrButton);
867         QWidget::setTabOrder(programKrButton, programKrBox);
868         QWidget::setTabOrder(programKrBox, programKrSIBox);
869         QWidget::setTabOrder(programKrSIBox, programKrModeBox);
870         QWidget::setTabOrder(programKrModeBox, pteMacroBox);
871         QWidget::setTabOrder(pteMacroBox, rb_delay_no);
872         QWidget::setTabOrder(rb_delay_no, rb_delay_default);
873         QWidget::setTabOrder(rb_delay_default, rb_delay_asTyped);
874         QWidget::setTabOrder(rb_delay_asTyped, pteMacroComment);
875         QWidget::setTabOrder(pteMacroComment, pteMacroText);
```

```
876         QWidget::setTabOrder(pteMacroText, btnStartMacro);
877         QWidget::setTabOrder(btnStartMacro, btnStopMacro);
878         QWidget::setTabOrder(btnStopMacro, btnClearMacro);
879         QWidget::setTabOrder(btnClearMacro, unbindButton);
880         QWidget::setTabOrder(unbindButton, resetButton);
881         QWidget::setTabOrder(resetButton, cancelButton);
882         QWidget::setTabOrder(cancelButton, applyButton);
883
884         retranslateUi(RebindWidget);
885         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    pteMacroText, SLOT(clear()));
886         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    pteMacroComment, SLOT(clear()));
887         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    btnStopMacro, SLOT(animateClick()));
888         QObject::connect(cancelButton, SIGNAL(clicked()), txtBuffer, SLOT(clear()));
889         QObject::connect(btnClearMacro, SIGNAL(clicked()),
    txtBuffer, SLOT(clear()));
890         QObject::connect(btnStartMacro, SIGNAL(clicked()),
    txtBuffer, SLOT(clear()));
891         QObject::connect(unbindButton, SIGNAL(clicked()), txtBuffer, SLOT(clear()));
892         QObject::connect(resetButton, SIGNAL(clicked()), txtBuffer, SLOT(clear()));
893
894         tabWidget->setCurrentIndex(5);
895         applyButton->setDefault(true);
896
897
898         QMetaObject::connectSlotsByName(RebindWidget);
899     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:

12.101.3 Field Documentation

12.101.3.1 QWidget* Ui_RebindWidget::actionTab

Definition at line 92 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.2 QComboBox* Ui_RebindWidget::animBox

Definition at line 84 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_animBox_currentIndexChanged(), RebindWidget::on_animButton_clicked(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and

setupUi().

12.101.3.3 QRadioButton* Ui_RebindWidget::animButton

Definition at line 90 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_animBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.4 QCheckBox* Ui_RebindWidget::animKrBox

Definition at line 87 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.5 QCheckBox* Ui_RebindWidget::animOnceBox

Definition at line 88 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.6 QWidget* Ui_RebindWidget::animTab

Definition at line 81 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.7 QPushButton* Ui_RebindWidget::applyButton

Definition at line 161 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), and setupUi().

12.101.3.8 QPushButton* Ui_RebindWidget::btnClearMacro

Definition at line 146 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.9 QPushButton* Ui_RebindWidget::btnStartMacro

Definition at line 147 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), and setupUi().

12.101.3.10 QPushButton* Ui_RebindWidget::btnStopMacro

Definition at line 145 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), and setupUi().

12.101.3.11 QPushButton* Ui_RebindWidget::cancelButton

Definition at line 160 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.12 QHBoxLayout* Ui_RebindWidget::delayButtons

Definition at line 151 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.13 QComboBox* Ui_RebindWidget::dpiBox

Definition at line 68 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_dpiBox_currentIndexChanged(), RebindWidget::on_dpiButton_clicked(), retranslateUi(), RebindWidget::setBind(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.14 QRadioButton* Ui_RebindWidget::dpiButton

Definition at line 62 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_dpiBox_currentIndexChanged(), retranslateUi(), RebindWidget::setBind(), and setupUi().

12.101.3.15 QLabel* Ui_RebindWidget::dpiCustLabel

Definition at line 78 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_dpiBox_currentIndexChanged(), retranslateUi(), RebindWidget::setBind(), and setupUi().

12.101.3.16 QSpinBox* Ui_RebindWidget::dpiCustXBox

Definition at line 73 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_dpiBox_currentIndexChanged(), RebindWidget::setBind(), RebindWidget::setSelection(), and setupUi().

12.101.3.17 QSpinBox* Ui_RebindWidget::dpiCustYBox

Definition at line 69 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_dpiBox_currentIndexChanged(), RebindWidget::setBind(), RebindWidget::setSelection(), and setupUi().

12.101.3.18 QLabel* Ui_RebindWidget::dpiWarning

Definition at line 74 of file ui_rebindwidget.h.

Referenced by retranslateUi(), RebindWidget::setBind(), and setupUi().

12.101.3.19 QComboBox* Ui_RebindWidget::fnBox

Definition at line 44 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_fnBox_currentIndexChanged(), RebindWidget::on_fnButton_clicked(), RebindWidget::RebindWidget(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.20 QRadioButton* Ui_RebindWidget::fnButton

Definition at line 56 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_fnBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.21 QGridLayout* Ui_RebindWidget::gridLayout

Definition at line 61 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.22 QGridLayout* Ui_RebindWidget::gridLayout_10

Definition at line 144 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.23 QGridLayout* Ui_RebindWidget::gridLayout_2

Definition at line 41 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.24 QGridLayout* Ui_RebindWidget::gridLayout_3

Definition at line 93 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.25 QGridLayout* Ui_RebindWidget::gridLayout_4

Definition at line 111 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.26 QGridLayout* Ui_RebindWidget::gridLayout_5

Definition at line 82 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.27 QGridLayout* Ui_RebindWidget::gridLayout_8

Definition at line 134 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.28 QGroupBox* Ui_RebindWidget::groupBox_2

Definition at line 135 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.29 QGroupBox* Ui_RebindWidget::groupBox_3

Definition at line 141 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.30 QGroupBox* Ui_RebindWidget::groupBox_4

Definition at line 142 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.31 QHBoxLayout* Ui_RebindWidget::horizontalLayout

Definition at line 156 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.32 QHBoxLayout* Ui_RebindWidget::horizontalLayout_2

Definition at line 122 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.33 QHBoxLayout* Ui_RebindWidget::horizontalLayout_3

Definition at line 127 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.34 QSpacerItem* Ui_RebindWidget::horizontalSpacer

Definition at line 113 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.35 QSpacerItem* Ui_RebindWidget::horizontalSpacer_10

Definition at line 107 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.36 QSpacerItem* Ui_RebindWidget::horizontalSpacer_11

Definition at line 116 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.37 QSpacerItem* Ui_RebindWidget::horizontalSpacer_12

Definition at line 120 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.38 QSpacerItem* Ui_RebindWidget::horizontalSpacer_13

Definition at line 117 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.39 QSpacerItem* Ui_RebindWidget::horizontalSpacer_14

Definition at line 157 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.40 QSpacerItem* Ui_RebindWidget::horizontalSpacer_15

Definition at line 125 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.41 QSpacerItem* Ui_RebindWidget::horizontalSpacer_16

Definition at line 130 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.42 QSpacerItem* Ui_RebindWidget::horizontalSpacer_17

Definition at line 59 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.43 QSpacerItem* Ui_RebindWidget::horizontalSpacer_18

Definition at line 65 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.44 QSpacerItem* Ui_RebindWidget::horizontalSpacer_19

Definition at line 79 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.45 QSpacerItem* Ui_RebindWidget::horizontalSpacer_2

Definition at line 47 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.46 QSpacerItem* Ui_RebindWidget::horizontalSpacer_20

Definition at line 66 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.47 QSpacerItem* Ui_RebindWidget::horizontalSpacer_21

Definition at line 80 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.48 QSpacerItem* Ui_RebindWidget::horizontalSpacer_22

Definition at line 76 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.49 QSpacerItem* Ui_RebindWidget::horizontalSpacer_23

Definition at line 91 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.50 QSpacerItem* Ui_RebindWidget::horizontalSpacer_24

Definition at line 85 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.51 QSpacerItem* Ui_RebindWidget::horizontalSpacer_25

Definition at line 86 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.52 QSpacerItem* Ui_RebindWidget::horizontalSpacer_26

Definition at line 89 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.53 QSpacerItem* Ui_RebindWidget::horizontalSpacer_3

Definition at line 50 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.54 QSpacerItem* Ui_RebindWidget::horizontalSpacer_4

Definition at line 54 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.55 QSpacerItem* Ui_RebindWidget::horizontalSpacer_5

Definition at line 94 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.56 QSpacerItem* Ui_RebindWidget::horizontalSpacer_6

Definition at line 109 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.57 QSpacerItem* Ui_RebindWidget::horizontalSpacer_7

Definition at line 108 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.58 QSpacerItem* Ui_RebindWidget::horizontalSpacer_8

Definition at line 106 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.59 QSpacerItem* Ui_RebindWidget::horizontalSpacer_9

Definition at line 53 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.60 QWidget* Ui_RebindWidget::keyTab

Definition at line 40 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.61 QLabel* Ui_RebindWidget::label

Definition at line 148 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.62 QWidget* Ui_RebindWidget::layoutWidget

Definition at line 139 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.63 QWidget* Ui_RebindWidget::layoutWidget1

Definition at line 133 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.64 QWidget* Ui_RebindWidget::layoutWidget2

Definition at line 150 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.65 QWidget* Ui_RebindWidget::layoutWidget_2

Definition at line 143 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.66 QLabel* Ui_RebindWidget::lbl_macro

Definition at line 149 of file ui_rebindwidget.h.

Referenced by RebindWidget::helpStatus(), retranslateUi(), and setupUi().

12.101.3.67 QComboBox* Ui_RebindWidget::lightBox

Definition at line 102 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_lightBox_currentIndexChanged(), RebindWidget::on_lightButton_clicked(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.68 QRadioButton* Ui_RebindWidget::lightButton

Definition at line 104 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_lightBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.69 QCheckBox* Ui_RebindWidget::lightWrapBox

Definition at line 97 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_lightBox_currentIndexChanged(), RebindWidget::RebindWidget(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.70 QComboBox* Ui_RebindWidget::lockBox

Definition at line 103 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_lockBox_currentIndexChanged(), RebindWidget::on_lockButton_clicked(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.71 QRadioButton* Ui_RebindWidget::lockButton

Definition at line 100 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_lockBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.72 QWidget* Ui_RebindWidget::macroTab

Definition at line 132 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.73 QComboBox* Ui_RebindWidget::mb2Box

Definition at line 67 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_mb2Box_currentIndexChanged(), RebindWidget::on_mb2Button_clicked(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.74 QRadioButton* Ui_RebindWidget::mb2Button

Definition at line 77 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_mb2Box_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.75 QComboBox* Ui_RebindWidget::mbBox

Definition at line 70 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_mbBox_currentIndexChanged(), RebindWidget::on_mbButton_clicked(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.76 QRadioButton* Ui_RebindWidget::mbButton

Definition at line 72 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_mbBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.77 QComboBox* Ui_RebindWidget::mediaBox

Definition at line 57 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_mediaBox_currentIndexChanged(), RebindWidget::on_mediaButton_clicked(), RebindWidget::RebindWidget(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.78 QRadioButton* Ui_RebindWidget::mediaButton

Definition at line 51 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_mediaBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.79 QComboBox* Ui_RebindWidget::modBox

Definition at line 49 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_modBox_currentIndexChanged(), RebindWidget::on_modButton_clicked(), RebindWidget::RebindWidget(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.80 QRadioButton* Ui_RebindWidget::modButton

Definition at line 55 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_modBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.81 QComboBox* Ui_RebindWidget::modeBox

Definition at line 105 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_modeBox_currentIndexChanged(), RebindWidget::on_modeButton_clicked(), retranslateUi(), RebindWidget::setBind(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.82 QRadioButton* Ui_RebindWidget::modeButton

Definition at line 95 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_modeBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.83 QCheckBox* Ui_RebindWidget::modeWrapBox

Definition at line 96 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_modeBox_currentIndexChanged(), RebindWidget::RebindWidget(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.84 QWidget* Ui_RebindWidget::mouseTab

Definition at line 60 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.85 QComboBox* Ui_RebindWidget::numBox

Definition at line 42 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_numBox_currentIndexChanged(), RebindWidget::on_numButton_clicked(), RebindWidget::RebindWidget(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.86 QRadioButton* Ui_RebindWidget::numButton

Definition at line 58 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_numBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.87 QLineEdit* Ui_RebindWidget::programKpBox

Definition at line 115 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKpBox_textChanged(), RebindWidget::on_programKpButton_clicked(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.88 QCheckBox* Ui_RebindWidget::programKpButton

Definition at line 114 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKpBox_textChanged(), retranslateUi(), RebindWidget::setBox(), and setupUi().

12.101.3.89 QWidget* Ui_RebindWidget::programKpExtra

Definition at line 121 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKpBox_textChanged(), RebindWidget::RebindWidget(), and setupUi().

12.101.3.90 QComboBox* Ui_RebindWidget::programKpModeBox

Definition at line 124 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKpSIBox_clicked(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.91 QCheckBox* Ui_RebindWidget::programKpSIBox

Definition at line 123 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.92 QLineEdit* Ui_RebindWidget::programKrBox

Definition at line 118 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKrBox_textChanged(), RebindWidget::on_programKrButton_clicked(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.93 QCheckBox* Ui_RebindWidget::programKrButton

Definition at line 119 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKrBox_textChanged(), retranslateUi(), RebindWidget::setBox(), and setupUi().

12.101.3.94 QWidget* Ui_RebindWidget::programKrExtra

Definition at line 126 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_programKrBox_textChanged(), RebindWidget::RebindWidget(), and setupUi().

12.101.3.95 QComboBox* Ui_RebindWidget::programKrModeBox

Definition at line 129 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_programKrSIBox_clicked(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.96 QCheckBox* Ui_RebindWidget::programKrSIBox

Definition at line 128 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.97 QWidget* Ui_RebindWidget::programTab

Definition at line 110 of file ui_rebindwidget.h.

Referenced by retranslateUi(), and setupUi().

12.101.3.98 QLabel* Ui_RebindWidget::progTipLabel

Definition at line 131 of file ui_rebindwidget.h.

Referenced by RebindWidget::RebindWidget(), retranslateUi(), and setupUi().

12.101.3.99 QPlainTextEdit* Ui_RebindWidget::pteMacroBox

Definition at line 136 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::convertMacroBox(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_pteMacroBox_textChanged(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.100 QPlainTextEdit* Ui_RebindWidget::pteMacroComment

Definition at line 137 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.101 QPlainTextEdit* Ui_RebindWidget::pteMacroText

Definition at line 138 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_btnStartMacro_clicked(), retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.102 QRadioButton* Ui_RebindWidget::rb_delay_asTyped

Definition at line 154 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), RebindWidget::setCorrectRadioButton(), and setupUi().

12.101.3.103 QRadioButton* Ui_RebindWidget::rb_delay_default

Definition at line 153 of file ui_rebindwidget.h.

Referenced by RebindWidget::convertMacroBox(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), RebindWidget::setCorrectRadioButton(), and setupUi().

12.101.3.104 QRadioButton* Ui_RebindWidget::rb_delay_no

Definition at line 152 of file ui_rebindwidget.h.

Referenced by RebindWidget::convertMacroBox(), RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), RebindWidget::setCorrectRadioButton(), and setupUi().

12.101.3.105 QPushButton* Ui_RebindWidget::resetButton

Definition at line 159 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), and setupUi().

12.101.3.106 QTabWidget* Ui_RebindWidget::tabWidget

Definition at line 39 of file ui_rebindwidget.h.

Referenced by retranslateUi(), RebindWidget::setSelection(), and setupUi().

12.101.3.107 QLabel* Ui_RebindWidget::txtBuffer

Definition at line 155 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::convertMacroBox(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.108 QComboBox* Ui_RebindWidget::typingBox

Definition at line 43 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_typingBox_currentIndexChanged(), RebindWidget::on_typingButton_clicked(), retranslateUi(), RebindWidget::setBind(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.109 QRadioButton* Ui_RebindWidget::typingButton

Definition at line 48 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_typingBox_currentIndexChanged(), retranslateUi(), and setupUi().

12.101.3.110 QPushButton* Ui_RebindWidget::unbindButton

Definition at line 158 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_btnStartMacro_clicked(), RebindWidget::on_btnStopMacro_clicked(), retranslateUi(), and setupUi().

12.101.3.111 QVBoxLayout* Ui_RebindWidget::verticalLayout

Definition at line 38 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.112 QVBoxLayout* Ui_RebindWidget::verticalLayout_2

Definition at line 140 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.113 QSpacerItem* Ui_RebindWidget::verticalSpacer

Definition at line 112 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.114 QSpacerItem* Ui_RebindWidget::verticalSpacer_10

Definition at line 45 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.115 QSpacerItem* Ui_RebindWidget::verticalSpacer_11

Definition at line 83 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.116 QSpacerItem* Ui_RebindWidget::verticalSpacer_2

Definition at line 63 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.117 QSpacerItem* Ui_RebindWidget::verticalSpacer_3

Definition at line 46 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.118 QSpacerItem* Ui_RebindWidget::verticalSpacer_4

Definition at line 52 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.119 QSpacerItem* Ui_RebindWidget::verticalSpacer_5

Definition at line 101 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.120 QSpacerItem* Ui_RebindWidget::verticalSpacer_6

Definition at line 99 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.121 QSpacerItem* Ui_RebindWidget::verticalSpacer_7

Definition at line 98 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.122 QSpacerItem* Ui_RebindWidget::verticalSpacer_8

Definition at line 64 of file ui_rebindwidget.h.

Referenced by setupUi().

12.101.3.123 QComboBox* Ui_RebindWidget::wheelBox

Definition at line 71 of file ui_rebindwidget.h.

Referenced by RebindWidget::applyChanges(), RebindWidget::on_wheelBox_currentIndexChanged(), RebindWidget::on_wheelButton_clicked(), retranslateUi(), RebindWidget::setBox(), RebindWidget::setSelection(), and setupUi().

12.101.3.124 QRadioButton* Ui_RebindWidget::wheelButton

Definition at line 75 of file ui_rebindwidget.h.

Referenced by RebindWidget::on_wheelBox_currentIndexChanged(), retranslateUi(), and setupUi().

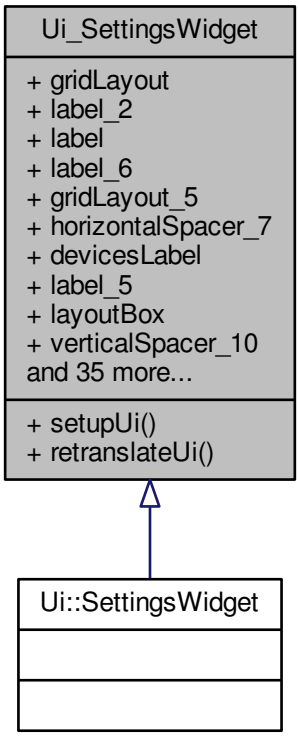
The documentation for this class was generated from the following file:

- src/ckb/ui_rebindwidget.h

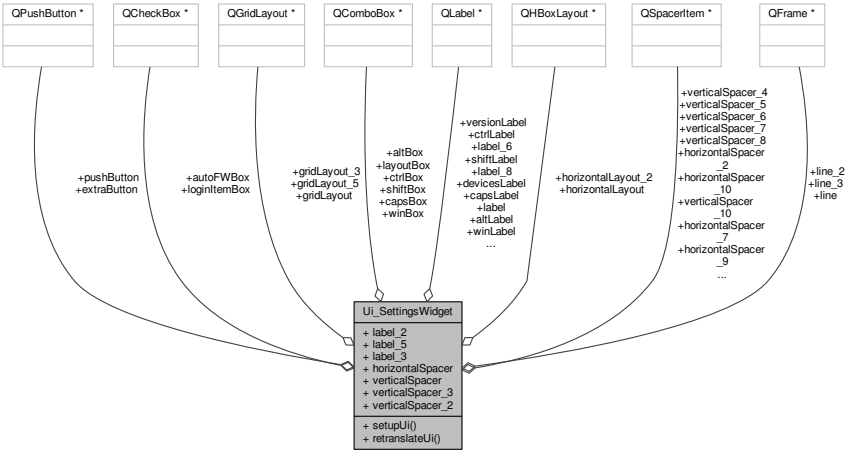
12.102 Ui_SettingsWidget Class Reference

`#include <src/ckb/ui_settingswidget.h>`

Inheritance diagram for Ui_SettingsWidget:



Collaboration diagram for Ui_SettingsWidget:



Public Member Functions

- void setupUi (QWidget *SettingsWidget)
- void retranslateUi (QWidget *SettingsWidget)

Data Fields

- QGridLayout * gridLayout
- QLabel * label_2
- QLabel * label
- QLabel * label_6
- QGridLayout * gridLayout_5
- QSpacerItem * horizontalSpacer_7
- QLabel * devicesLabel
- QLabel * label_5
- QComboBox * layoutBox
- QSpacerItem * verticalSpacer_10
- QHBoxLayout * horizontalLayout_2
- QLabel * label_3
- QSpacerItem * horizontalSpacer
- QPushButton * pushButton
- QSpacerItem * verticalSpacer
- QSpacerItem * verticalSpacer_8
- QCheckBox * loginItemBox
- QSpacerItem * verticalSpacer_6
- QCheckBox * autoFWBox
- QFrame * line_2
- QLabel * label_8
- QGridLayout * gridLayout_3
- QLabel * shiftLabel
- QLabel * capsLabel
- QSpacerItem * horizontalSpacer_10
- QComboBox * capsBox
- QLabel * ctrlLabel

- QSpacerItem * horizontalSpacer_9
- QComboBox * ctrlBox
- QComboBox * shiftBox
- QLabel * winLabel
- QComboBox * winBox
- QComboBox * altBox
- QLabel * altLabel
- QFrame * line
- QSpacerItem * verticalSpacer_3
- QSpacerItem * verticalSpacer_5
- QFrame * line_3
- QSpacerItem * verticalSpacer_4
- QSpacerItem * verticalSpacer_2
- QLabel * versionLabel
- QHBoxLayout * horizontalLayout
- QPushButton * extraButton
- QSpacerItem * horizontalSpacer_2
- QSpacerItem * verticalSpacer_7

12.102.1 Detailed Description

Definition at line 29 of file ui_settingswidget.h.

12.102.2 Member Function Documentation

12.102.2.1 void Ui_SettingsWidget::retranslateUi (QWidget * *SettingsWidget*) [inline]

Definition at line 356 of file ui_settingswidget.h.

References altLabel, autoFWBox, capsLabel, ctrlLabel, devicesLabel, extraButton, label, label_2, label_3, label_5, label_6, label_8, loginItemBox, pushButton, shiftLabel, versionLabel, and winLabel.

Referenced by setupUi().

```
357     {
358         SettingsWidget->setWindowTitle(QApplication::translate("SettingsWidget", "Form", Q_NULLPTR));
359         label_2->setText(QApplication::translate("SettingsWidget", "\302\251 2014-2016 <a href=\"
      https://github.com/ccMSC/\" style=\"text-decoration:none;\">ccMSC</a>.<br/>\302\251 2017 <a href=\"
      https://github.com/mattanger/ckb-next/graphs/contributors\" style=\"text-decoration:none;\">The ckb-next development
       team</a>.", Q_NULLPTR));
360         label->setText(QApplication::translate("SettingsWidget", "Modifier keys", Q_NULLPTR));
361         label_6->setText(QApplication::translate("SettingsWidget", "Application", Q_NULLPTR));
362         devicesLabel->setText(QApplication::translate("SettingsWidget", "No devices connected",
      Q_NULLPTR));
363         label_5->setText(QApplication::translate("SettingsWidget", "Keyboard layout:", Q_NULLPTR));
364         label_3->setText(QApplication::translate("SettingsWidget", "<a href=\"
      https://github.com/mattanger/ckb-next\" style=\"text-decoration:none;\">https://github.com/mattanger/ckb-next</a>", Q_NU
365         pushButton->setText(QApplication::translate("SettingsWidget", "Quit", Q_NULLPTR));
366 #ifndef QT_NO_TOOLTIP
367         loginItemBox->setToolTip(QApplication::translate("SettingsWidget", "ckb-next will be
      started when you log in to your computer.", Q_NULLPTR));
368 #endif // QT_NO_TOOLTIP
369         loginItemBox->setText(QApplication::translate("SettingsWidget", "Start ckb-next at
      login", Q_NULLPTR));
370 #ifndef QT_NO_TOOLTIP
371         autoFWBox->setToolTip(QApplication::translate("SettingsWidget", "You will be notified when
      new firmware versions are available. You'll have the option to install them immediately or wait until
      later.", Q_NULLPTR));
372 #endif // QT_NO_TOOLTIP
373         autoFWBox->setText(QApplication::translate("SettingsWidget", "Check for new firmware
      automatically", Q_NULLPTR));
374         label_8->setText(QApplication::translate("SettingsWidget", "These will override the keyboard
      profile. See \"Binding\" tab for more settings.", Q_NULLPTR));
375         shiftLabel->setText(QApplication::translate("SettingsWidget", "Shift:", Q_NULLPTR));
376         capsLabel->setText(QApplication::translate("SettingsWidget", "Caps Lock:", Q_NULLPTR));
377         ctrlLabel->setText(QApplication::translate("SettingsWidget", "Ctrl:", Q_NULLPTR));
378         winLabel->setText(QApplication::translate("SettingsWidget", "Windows:", Q_NULLPTR));
```

```
379         altLabel->setText(QApplication::translate("SettingsWidget", "Alt:", Q_NULLPTR));
380         versionLabel->setText(QApplication::translate("SettingsWidget", "ckb-next", Q_NULLPTR))
    ;
381         extraButton->setText(QApplication::translate("SettingsWidget", "More settings",
    Q_NULLPTR));
382     } // retranslateUi
```

Here is the caller graph for this function:

12.102.2.2 void Ui_SettingsWidget::setupUi (QWidget * *SettingsWidget*) [inline]

Definition at line 78 of file ui_settingswidget.h.

References altBox, altLabel, autoFWBox, capsBox, capsLabel, ctrlBox, ctrlLabel, devicesLabel, extraButton, gridLayout, gridLayout_3, gridLayout_5, horizontalLayout, horizontalLayout_2, horizontalSpacer, horizontalSpacer_10, horizontalSpacer_2, horizontalSpacer_7, horizontalSpacer_9, label, label_2, label_3, label_5, label_6, label_8, layoutBox, line, line_2, line_3, loginItemBox, pushButton, retranslateUi(), shiftBox, shiftLabel, versionLabel, verticalSpacer, verticalSpacer_10, verticalSpacer_2, verticalSpacer_3, verticalSpacer_4, verticalSpacer_5, verticalSpacer_6, verticalSpacer_7, verticalSpacer_8, winBox, and winLabel.

Referenced by SettingsWidget::SettingsWidget().

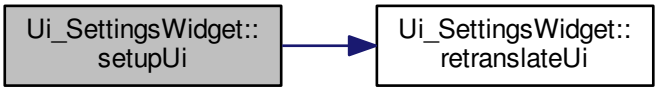
```
79     {
80         if (SettingsWidget->objectName().isEmpty())
81             SettingsWidget->setObjectName(QStringLiteral("SettingsWidget"));
82         SettingsWidget->resize(827, 654);
83         gridLayout = new QGridLayout(SettingsWidget);
84         gridLayout->setObjectName(QStringLiteral("gridLayout"));
85         label_2 = new QLabel(SettingsWidget);
86         label_2->setObjectName(QStringLiteral("label_2"));
87         label_2->setAlignment(Qt::AlignLeading|Qt::AlignLeft|Qt::AlignVCenter);
88         label_2->setOpenExternalLinks(true);
89 
90         gridLayout->addWidget(label_2, 20, 0, 1, 2);
91 
92         label = new QLabel(SettingsWidget);
93         label->setObjectName(QStringLiteral("label"));
94         QFont font;
95         font.setBold(true);
96         font.setWeight(75);
97         label->setFont(font);
98 
99         gridLayout->addWidget(label, 6, 0, 1, 2);
100 
101          label_6 = new QLabel(SettingsWidget);
102          label_6->setObjectName(QStringLiteral("label_6"));
103          label_6->setFont(font);
104 
105          gridLayout->addWidget(label_6, 12, 0, 1, 2);
106 
107          gridLayout_5 = new QGridLayout();
108          gridLayout_5->setObjectName(QStringLiteral("gridLayout_5"));
109          horizontalSpacer_7 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
110 
111          gridLayout_5->addItem(horizontalSpacer_7, 2, 2, 1, 1);
112 
113          devicesLabel = new QLabel(SettingsWidget);
114          devicesLabel->setObjectName(QStringLiteral("devicesLabel"));
115 
116          gridLayout_5->addWidget(devicesLabel, 0, 0, 1, 3);
117 
118          label_5 = new QLabel(SettingsWidget);
119          label_5->setObjectName(QStringLiteral("label_5"));
120 
```

```
121         gridLayout_5->addWidget(label_5, 2, 0, 1, 1);
122
123         layoutBox = new QComboBox(SettingsWidget);
124         layoutBox->setObjectName(QStringLiteral("layoutBox"));
125
126         gridLayout_5->addWidget(layoutBox, 2, 1, 1, 1);
127
128         verticalSpacer_10 = new QSpacerItem(20, 6, QSizePolicy::Minimum,
    QSizePolicy::Fixed);
129
130         gridLayout_5->addItem(verticalSpacer_10, 1, 0, 1, 1);
131
132
133         gridLayout->addLayout(gridLayout_5, 4, 1, 1, 1);
134
135         horizontalLayout_2 = new QHBoxLayout();
136         horizontalLayout_2->setObjectName(QStringLiteral("horizontalLayout_2"));
137         label_3 = new QLabel(SettingsWidget);
138         label_3->setObjectName(QStringLiteral("label_3"));
139         label_3->setOpenExternalLinks(true);
140
141         horizontalLayout_2->addWidget(label_3);
142
143         horizontalSpacer = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
144
145         horizontalLayout_2->addItem(horizontalSpacer);
146
147         pushButton = new QPushButton(SettingsWidget);
148         pushButton->setObjectName(QStringLiteral("pushButton"));
149         pushButton->setFlat(false);
150
151         horizontalLayout_2->addWidget(pushButton);
152
153
154         gridLayout->addLayout(horizontalLayout_2, 21, 0, 1, 2);
155
156         verticalSpacer = new QSpacerItem(20, 40, QSizePolicy::Minimum, QSizePolicy::Expanding
    );
157
158         gridLayout->addItem(verticalSpacer, 19, 0, 1, 2);
159
160         verticalSpacer_8 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
161
162         gridLayout->addItem(verticalSpacer_8, 16, 0, 1, 1);
163
164         loginItemBox = new QCheckBox(SettingsWidget);
165         loginItemBox->setObjectName(QStringLiteral("loginItemBox"));
166
167         gridLayout->addWidget(loginItemBox, 16, 1, 1, 1);
168
169         verticalSpacer_6 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
170
171         gridLayout->addItem(verticalSpacer_6, 15, 0, 1, 1);
172
173         autoFWBox = new QCheckBox(SettingsWidget);
174         autoFWBox->setObjectName(QStringLiteral("autoFWBox"));
175         autoFWBox->setChecked(true);
176
177         gridLayout->addWidget(autoFWBox, 15, 1, 1, 1);
178
179         line_2 = new QFrame(SettingsWidget);
180         line_2->setObjectName(QStringLiteral("line_2"));
181         line_2->setFrameShape(QFrame::HLine);
182         line_2->setFrameShadow(QFrame::Sunken);
183
184         gridLayout->addWidget(line_2, 13, 0, 1, 2);
185
186         label_8 = new QLabel(SettingsWidget);
187         label_8->setObjectName(QStringLiteral("label_8"));
188         QFont font1;
189         font1.setItalic(true);
190         label_8->setFont(font1);
191
192         gridLayout->addWidget(label_8, 8, 1, 1, 1);
193
194         gridLayout_3 = new QGridLayout();
195         gridLayout_3->setObjectName(QStringLiteral("gridLayout_3"));
196         shiftLabel = new QLabel(SettingsWidget);
197         shiftLabel->setObjectName(QStringLiteral("shiftLabel"));
198         QSizePolicy sizePolicy(QSizePolicy::Expanding, QSizePolicy::Preferred);
199         sizePolicy.setHorizontalStretch(0);
200         sizePolicy.setVerticalStretch(0);
201         sizePolicy.setHeightForWidth(shiftLabel->sizePolicy().hasHeightForWidth());
202         shiftLabel->setSizePolicy(sizePolicy);
```

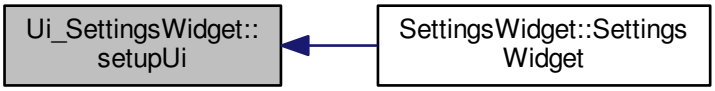
```
203         shiftLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
204
205         gridLayout_3->addWidget(shiftLabel, 0, 3, 1, 1);
206
207         capsLabel = new QLabel(SettingsWidget);
208         capsLabel->setObjectName(QStringLiteral("capsLabel"));
209         capsLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
210
211         gridLayout_3->addWidget(capsLabel, 0, 1, 1, 1);
212
213         horizontalSpacer_10 = new QSpacerItem(0, 0, QSizePolicy::Fixed,
    QSizePolicy::Minimum);
214
215         gridLayout_3->addItem(horizontalSpacer_10, 0, 0, 1, 1);
216
217         capsBox = new QComboBox(SettingsWidget);
218         capsBox->setObjectName(QStringLiteral("capsBox"));
219         QSizePolicy sizePolicy1(QSizePolicy::Expanding, QSizePolicy::Fixed);
220         sizePolicy1.setHorizontalStretch(0);
221         sizePolicy1.setVerticalStretch(0);
222         sizePolicy1.setHeightForWidth(capsBox->sizePolicy().hasHeightForWidth());
223         capsBox->setSizePolicy(sizePolicy1);
224
225         gridLayout_3->addWidget(capsBox, 0, 2, 1, 1);
226
227         ctrlLabel = new QLabel(SettingsWidget);
228         ctrlLabel->setObjectName(QStringLiteral("ctrlLabel"));
229         ctrlLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
230
231         gridLayout_3->addWidget(ctrlLabel, 1, 1, 1, 1);
232
233         horizontalSpacer_9 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
234
235         gridLayout_3->addItem(horizontalSpacer_9, 1, 7, 1, 1);
236
237         ctrlBox = new QComboBox(SettingsWidget);
238         ctrlBox->setObjectName(QStringLiteral("ctrlBox"));
239
240         gridLayout_3->addWidget(ctrlBox, 1, 2, 1, 1);
241
242         shiftBox = new QComboBox(SettingsWidget);
243         shiftBox->setObjectName(QStringLiteral("shiftBox"));
244         sizePolicy1.setHeightForWidth(shiftBox->sizePolicy().hasHeightForWidth());
245         shiftBox->setSizePolicy(sizePolicy1);
246
247         gridLayout_3->addWidget(shiftBox, 0, 4, 1, 1);
248
249         winLabel = new QLabel(SettingsWidget);
250         winLabel->setObjectName(QStringLiteral("winLabel"));
251         sizePolicy.setHeightForWidth(winLabel->sizePolicy().hasHeightForWidth());
252         winLabel->setSizePolicy(sizePolicy);
253         winLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
254
255         gridLayout_3->addWidget(winLabel, 1, 3, 1, 1);
256
257         winBox = new QComboBox(SettingsWidget);
258         winBox->setObjectName(QStringLiteral("winBox"));
259         sizePolicy1.setHeightForWidth(winBox->sizePolicy().hasHeightForWidth());
260         winBox->setSizePolicy(sizePolicy1);
261
262         gridLayout_3->addWidget(winBox, 1, 4, 1, 1);
263
264         altBox = new QComboBox(SettingsWidget);
265         altBox->setObjectName(QStringLiteral("altBox"));
266         sizePolicy1.setHeightForWidth(altBox->sizePolicy().hasHeightForWidth());
267         altBox->setSizePolicy(sizePolicy1);
268
269         gridLayout_3->addWidget(altBox, 1, 6, 1, 1);
270
271         altLabel = new QLabel(SettingsWidget);
272         altLabel->setObjectName(QStringLiteral("altLabel"));
273         sizePolicy.setHeightForWidth(altLabel->sizePolicy().hasHeightForWidth());
274         altLabel->setSizePolicy(sizePolicy);
275         altLabel->setAlignment(Qt::AlignRight|Qt::AlignTrailing|Qt::AlignVCenter);
276
277         gridLayout_3->addWidget(altLabel, 1, 5, 1, 1);
278
279
280         gridLayout->addLayout(gridLayout_3, 10, 1, 1, 1);
281
282         line = new QFrame(SettingsWidget);
283         line->setObjectName(QStringLiteral("line"));
284         line->setFrameShape(QFrame::HLine);
285         line->setFrameShadow(QFrame::Sunken);
286
287         gridLayout->addWidget(line, 1, 0, 1, 2);
```

```
288
289         verticalSpacer_3 = new QSpacerItem(0, 5, QSizePolicy::Minimum, QSizePolicy::Fixed);
290
291         gridLayout->addItem(verticalSpacer_3, 2, 1, 1, 1);
292
293         verticalSpacer_5 = new QSpacerItem(20, 12, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
294
295         gridLayout->addItem(verticalSpacer_5, 11, 1, 1, 1);
296
297         line_3 = new QFrame(SettingsWidget);
298         line_3->setObjectName(QStringLiteral("line_3"));
299         QSizePolicy sizePolicy2(QSizePolicy::Minimum, QSizePolicy::Fixed);
300         sizePolicy2.setHorizontalStretch(0);
301         sizePolicy2.setVerticalStretch(0);
302         sizePolicy2.setHeightForWidth(line_3->sizePolicy().hasHeightForWidth());
303         line_3->setSizePolicy(sizePolicy2);
304         line_3->setFrameShape(QFrame::HLine);
305         line_3->setFrameShadow(QFrame::Sunken);
306
307         gridLayout->addWidget(line_3, 7, 0, 1, 2);
308
309         verticalSpacer_4 = new QSpacerItem(20, 6, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
310
311         gridLayout->addItem(verticalSpacer_4, 9, 1, 1, 1);
312
313         verticalSpacer_2 = new QSpacerItem(20, 12, QSizePolicy::Minimum, QSizePolicy::Fixed
    );
314
315         gridLayout->addItem(verticalSpacer_2, 5, 1, 1, 1);
316
317         versionLabel = new QLabel(SettingsWidget);
318         versionLabel->setObjectName(QStringLiteral("versionLabel"));
319         versionLabel->setFont(font);
320
321         gridLayout->addWidget(versionLabel, 0, 0, 1, 2);
322
323         horizontalLayout = new QHBoxLayout();
324         horizontalLayout->setObjectName(QStringLiteral("horizontalLayout"));
325         extraButton = new QPushButton(SettingsWidget);
326         extraButton->setObjectName(QStringLiteral("extraButton"));
327
328         horizontalLayout->addWidget(extraButton);
329
330         horizontalSpacer_2 = new QSpacerItem(40, 20, QSizePolicy::Expanding,
    QSizePolicy::Minimum);
331
332         horizontalLayout->addItem(horizontalSpacer_2);
333
334
335         gridLayout->addLayout(horizontalLayout, 17, 1, 1, 1);
336
337         verticalSpacer_7 = new QSpacerItem(0, 28, QSizePolicy::Minimum, QSizePolicy::Fixed)
    ;
338
339         gridLayout->addItem(verticalSpacer_7, 17, 0, 1, 1);
340
341         QWidget::setTabOrder(layoutBox, capsBox);
342         QWidget::setTabOrder(capsBox, shiftBox);
343         QWidget::setTabOrder(shiftBox, ctrlBox);
344         QWidget::setTabOrder(ctrlBox, winBox);
345         QWidget::setTabOrder(winBox, altBox);
346         QWidget::setTabOrder(altBox, autoFWBox);
347         QWidget::setTabOrder(autoFWBox, loginItemBox);
348         QWidget::setTabOrder(loginItemBox, extraButton);
349         QWidget::setTabOrder(extraButton, pushButton);
350
351         retranslateUi(SettingsWidget);
352
353         QMetaObject::connectSlotsByName(SettingsWidget);
354     } // setupUi
```

Here is the call graph for this function:



Here is the caller graph for this function:



12.102.3 Field Documentation

12.102.3.1 QComboBox* Ui_SettingsWidget::altBox

Definition at line 64 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), setupUi(), and SettingsWidget::updateModifiers().

12.102.3.2 QLabel* Ui_SettingsWidget::altLabel

Definition at line 65 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.3 QCheckBox* Ui_SettingsWidget::autoFWBox

Definition at line 50 of file ui_settingswidget.h.

Referenced by retranslateUi(), SettingsWidget::SettingsWidget(), and setupUi().

12.102.3.4 QComboBox* Ui_SettingsWidget::capsBox

Definition at line 57 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), setupUi(), and SettingsWidget::updateModifiers().

12.102.3.5 QLabel* Ui_SettingsWidget::capsLabel

Definition at line 55 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.6 QComboBox* Ui_SettingsWidget::ctrlBox

Definition at line 60 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), setupUi(), and SettingsWidget::updateModifiers().

12.102.3.7 QLabel* Ui_SettingsWidget::ctrlLabel

Definition at line 58 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.8 QLabel* Ui_SettingsWidget::devicesLabel

Definition at line 38 of file ui_settingswidget.h.

Referenced by retranslateUi(), SettingsWidget::setStatus(), and setupUi().

12.102.3.9 QPushButton* Ui_SettingsWidget::extraButton

Definition at line 74 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.10 QGridLayout* Ui_SettingsWidget::gridLayout

Definition at line 32 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.11 QGridLayout* Ui_SettingsWidget::gridLayout_3

Definition at line 53 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.12 QGridLayout* Ui_SettingsWidget::gridLayout_5

Definition at line 36 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.13 QHBoxLayout* Ui_SettingsWidget::horizontalLayout

Definition at line 73 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.14 QHBoxLayout* Ui_SettingsWidget::horizontalLayout_2

Definition at line 42 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.15 QSpacerItem* Ui_SettingsWidget::horizontalSpacer

Definition at line 44 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.16 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_10

Definition at line 56 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.17 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_2

Definition at line 75 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.18 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_7

Definition at line 37 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.19 QSpacerItem* Ui_SettingsWidget::horizontalSpacer_9

Definition at line 59 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.20 QLabel* Ui_SettingsWidget::label

Definition at line 34 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.21 QLabel* Ui_SettingsWidget::label_2

Definition at line 33 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.22 QLabel* Ui_SettingsWidget::label_3

Definition at line 43 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.23 QLabel* Ui_SettingsWidget::label_5

Definition at line 39 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.24 QLabel* Ui_SettingsWidget::label_6

Definition at line 35 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.25 QLabel* Ui_SettingsWidget::label_8

Definition at line 52 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.26 QComboBox* Ui_SettingsWidget::layoutBox

Definition at line 40 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), setupUi(), and SettingsWidget::showLayoutDialog().

12.102.3.27 QFrame* Ui_SettingsWidget::line

Definition at line 66 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.28 QFrame* Ui_SettingsWidget::line_2

Definition at line 51 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.29 QFrame* Ui_SettingsWidget::line_3

Definition at line 69 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.30 QCheckBox* Ui_SettingsWidget::loginItemBox

Definition at line 48 of file ui_settingswidget.h.

Referenced by retranslateUi(), SettingsWidget::SettingsWidget(), and setupUi().

12.102.3.31 QPushButton* Ui_SettingsWidget::pushButton

Definition at line 45 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.32 QComboBox* Ui_SettingsWidget::shiftBox

Definition at line 61 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), setupUi(), and SettingsWidget::updateModifiers().

12.102.3.33 QLabel* Ui_SettingsWidget::shiftLabel

Definition at line 54 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

12.102.3.34 QLabel* Ui_SettingsWidget::versionLabel

Definition at line 72 of file ui_settingswidget.h.

Referenced by retranslateUi(), setupUi(), and SettingsWidget::setVersion().

12.102.3.35 QSpacerItem* Ui_SettingsWidget::verticalSpacer

Definition at line 46 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.36 QSpacerItem* Ui_SettingsWidget::verticalSpacer_10

Definition at line 41 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.37 QSpacerItem* Ui_SettingsWidget::verticalSpacer_2

Definition at line 71 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.38 QSpacerItem* Ui_SettingsWidget::verticalSpacer_3

Definition at line 67 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.39 QSpacerItem* Ui_SettingsWidget::verticalSpacer_4

Definition at line 70 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.40 QSpacerItem* Ui_SettingsWidget::verticalSpacer_5

Definition at line 68 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.41 QSpacerItem* Ui_SettingsWidget::verticalSpacer_6

Definition at line 49 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.42 QSpacerItem* Ui_SettingsWidget::verticalSpacer_7

Definition at line 76 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.43 QSpacerItem* Ui_SettingsWidget::verticalSpacer_8

Definition at line 47 of file ui_settingswidget.h.

Referenced by setupUi().

12.102.3.44 QComboBox* Ui_SettingsWidget::winBox

Definition at line 63 of file ui_settingswidget.h.

Referenced by SettingsWidget::SettingsWidget(), setupUi(), and SettingsWidget::updateModifiers().

12.102.3.45 QLabel* Ui_SettingsWidget::winLabel

Definition at line 62 of file ui_settingswidget.h.

Referenced by retranslateUi(), and setupUi().

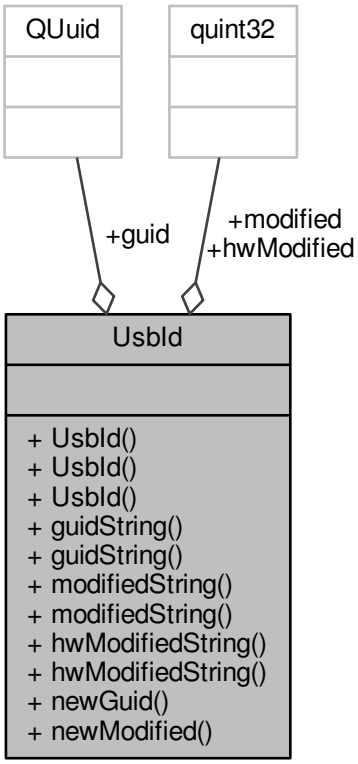
The documentation for this class was generated from the following file:

- src/ckb/ui_settingswidget.h

12.103 UsbId Struct Reference

`#include <src/ckb/kbmode.h>`

Collaboration diagram for UsbId:



Public Member Functions

- UsbId (const QString &_guid, quint32 _modified)
- UsbId (const QString &_guid, const QString &_modified)
- UsbId ()
- QString guidString () const
- void guidString (const QString &newGuid)
- QString modifiedString () const
- void modifiedString (const QString &newModified)
- QString hwModifiedString () const
- void hwModifiedString (const QString &newModified)
- void newGuid ()
- void newModified ()

Data Fields

- QUuid guid
- quint32 modified
- quint32 hwModified

12.103.1 Detailed Description

Definition at line 11 of file kbmode.h.

12.103.2 Constructor & Destructor Documentation

12.103.2.1 UsbId::UsbId (const QString & *_guid,* quint32 *_modified*) [inline]

Definition at line 16 of file kbmode.h.

```
16 : guid(_guid), modified(_modified), hwModified(_modified) {}
```

12.103.2.2 UsbId::UsbId (const QString & *_guid,* const QString & *_modified*) [inline]

Definition at line 17 of file kbmode.h.

```
17 : guid(_guid), modified(_modified.toUInt(0, 16)), hwModified(
      modified) {}
```

12.103.2.3 UsbId::UsbId ( ) [inline]

Definition at line 18 of file kbmode.h.

```
18 : guid(QUuid::createUuid()),modified(0) {}
```

12.103.3 Member Function Documentation

12.103.3.1 QString UsbId::guidString ( ) const [inline]

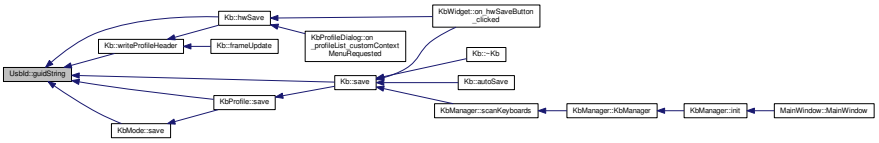
Definition at line 20 of file kbmode.h.

References guid.

Referenced by Kb::hwSave(), KbProfile::save(), KbMode::save(), Kb::save(), and Kb::writeProfileHeader().

```
20 { return guid.toString().toUpper(); }
```

Here is the caller graph for this function:



12.103.3.2 void UsbId::guidString (const QString & *newGuid*) [inline]

Definition at line 21 of file kbmode.h.

References guid, and newGuid().

```
21 { guid = newGuid; }
```

Here is the call graph for this function:

12.103.3.3 QString UsbId::hwModifiedString ( ) const [inline]

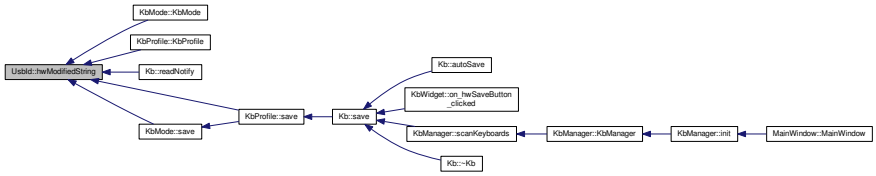
Definition at line 24 of file kbmode.h.

References hwModified.

Referenced by KbMode::KbMode(), KbProfile::KbProfile(), Kb::readNotify(), KbProfile::save(), and KbMode::save().

```
24 { return QString::number(hwModified, 16); }
```

Here is the caller graph for this function:



12.103.3.4 void UsbId::hwModifiedString (const QString & *newModified*) [inline]

Definition at line 25 of file kbmode.h.

References hwModified.

```
25 { hwModified = newModified.toUInt(0, 16); }
```

12.103.3.5 QString UsbId::modifiedString ( ) const [inline]

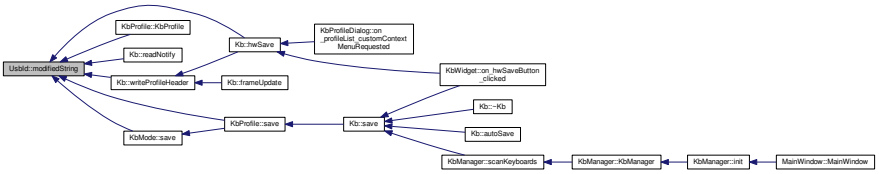
Definition at line 22 of file kbmode.h.

References modified.

Referenced by Kb::hwSave(), KbProfile::KbProfile(), Kb::readNotify(), KbProfile::save(), KbMode::save(), and Kb::writeProfileHeader().

```
22 { return QString::number(modified, 16); }
```

Here is the caller graph for this function:



12.103.3.6 void UsbId::modifiedString (const QString & *newModified*) [inline]

Definition at line 23 of file kbmode.h.

References modified.

```
23 { modified = newModified.toUInt(0, 16); }
```

12.103.3.7 void UsbId::newGuid ( ) [inline]

Definition at line 28 of file kbmode.h.

References guid.

Referenced by guidString().

```
28 { guid = QUuid::createUuid(); }
```

Here is the caller graph for this function:

12.103.3.8 void UsbId::newModified ( ) [inline]

Definition at line 29 of file kbmode.h.

References modified.

Referenced by KbProfile::save(), and KbMode::save().

```
29 { quint32 newMod; do { newMod = qrand(); } while(newMod == modified);
      modified = newMod; }
```

Here is the caller graph for this function:

12.103.4 Field Documentation

12.103.4.1 QUuid UsbId::guid

Definition at line 12 of file kbmode.h.

Referenced by KbProfile::find(), Kb::find(), guidString(), KbMode::KbMode(), KbProfile::KbProfile(), newGuid(), KbWidget::on_modesList_customContextMenuRequested(), KbWidget::on_modesList_itemChanged(), KbWidget::on_modesList_itemClicked(), KbProfileDialog::on_profileList_customContextMenuRequested(), KbProfileDialog::on_profileList_itemChanged(), KbProfileDialog::on_profileList_itemClicked(), KbWidget::profileChanged(), Kb::readNotify(), and KbProfileDialog::repopulate().

12.103.4.2 quint32 UsbId::hwModified

Definition at line 14 of file kbmode.h.

Referenced by hwModifiedString(), Kb::hwSave(), KbMode::KbMode(), and KbProfile::KbProfile().

12.103.4.3 quint32 UsbId::modified

Definition at line 13 of file kbmode.h.

Referenced by Kb::hwSave(), KbMode::KbMode(), KbProfile::KbProfile(), modifiedString(), and newModified().

The documentation for this struct was generated from the following file:

- src/ckb/kbmode.h

Chapter 13

File Documentation

13.1 BUILD.md File Reference

13.2 DAEMON.md File Reference

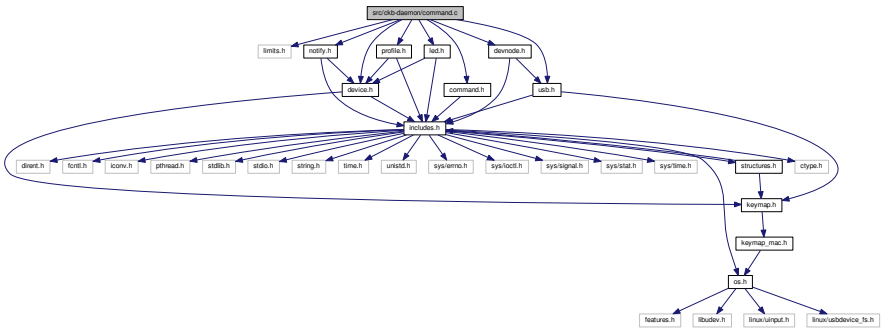
13.3 README.md File Reference

13.4 ROADMAP.md File Reference

13.5 src/ckb-daemon/command.c File Reference

```
#include <limits.h>
#include "command.h"
#include "device.h"
#include "devnode.h"
#include "led.h"
#include "notify.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for command.c:



Macros

- #define TRY_WITH_RESET(action)

Functions

- int readcmd (usbdevice *kb, const char *line)

Variables

- static const char *const cmd_strings [(CMD_LAST-CMD_FIRST+2)-1]

13.5.1 Macro Definition Documentation

13.5.1.1 #define TRY_WITH_RESET(*action*)

Value:

```
while(action){              \
        if(usb_tryreset(kb)){   \
            free(word);         \
            return 1;           \
        }                       \
    }
```

Definition at line 59 of file command.c.

Referenced by readcmd().

13.5.2 Function Documentation

13.5.2.1 int readcmd (usbdevice * *kb*, const char * *line*)

< Because length of word is length of line + 1, there should be no problem with buffer overflow.

Definition at line 68 of file command.c.

References ACCEL, ACTIVE, usbdevice::active, BIND, CMD_COUNT, CMD_FIRST, cmd_strings, usbprofile::currentmode, DELAY, usbdevice::delay, DITHER, usbdevice::dither, devcmd::do_cmd, devcmd::do_io, devcmd::do_macro, DPI, DPISEL, ERASE, ERASEPROFILE, FEAT_ANSI, FEAT_BIND, FEAT_ISO, FEAT_LMASK, FEAT_MOUSEACCEL, FEAT_NOTIFY, usbdevice::features, lighting::forceupdate, FPS, FWUPDATE, GET, HAS_FEATURES, HWLOAD, HWSAVE, IAUTO, ID, IDLE, INDEX_OF, INOTIFY, IOFF, ION, IS_FULLRANGE, IS_MOUSE_DEV, keymap, LAYOUT, left, LIFT, usbmode::light, MACRO, mknotifynode(), MODE, usbprofile::mode, MODE_COUNT, N_KEYS_EXTENDED, NAME, NEEDS_FW_UPDATE, NONE, NOTIFY, NOTIFYOFF, NOTIFYON, OUTFIFO_MAX, POLLRATE, usbdevice::profile, PROFILEID, PROFILENAME, REBIND, RESTART, RGB, right(), rmnotifynode(), SCROLL_ACCELERATED, SCROLL_MAX, SCROLL_MIN, SCROLLSPEED, SNAP, SWITCH, TRY_WITH_RESET, UNBIND, usbdevice::usbdelay, and usbdevice::vtable.

Referenced by devmain().

```
68                                          {
69     char* word = malloc(strlen(line) + 1);
70     int wordlen;
71     const char* newline = 0;
72     const devcmd* vt = kb->vtable;
73     usbprofile* profile = kb->profile;
74     usbmode* mode = 0;
75     int notifynumber = 0;
76     // Read words from the input
77     cmd command = NONE;
78     while(sscanf(line, "%s%n", word, &wordlen) == 1){
79         line += wordlen;
80         // If we passed a newline, reset the context
81         if(line > newline){
82             mode = profile->currentmode;
83             command = NONE;
84             notifynumber = 0;
85             newline = strchr(line, '\n');
86             if(!newline)
87                 newline = line + strlen(line);
```

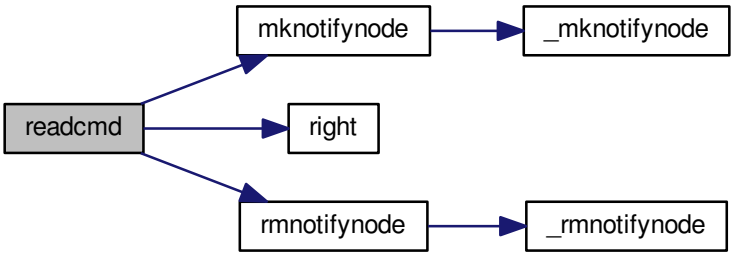
```
88          }
89          // Check for a command word
90          for(int i = 0; i < CMD_COUNT - 1; i++){
91              if(!strcmp(word, cmd_strings[i])){
92                  command = i + CMD_FIRST;
93  #ifndef OS_MAC
94                  // Layout and mouse acceleration aren't used on Linux; ignore
95                  if(command == LAYOUT || command == ACCEL || command ==
      SCROLLSPEED)
96                      command = NONE;
97  #endif
98                  // Most commands require parameters, but a few are actions in and of themselves
99                  if(command != SWITCH
100                         && command != HWLOAD && command != HWSAVE
101                         && command != ACTIVE && command != IDLE
102                         && command != ERASE && command != ERASEPROFILE
103                         && command != RESTART)
104                     goto next_loop;
105                 break;
106             }
107         }
108
109         // Set current notification node when given @number
110         int newnotify;
111         if(sscanf(word, "@%u", &newnotify) == 1 && newnotify < OUTFIFO_MAX){
112             notifynumber = newnotify;
113             continue;
114         }
115
116         // Reject unrecognized commands. Reject bind or notify related commands if the keyboard doesn't
       have the feature enabled.
117         if(command == NONE
118                 || ((!HAS_FEATURES(kb, FEAT_BIND) && (command ==
      BIND || command == UNBIND || command == REBIND || command ==
      MACRO || command == DELAY))
119                         || (!HAS_FEATURES(kb, FEAT_NOTIFY) && command ==
      NOTIFY))){
120             next_loop:
121             continue;
122         }
123         // Reject anything not related to fwupdate if device has a bricked FW
124         if(NEEDS_FW_UPDATE(kb) && command != FWUPDATE && command !=
      NOTIFYON && command != NOTIFYOFF)
125             continue;
126
127         // Specially handled commands - these are available even when keyboard is IDLE
128         switch(command){
129         case NOTIFYON: {
130             // Notification node on
131             int notify;
132             if(sscanf(word, "%u", &notify) == 1)
133                 mknotifynode(kb, notify);
134             continue;
135         } case NOTIFYOFF: {
136             // Notification node off
137             int notify;
138             if(sscanf(word, "%u", &notify) == 1 && notify != 0) // notify0 can't be removed
139                 rmnotifynode(kb, notify);
140             continue;
141         } case GET:
142             // Output data to notification node
143             vt->get(kb, mode, notifynumber, 0, word);
144             continue;
145         case LAYOUT:
146             // OSX: switch ANSI/ISO keyboard layout
147             if(!strcmp(word, "ansi"))
148                 kb->features = (kb->features & ~FEAT_LMASK) |
      FEAT_ANSI;
149             else if(!strcmp(word, "iso"))
150                 kb->features = (kb->features & ~FEAT_LMASK) |
      FEAT_ISO;
151             continue;
152 #ifdef OS_MAC
153         case ACCEL:
154             // OSX mouse acceleration on/off
155             if(!strcmp(word, "on"))
156                 kb->features |= FEAT_MOUSEACCEL;
157             else if(!strcmp(word, "off"))
158                 kb->features &= ~FEAT_MOUSEACCEL;
159             continue;
160         case SCROLLSPEED:{
161             int newscroll;
162             if(sscanf(word, "%d", &newscroll) != 1)
163                 break;
164             if(newscroll < SCROLL_MIN)
165                 newscroll = SCROLL_ACCELERATED;
166             if(newscroll > SCROLL_MAX)
```

```
167                 newscroll = SCROLL_MAX;
168             kb->scroll_rate = newscroll;
169             continue;
170         }
171 #endif
172         case MODE: {
173             // Select a mode number (1 - 6)
174             int newmode;
175             if(sscanf(word, "%u", &newmode) == 1 && newmode > 0 && newmode <=
    MODE_COUNT)
176                 mode = profile->mode + newmode - 1;
177             continue;
178         }
179         case FPS: {
180             // USB command delay (2 - 10ms)
181             uint framerate;
182             if(sscanf(word, "%u", &framerate) == 1 && framerate > 0){
183                 // Not all devices require the same number of messages per frame; select delay
     appropriately
184                 uint per_frame = IS_MOUSE_DEV(kb) ? 2 : IS_FULLRANGE(kb) ? 14 : 5;
185                 uint delay = 1000 / framerate / per_frame;
186                 if(delay < 2)
187                     delay = 2;
188                 else if(delay > 10)
189                     delay = 10;
190                 kb->usbdelay = delay;
191             }
192             continue;
193         }
194         case DITHER: {
195             // 0: No dither, 1: Ordered dither.
196             uint dither;
197             if(sscanf(word, "%u", &dither) == 1 && dither <= 1){
198                 kb->dither = dither;
199                 profile->currentmode->light.forceupdate = 1;
200                 mode->light.forceupdate = 1;
201             }
202             continue;
203         }
204         case DELAY: {
205             long int delay;
206             if(sscanf(word, "%ld", &delay) == 1 && 0 <= delay && delay < UINT_MAX) {
207                 // Add delay of `newdelay` microseconds to macro playback
208                 kb->delay = (unsigned int)delay;
209             } else if(strcmp(word, "on") == 0) {
210                 // allow previous syntax, `delay on` means use old `long macro delay`
211                 kb->delay = UINT_MAX;
212             } else {
213                 // bad parameter to handle false commands like "delay off"
214                 kb->delay = 0; // No delay.
215             }
216             continue;
217         }
218         case RESTART: {
219             char mybuffer[] = "no reason specified";
220             if (sscanf(line, " %[^\n]", word) == -1) {
221                 word = mybuffer;
222             }
223             vt->do_cmd[command](kb, mode, notifynumber, 0, word);
224             continue;
225         }
226         default:;
227         }
228 
229         // If a keyboard is inactive, it must be activated before receiving any other commands
230         if(!kb->active){
231             if(command == ACTIVE)
232                 TRY_WITH_RESET(vt->active(kb, mode, notifynumber, 0, 0));
233             continue;
234         }
235         // Specially handled commands only available when keyboard is ACTIVE
236         switch(command){
237         case IDLE:
238             TRY_WITH_RESET(vt->idle(kb, mode, notifynumber, 0, 0));
239             continue;
240         case SWITCH:
241             if(profile->currentmode != mode){
242                 profile->currentmode = mode;
243                 // Set mode light for non-RGB K95
244                 int index = INDEX_OF(mode, profile->mode);
245                 vt->setmodeindex(kb, index);
246             }
247             continue;
248         case HWLOAD: case HWSAVE:{
249             char delay = kb->usbdelay;
250             // Ensure delay of at least 10ms as the device can get overwhelmed otherwise
251             if(delay < 10)
```

```
252                 kb->usbdelay = 10;
253             // Try to load/save the hardware profile. Reset on failure, disconnect if reset fails.
254             TRY_WITH_RESET(vt->do_io[command](kb, mode, notifynumber, 1, 0));
255             // Re-send the current RGB state as it sometimes gets scrambled
256             TRY_WITH_RESET(vt->updatergb(kb, 1));
257             kb->usbdelay = delay;
258             continue;
259         }
260         case FWUPDATE:
261             // FW update parses a whole word. Unlike hwload/hwsave, there's no try again on failure.
262             if(vt->fwupdate(kb, mode, notifynumber, 0, word)){
263                 free(word);
264                 return 1;
265             }
266             continue;
267         case POLLRATE: {
268             uint rate;
269             if(sscanf(word, "%u", &rate) == 1 && (rate == 1 || rate == 2 || rate == 4 || rate == 8))
270                 TRY_WITH_RESET(vt->pollrate(kb, mode, notifynumber, rate, 0));
271             continue;
272         }
273         case ERASEPROFILE:
274             // Erase the current profile
275             vt->eraseprofile(kb, mode, notifynumber, 0, 0);
276             // Update profile/mode pointers
277             profile = kb->profile;
278             mode = profile->currentmode;
279             continue;
280         case ERASE: case NAME: case IOFF: case ION: case IAUTO: case
    INOTIFY: case PROFILENAME: case ID: case PROFILEID: case
    DPISEL: case LIFT: case SNAP:
281             // All of the above just parse the whole word
282             vt->do_cmd[command](kb, mode, notifynumber, 0, word);
283             continue;
284         case RGB: {
285             // RGB command has a special response for a single hex constant
286             int r, g, b;
287             if(sscanf(word, "%02x%02x%02x", &r, &g, &b) == 3){
288                 // Set all keys
289                 for(int i = 0; i < N_KEYS_EXTENDED; i++)
290                     vt->rgb(kb, mode, notifynumber, i, word);
291                 continue;
292             }
293             break;
294         }
295         case MACRO:
296             if(!strcmp(word, "clear")){
297                 // Macro has a special clear command
298                 vt->macro(kb, mode, notifynumber, 0, 0);
299                 continue;
300             }
301             break;
302         default:;
303         }
304         // For anything else, split the parameter at the colon
305         int left = -1;
306         sscanf(word, "%*[^:]%n", &left);
307         if(left <= 0)
308             continue;
309         const char* right = word + left;
310         if(right[0] == ':')
311             right++;
312         // Macros and DPI have a separate left-side handler
313         if(command == MACRO || command == DPI){
314             word[left] = 0;
315             vt->do_macro[command](kb, mode, notifynumber, word, right);
316             continue;
317         }
318         // Scan the left side for key names and run the requested command
319         int position = 0, field = 0;
320         char keyname[11];
321         while(position < left && sscanf(word + position, "%10[^:,]%n", keyname, &field) == 1){
322             int keycode;
323             if(!strcmp(keyname, "all")){
324                 // Set all keys
325                 for(int i = 0; i < N_KEYS_EXTENDED; i++)
326                     vt->do_cmd[command](kb, mode, notifynumber, i, right);
327             } else if((sscanf(keyname, "#%d", &keycode) && keycode >= 0 && keycode <
    N_KEYS_EXTENDED)
328                       || (sscanf(keyname, "#x%x", &keycode) && keycode >= 0 && keycode <
    N_KEYS_EXTENDED)){
329                 // Set a key numerically
330                 vt->do_cmd[command](kb, mode, notifynumber, keycode, right);
331             } else {
332                 // Find this key in the keymap
333                 for(unsigned i = 0; i < N_KEYS_EXTENDED; i++){
334                     if(keymap[i].name && !strcmp(keyname, keymap[i].name)){
```

```
335                         vt->do_cmd[command](kb, mode, notifynumber, i,
    right);
336                         break;
337                     }
338                 }
339             }
340             if(word[position += field] == ',')
341                 position++;
342         }
343     }
344
345     // Finish up
346     if(!NEEDS_FW_UPDATE(kb)){
347         TRY_WITH_RESET(vt->updatergb(kb, 0));
348         TRY_WITH_RESET(vt->updatedpi(kb, 0));
349     }
350     free(word);
351     return 0;
352 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.5.3 Variable Documentation

13.5.3.1 const char* const cmd_strings[(CMD_LAST-CMD_FIRST+2)-1] [static]

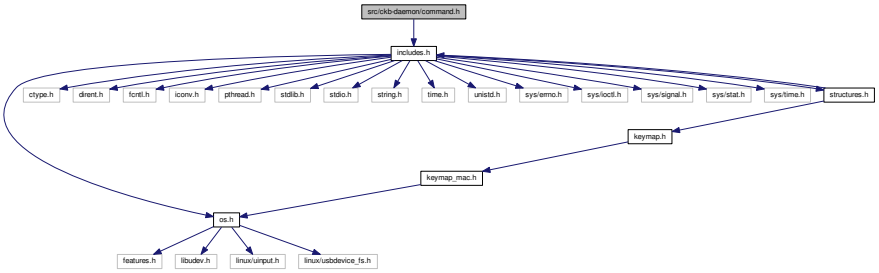
Definition at line 10 of file command.c.

Referenced by readcmd().

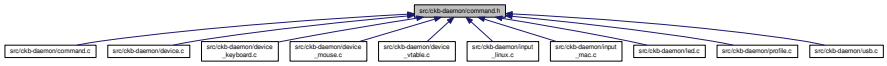
13.6 src/ckb-daemon/command.h File Reference

```
#include "includes.h"
```

Include dependency graph for command.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- union devcmd
- struct devcmd.__unnamed__

Macros

- #define CMD_COUNT (CMD_LAST - CMD_FIRST + 2)
- #define CMD_DEV_COUNT (CMD_LAST - CMD_VT_FIRST + 1)

Typedefs

- typedef void(∗ cmdhandler)(usbdevice ∗kb, usbmode ∗modeidx, int notifyidx, int keyindex, const char ∗parameter)
- typedef int(∗ cmdhandler_io)(usbdevice ∗kb, usbmode ∗modeidx, int notifyidx, int keyindex, const char ∗parameter)
- typedef void(∗ cmdhandler_mac)(usbdevice ∗kb, usbmode ∗modeidx, int notifyidx, const char ∗keys, const char ∗assignment)
- typedef union devcmd devcmd

Enumerations

- enum cmd {
  NONE = -11, DELAY = -10, CMD_FIRST = DELAY, MODE = -9,
  SWITCH = -8, LAYOUT = -7, ACCEL = -6, SCROLLSPEED = -5,
  NOTIFYON = -4, NOTIFYOFF = -3, FPS = -2, DITHER = -1,
  HWLOAD = 0, CMD_VT_FIRST = 0, HWSAVE, FWUPDATE,
  POLLRATE, ACTIVE, IDLE, ERASE,
  ERASEPROFILE, NAME, PROFILENAME, ID,
  PROFILEID, RGB, IOFF, ION,
  IAUTO, BIND, UNBIND, REBIND,
  MACRO, DPI, DPISEL, LIFT,
  SNAP, NOTIFY, INOTIFY, GET,
  RESTART, CMD_LAST = RESTART }

Functions

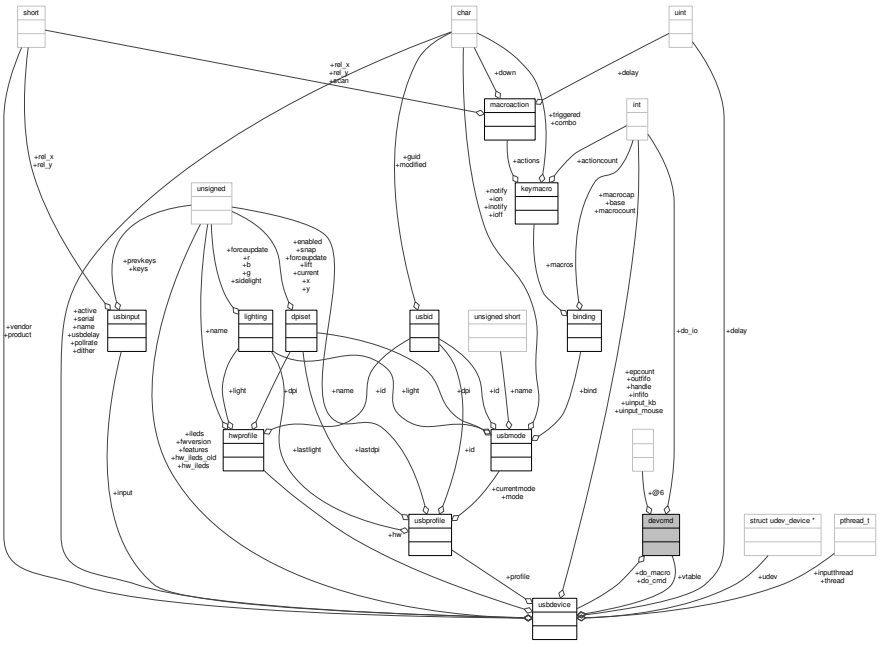
- int readcmd (usbdevice *kb, const char *line)

13.6.1 Data Structure Documentation

13.6.1.1 union devcmd

Definition at line 73 of file command.h.

Collaboration diagram for devcmd:



Data Fields

struct devcmd	__unnamed__	
cmdhandler	do_cmd[(CMD_-LAST-CMD_VT-_FIRST+1)]	
cmdhandler_io	do_io[(CMD_LA-ST-CMD_VT_FI-RST+1)]	
cmdhandler_-mac	do_macro[(CM-D_LAST-CMD_-VT_FIRST+1)]	

13.6.2 Macro Definition Documentation

13.6.2.1 #define CMD_COUNT (CMD_LAST - CMD_FIRST + 2)

Definition at line 65 of file command.h.

Referenced by readcmd().

13.6.2.2 #define CMD_DEV_COUNT (CMD_LAST - CMD_VT_FIRST + 1)

Definition at line 66 of file command.h.

13.6.3 Typedef Documentation

13.6.3.1 typedef void(∗ cmdhandler)(usbdevice ∗kb, usbmode ∗modeidx, int notifyidx, int keyindex, const char ∗parameter)

Definition at line 70 of file command.h.

13.6.3.2 typedef int(∗ cmdhandler_io)(usbdevice ∗kb, usbmode ∗modeidx, int notifyidx, int keyindex, const char ∗parameter)

Definition at line 71 of file command.h.

13.6.3.3 typedef void(∗ cmdhandler_mac)(usbdevice ∗kb, usbmode ∗modeidx, int notifyidx, const char ∗keys, const char ∗assignment)

Definition at line 72 of file command.h.

13.6.3.4 typedef union devcmd devcmd

13.6.4 Enumeration Type Documentation

13.6.4.1 enum cmd

Enumerator

NONE

DELAY

CMD_FIRST

MODE

SWITCH

LAYOUT

ACCEL

SCROLLSPEED

NOTIFYON

NOTIFYOFF

FPS

DITHER

HWLOAD

CMD_VT_FIRST

HWSAVE

FWUPDATE

POLLRATE

ACTIVE

IDLE

ERASE

ERASEPROFILE

NAME

PROFILENAME

ID

PROFILEID

RGB

IOFF

ION

IAUTO

BIND

UNBIND

REBIND

MACRO

DPI

DPISEL

LIFT

SNAP

NOTIFY

INOTIFY

GET

RESTART

CMD_LAST

Definition at line 7 of file command.h.

```
7             {
8     // Special - handled by readcmd, no device functions
9     NONE        = -11,
10     DELAY       = -10,   CMD_FIRST = DELAY,
11     MODE        = -9,
12     SWITCH      = -8,
13     LAYOUT      = -7,
14     ACCEL       = -6,
15     SCROLLSPEED = -5,
16     NOTIFYON    = -4,
17     NOTIFYOFF   = -3,
18     FPS         = -2,
19     DITHER      = -1,
20
21     // Hardware data
22     HWLOAD      = 0,    CMD_VT_FIRST = 0,
23     HWSAVE,
24     FWUPDATE,
25     POLLRATE,
26
27     // Software control on/off
28     ACTIVE,
29     IDLE,
30
31     // Profile/mode metadata
32     ERASE,
33     ERASEPROFILE,
34     NAME,
35     PROFILENAME,
36     ID,
37     PROFILEID,
38
39     // LED control
40     RGB,
41     IOFF,
42     ION,
43     IAUTO,
44
45     // Key binding control
46     BIND,
```

```
47     UNBIND,
48     REBIND,
49     MACRO,
50
51     // DPI control
52     DPI,
53     DPISEL,
54     LIFT,
55     SNAP,
56
57     // Notifications and output
58     NOTIFY,
59     INOTIFY,
60     GET,
61     RESTART,
62
63     CMD_LAST = RESTART
64 } cmd;
```

13.6.5 Function Documentation

13.6.5.1 int readcmd (usbdevice * *kb*, const char * *line*)

< Because length of word is length of line + 1, there should be no problem with buffer overflow.

Definition at line 68 of file command.c.

References ACCEL, ACTIVE, usbdevice::active, BIND, CMD_COUNT, CMD_FIRST, cmd_strings, usbprofile::currentmode, DELAY, usbdevice::delay, DITHER, usbdevice::dither, devcmd::do_cmd, devcmd::do_io, devcmd::do_macro, DPI, DPISEL, ERASE, ERASEPROFILE, FEAT_ANSI, FEAT_BIND, FEAT_ISO, FEAT_LMASK, FEAT_MOUSEACCEL, FEAT_NOTIFY, usbdevice::features, lighting::forceupdate, FPS, FWUPDATE, GET, HAS_FEATURES, HWLOAD, HWSAVE, IAUTO, ID, IDLE, INDEX_OF, INOTIFY, IOFF, ION, IS_FULLRANGE, IS_MOUSE_DEV, keymap, LAYOUT, left, LIFT, usbmode::light, MACRO, mknotifynode(), MODE, usbprofile::mode, MODE_COUNT, N_KEYS_EXTENDED, NAME, NEEDS_FW_UPDATE, NONE, NOTIFY, NOTIFYOFF, NOTIFYON, OUTFIFO_MAX, POLLRATE, usbdevice::profile, PROFILEID, PROFILENAME, REBIND, RESTART, RGB, right(), rmnotifynode(), SCROLL_ACCELERATED, SCROLL_MAX, SCROLL_MIN, SCROLLSPEED, SNAP, SWITCH, TRY_WITH_RESET, UNBIND, usbdevice::usbdelay, and usbdevice::vtable.

Referenced by devmain().

```
68                                          {
69     char* word = malloc(strlen(line) + 1);
70     int wordlen;
71     const char* newline = 0;
72     const devcmd* vt = kb->vtable;
73     usbprofile* profile = kb->profile;
74     usbmode* mode = 0;
75     int notifynumber = 0;
76     // Read words from the input
77     cmd command = NONE;
78     while(sscanf(line, "%s%n", word, &wordlen) == 1){
79         line += wordlen;
80         // If we passed a newline, reset the context
81         if(line > newline){
82             mode = profile->currentmode;
83             command = NONE;
84             notifynumber = 0;
85             newline = strchr(line, '\n');
86             if(!newline)
87                 newline = line + strlen(line);
88         }
89         // Check for a command word
90         for(int i = 0; i < CMD_COUNT - 1; i++){
91             if(!strcmp(word, cmd_strings[i])){
92                 command = i + CMD_FIRST;
93 #ifndef OS_MAC
94                 // Layout and mouse acceleration aren't used on Linux; ignore
95                 if(command == LAYOUT || command == ACCEL || command ==
    SCROLLSPEED)
96                     command = NONE;
97 #endif
98                 // Most commands require parameters, but a few are actions in and of themselves
99                 if(command != SWITCH
100                         && command != HWLOAD && command != HWSAVE
101                         && command != ACTIVE && command != IDLE
102                         && command != ERASE && command != ERASEPROFILE
103                         && command != RESTART)
```

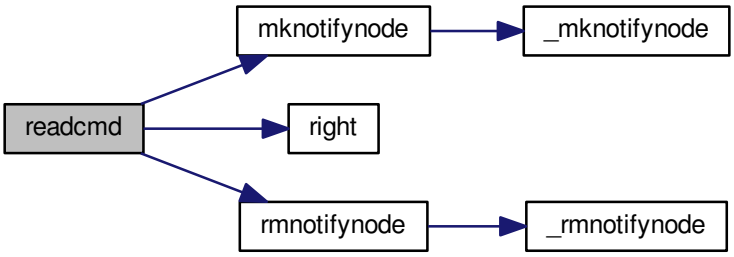
```
104                     goto next_loop;
105                 break;
106             }
107         }
108 
109         // Set current notification node when given @number
110         int newnotify;
111         if(sscanf(word, "@%u", &newnotify) == 1 && newnotify < OUTFIFO_MAX){
112             notifynumber = newnotify;
113             continue;
114         }
115 
116         // Reject unrecognized commands. Reject bind or notify related commands if the keyboard doesn't
     have the feature enabled.
117         if(command == NONE
118                 || ((!HAS_FEATURES(kb, FEAT_BIND) && (command ==
    BIND || command == UNBIND || command == REBIND || command ==
    MACRO || command == DELAY))
119                         || (!HAS_FEATURES(kb, FEAT_NOTIFY) && command ==
    NOTIFY))){
120             next_loop:
121             continue;
122         }
123         // Reject anything not related to fwupdate if device has a bricked FW
124         if(NEEDS_FW_UPDATE(kb) && command != FWUPDATE && command !=
    NOTIFYON && command != NOTIFYOFF)
125             continue;
126 
127         // Specially handled commands - these are available even when keyboard is IDLE
128         switch(command){
129         case NOTIFYON: {
130             // Notification node on
131             int notify;
132             if(sscanf(word, "%u", &notify) == 1)
133                 mknotifynode(kb, notify);
134             continue;
135         } case NOTIFYOFF: {
136             // Notification node off
137             int notify;
138             if(sscanf(word, "%u", &notify) == 1 && notify != 0) // notify0 can't be removed
139                 rmnotifynode(kb, notify);
140             continue;
141         } case GET:
142             // Output data to notification node
143             vt->get(kb, mode, notifynumber, 0, word);
144             continue;
145         case LAYOUT:
146             // OSX: switch ANSI/ISO keyboard layout
147             if(!strcmp(word, "ansi"))
148                 kb->features = (kb->features & ~FEAT_LMASK) |
    FEAT_ANSI;
149             else if(!strcmp(word, "iso"))
150                 kb->features = (kb->features & ~FEAT_LMASK) |
    FEAT_ISO;
151             continue;
152 #ifdef OS_MAC
153         case ACCEL:
154             // OSX mouse acceleration on/off
155             if(!strcmp(word, "on"))
156                 kb->features |= FEAT_MOUSEACCEL;
157             else if(!strcmp(word, "off"))
158                 kb->features &= ~FEAT_MOUSEACCEL;
159             continue;
160         case SCROLLSPEED:{
161             int newscroll;
162             if(sscanf(word, "%d", &newscroll) != 1)
163                 break;
164             if(newscroll < SCROLL_MIN)
165                 newscroll = SCROLL_ACCELERATED;
166             if(newscroll > SCROLL_MAX)
167                 newscroll = SCROLL_MAX;
168             kb->scroll_rate = newscroll;
169             continue;
170         }
171 #endif
172         case MODE: {
173             // Select a mode number (1 - 6)
174             int newmode;
175             if(sscanf(word, "%u", &newmode) == 1 && newmode > 0 && newmode <=
    MODE_COUNT)
176                 mode = profile->mode + newmode - 1;
177             continue;
178         }
179         case FPS: {
180             // USB command delay (2 - 10ms)
181             uint framerate;
182             if(sscanf(word, "%u", &framerate) == 1 && framerate > 0){
```

```
183                 // Not all devices require the same number of messages per frame; select delay
    appropriately
184                 uint per_frame = IS_MOUSE_DEV(kb) ? 2 : IS_FULLRANGE(kb) ? 14 : 5;
185                 uint delay = 1000 / framerate / per_frame;
186                 if(delay < 2)
187                     delay = 2;
188                 else if(delay > 10)
189                     delay = 10;
190                 kb->usbdelay = delay;
191             }
192             continue;
193         }
194         case DITHER: {
195             // 0: No dither, 1: Ordered dither.
196             uint dither;
197             if(sscanf(word, "%u", &dither) == 1 && dither <= 1){
198                 kb->dither = dither;
199                 profile->currentmode->light.forceupdate = 1;
200                 mode->light.forceupdate = 1;
201             }
202             continue;
203         }
204         case DELAY: {
205             long int delay;
206             if(sscanf(word, "%ld", &delay) == 1 && 0 <= delay && delay < UINT_MAX) {
207                 // Add delay of `newdelay` microseconds to macro playback
208                 kb->delay = (unsigned int)delay;
209             } else if(strcmp(word, "on") == 0) {
210                 // allow previous syntax, `delay on` means use old `long macro delay`
211                 kb->delay = UINT_MAX;
212             } else {
213                 // bad parameter to handle false commands like "delay off"
214                 kb->delay = 0; // No delay.
215             }
216             continue;
217         }
218         case RESTART: {
219             char mybuffer[] = "no reason specified";
220             if (sscanf(line, " %[^\n]", word) == -1) {
221                 word = mybuffer;
222             }
223             vt->do_cmd[command](kb, mode, notifynumber, 0, word);
224             continue;
225         }
226         default:;
227         }
228
229         // If a keyboard is inactive, it must be activated before receiving any other commands
230         if(!kb->active){
231             if(command == ACTIVE)
232                 TRY_WITH_RESET(vt->active(kb, mode, notifynumber, 0, 0));
233             continue;
234         }
235         // Specially handled commands only available when keyboard is ACTIVE
236         switch(command){
237         case IDLE:
238             TRY_WITH_RESET(vt->idle(kb, mode, notifynumber, 0, 0));
239             continue;
240         case SWITCH:
241             if(profile->currentmode != mode){
242                 profile->currentmode = mode;
243                 // Set mode light for non-RGB K95
244                 int index = INDEX_OF(mode, profile->mode);
245                 vt->setmodeindex(kb, index);
246             }
247             continue;
248         case HWLOAD: case HWSAVE:{
249             char delay = kb->usbdelay;
250             // Ensure delay of at least 10ms as the device can get overwhelmed otherwise
251             if(delay < 10)
252                 kb->usbdelay = 10;
253             // Try to load/save the hardware profile. Reset on failure, disconnect if reset fails.
254             TRY_WITH_RESET(vt->do_io[command](kb, mode, notifynumber, 1, 0));
255             // Re-send the current RGB state as it sometimes gets scrambled
256             TRY_WITH_RESET(vt->updatergb(kb, 1));
257             kb->usbdelay = delay;
258             continue;
259         }
260         case FWUPDATE:
261             // FW update parses a whole word. Unlike hwload/hwsave, there's no try again on failure.
262             if(vt->fwupdate(kb, mode, notifynumber, 0, word)){
263                 free(word);
264                 return 1;
265             }
266             continue;
267         case POLLRATE: {
268             uint rate;
```

```
269             if(sscanf(word, "%u", &rate) == 1 && (rate == 1 || rate == 2 || rate == 4 || rate == 8))
270                 TRY_WITH_RESET(vt->pollrate(kb, mode, notifynumber, rate, 0));
271             continue;
272         }
273         case ERASEPROFILE:
274             // Erase the current profile
275             vt->eraseprofile(kb, mode, notifynumber, 0, 0);
276             // Update profile/mode pointers
277             profile = kb->profile;
278             mode = profile->currentmode;
279             continue;
280         case ERASE: case NAME: case IOFF: case ION: case IAUTO: case
    INOTIFY: case PROFILENAME: case ID: case PROFILEID: case
    DPISEL: case LIFT: case SNAP:
281             // All of the above just parse the whole word
282             vt->do_cmd[command](kb, mode, notifynumber, 0, word);
283             continue;
284         case RGB: {
285             // RGB command has a special response for a single hex constant
286             int r, g, b;
287             if(sscanf(word, "%02x%02x%02x", &r, &g, &b) == 3){
288                 // Set all keys
289                 for(int i = 0; i < N_KEYS_EXTENDED; i++)
290                     vt->rgb(kb, mode, notifynumber, i, word);
291                 continue;
292             }
293             break;
294         }
295         case MACRO:
296             if(!strcmp(word, "clear")){
297                 // Macro has a special clear command
298                 vt->macro(kb, mode, notifynumber, 0, 0);
299                 continue;
300             }
301             break;
302         default:;
303         }
304         // For anything else, split the parameter at the colon
305         int left = -1;
306         sscanf(word, "%*[^:]%n", &left);
307         if(left <= 0)
308             continue;
309         const char* right = word + left;
310         if(right[0] == ':')
311             right++;
312         // Macros and DPI have a separate left-side handler
313         if(command == MACRO || command == DPI){
314             word[left] = 0;
315             vt->do_macro[command](kb, mode, notifynumber, word, right);
316             continue;
317         }
318         // Scan the left side for key names and run the requested command
319         int position = 0, field = 0;
320         char keyname[11];
321         while(position < left && sscanf(word + position, "%10[^:,]%n", keyname, &field) == 1){
322             int keycode;
323             if(!strcmp(keyname, "all")){
324                 // Set all keys
325                 for(int i = 0; i < N_KEYS_EXTENDED; i++)
326                     vt->do_cmd[command](kb, mode, notifynumber, i, right);
327             } else if((sscanf(keyname, "#%d", &keycode) && keycode >= 0 && keycode <
    N_KEYS_EXTENDED)
328                       || (sscanf(keyname, "#x%x", &keycode) && keycode >= 0 && keycode <
    N_KEYS_EXTENDED)){
329                 // Set a key numerically
330                 vt->do_cmd[command](kb, mode, notifynumber, keycode, right);
331             } else {
332                 // Find this key in the keymap
333                 for(unsigned i = 0; i < N_KEYS_EXTENDED; i++){
334                     if(keymap[i].name && !strcmp(keyname, keymap[i].name)){
335                         vt->do_cmd[command](kb, mode, notifynumber, i,
    right);
336                         break;
337                     }
338                 }
339             }
340             if(word[position += field] == ',')
341                 position++;
342         }
343     }
344
345     // Finish up
346     if(!NEEDS_FW_UPDATE(kb)){
347         TRY_WITH_RESET(vt->updatergb(kb, 0));
348         TRY_WITH_RESET(vt->updatedpi(kb, 0));
349     }
350     free(word);
```

```
351     return 0;
352 }
```

Here is the call graph for this function:

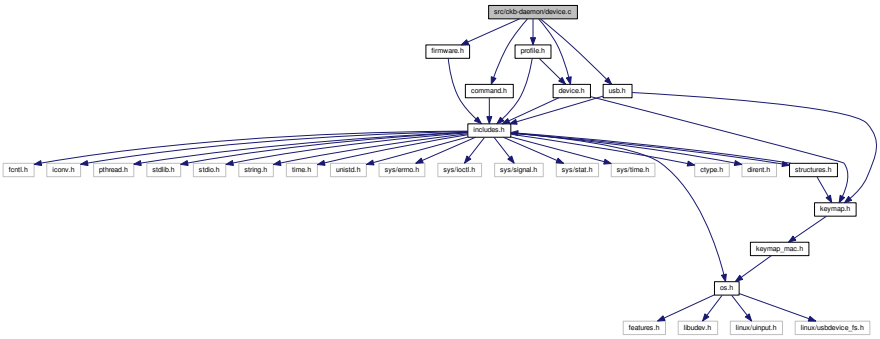


Here is the caller graph for this function:

13.7 src/ckb-daemon/device.c File Reference

```
#include "command.h"
#include "device.h"
#include "firmware.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for device.c:



Functions

- int _start_dev (usbdevice *kb, int makeactive)
- int start_dev (usbdevice *kb, int makeactive)

Variables

- int hwload_mode = 1

  hwload_mode = 1 means read hardware once. should be enough
- usbdevice keyboard [9]

  remember all usb devices. Needed for closeusb().
- pthread_mutex_t devlistmutex = PTHREAD_MUTEX_INITIALIZER
- pthread_mutex_t devmutex [9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

  Mutex for handling the usbdevice structure.
- pthread_mutex_t inputmutex [9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

  Mutex for dealing with usb input frames.
- pthread_mutex_t macromutex [9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

  Protecting macros against lightning: Both use usb_send.
- pthread_mutex_t macromutex2 [9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

  Protecting the single link list of threads and the macrovar.
- pthread_cond_t macrovar [9] = { [0 ... 9 -1] = PTHREAD_COND_INITIALIZER }

  This variable is used to stop and wakeup all macro threads which have to wait.

13.7.1 Function Documentation

13.7.1.1 int _start_dev (usbdevice * *kb*, int *makeactive*)

_start_dev get fw-info and pollrate; if available, install new firmware; get all hardware profiles.

Parameters

kb	the normal kb pointer to the usbdevice. Is also valid for mice.
makeactive	if set to 1, activate the device via setactive()

Returns

0 if success, other else

- This hacker code is tricky in mutliple aspects. What it means is:

  if hwload_mode == 0: just set pollrate to 0 and clear features in the bottom lines of the if-block.

  if hwload_mode == 1: if the device has FEAT_HWLOAD active, call getfwversion(). If it returns true, there was an error while detecting fw-version. Put error message, reset FEAT_HWLOAD and finalize as above.

  if hwload_mode == 2: if the device has FEAT_HWLOAD active, call getfwversion(). If it returns true, there was an error while detecting fw-version. Put error message and return directly from function with error.

  Why do not you just write it down?
- Now check if device needs a firmware update. If so, set it up and leave the function without error.
- Device needs a firmware update. Finish setting up but don't do anything.
- Load profile from device if the hw-pointer is not set yet and hw-loading is possible and allowed.

  return error if mode == 2 (load always) and loading got an error. Else reset HWLOAD feature, because hwload must be 1.

  That is real Horror code.

Definition at line 25 of file device.c.

References usbdevice::active, ckb_info, ckb_warn, DEV_MAX, FEAT_ADJRATE, FEAT_FWUPDATE, FEAT_FWVERSION, FEAT_HWLOAD, FEAT_POLLRATE, FEAT_RGB, usbdevice::features, usbdevice::fwversion, getfwversion(), HAS_FEATURES, usbdevice::hw, hwload_mode, hwloadprofile, NEEDS_FW_UPDATE, usbdevice::pollrate, and setactive.

Referenced by start_dev().

```
25                                              {
26      // Get the firmware version from the device
27      if(kb->pollrate == 0){
35          if(!hwload_mode || (HAS_FEATURES(kb, FEAT_HWLOAD) &&
    getfwversion(kb))){
36              if(hwload_mode == 2)
37                  // hwload=always. Report setup failure.
38                  return -1;
39              else if(hwload_mode){
40                  // hwload=once. Log failure, prevent trying again, and continue.
41                  ckb_warn("Unable to load firmware version/poll rate\n");
42                  kb->features &= ~FEAT_HWLOAD;
43              }
44              kb->pollrate = 0;
45              kb->features &= ~(FEAT_POLLRATE | FEAT_ADJRATE);
46              if(kb->fwversion == 0)
47                  kb->features &= ~(FEAT_FWVERSION |
    FEAT_FWUPDATE);
48          }
49      }
54      if(NEEDS_FW_UPDATE(kb)){
56          ckb_info("Device needs a firmware update. Please issue a fwupdate command.\n");
57          kb->features = FEAT_RGB | FEAT_FWVERSION |
    FEAT_FWUPDATE;
58          kb->active = 1;
59          return 0;
60      }
66      if(!kb->hw && hwload_mode && HAS_FEATURES(kb,
    FEAT_HWLOAD)){
67          if(hwloadprofile(kb, 1)){
68              if(hwload_mode == 2)
69                  return -1;
70              ckb_warn("Unable to load hardware profile\n");
71              kb->features &= ~FEAT_HWLOAD;
72          }
73      }
74      // Activate software mode if requested
75      if(makeactive)
76          return setactive(kb, 1);
77      #ifdef DEBUG
78      // 12 for each device + null terminator
79      char devlist[12*(DEV_MAX-1)+1];
80      int devlistpos = 0;
81      for(unsigned i = 1; i < DEV_MAX; i++){
82          devlistpos += sprintf(&devlist[devlistpos], "%u: 0x%x; ", i, keyboard[i].product);
83      }
84      ckb_info("Attached Devices: %s\n", devlist);
85      #endif
86      return 0;
87  }
```

Here is the call graph for this function:

Here is the caller graph for this function:

13.7.1.2 int start_dev (usbdevice * kb, int makeactive)

Definition at line 89 of file device.c.

References _start_dev(), USB_DELAY_DEFAULT, and usbdevice::usbdelay.

```
89                                          {
90     // Force USB interval to 10ms during initial setup phase; return to nominal 5ms after setup completes.
91     kb->usbdelay = 10;
92     int res = _start_dev(kb, makeactive);
93     kb->usbdelay = USB_DELAY_DEFAULT;
94     return res;
95 }
```

Here is the call graph for this function:

13.7.2 Variable Documentation

13.7.2.1 pthread_mutex_t devlistmutex = PTHREAD_MUTEX_INITIALIZER

Definition at line 11 of file device.c.

13.7.2.2 pthread_mutex_t devmutex[9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

Definition at line 12 of file device.c.

Referenced by _updateconnected(), quitWithLock(), and usb_rm_device().

13.7.2.3 int hwload_mode = 1

hwload_mode is defined in device.c

Definition at line 7 of file device.c.

Referenced by _start_dev(), _usbrecv(), _usbsend(), and main().

13.7.2.4 pthread_mutex_t inputmutex[9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

Definition at line 13 of file device.c.

13.7.2.5 usbdevice keyboard[9]

Definition at line 10 of file device.c.

Referenced by _mkdevpath(), _mknotifynode(), _rmnotifynode(), _setupusb(), _updateconnected(), closeusb(), main(), mkfwnode(), os_closeusb(), os_inputmain(), os_inputopen(), os_setupusb(), quitWithLock(), rmdevpath(), usb_rm_device(), and usbadd().

13.7.2.6 pthread_mutex_t macromutex[9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

Definition at line 14 of file device.c.

13.7.2.7 pthread_mutex_t macromutex2[9] = { [0 ... 9 -1] = PTHREAD_MUTEX_INITIALIZER }

Definition at line 15 of file device.c.

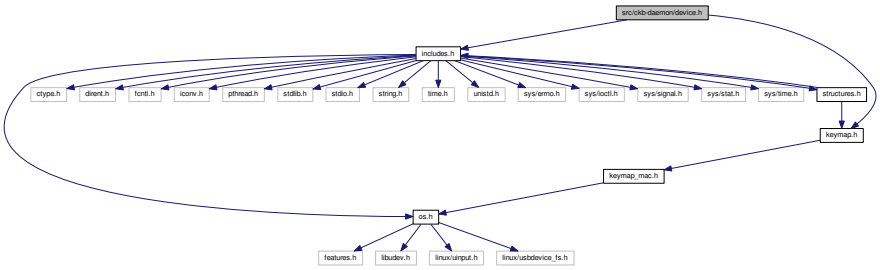
13.7.2.8 pthread_cond_t macrovar[9] = { [0 ... 9 -1] = PTHREAD_COND_INITIALIZER }

Definition at line 16 of file device.c.

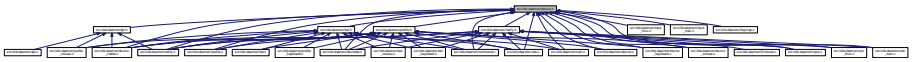
13.8 src/ckb-daemon/device.h File Reference

```
#include "includes.h"
#include "keymap.h"
```

Include dependency graph for device.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define DEV_MAX 9
- #define IS_CONNECTED(kb) ((kb) && (kb)->handle && (kb)->uinput_kb && (kb)->uinput_mouse)
- #define dmutex(kb) (devmutex + INDEX_OF(kb, keyboard))
- #define imutex(kb) (inputmutex + INDEX_OF(kb, keyboard))
- #define mmutex(kb) (macromutex + INDEX_OF(kb, keyboard))
- #define mmutex2(kb) (macromutex2 + INDEX_OF(kb, keyboard))
- #define mvar(kb) (macrovar + INDEX_OF(kb, keyboard))
- #define setactive(kb, makeactive) ((makeactive) ? (kb)->vtable->active((kb), 0, 0, 0, 0) : (kb)->vtable->idle((kb), 0, 0, 0, 0))

  setactive() calls via the corresponding kb->vtable either the active() or the idle() function.
  active() is called if the parameter makeactive is true, idle if it is false.
  What function is called effectively is device dependent. Have a look at device_vtable.c for more information.
- #define IN_HID 0x80
- #define IN_CORSAIR 0x40

- #define ACT_LIGHT 1
- #define ACT_NEXT 3
- #define ACT_NEXT_NOWRAP 5
- #define ACT_LOCK 8
- #define ACT_MR_RING 9
- #define ACT_M1 10
- #define ACT_M2 11
- #define ACT_M3 12

Functions

- int start_dev (usbdevice *kb, int makeactive)
- int start_kb_nrgb (usbdevice *kb, int makeactive)
- int setactive_kb (usbdevice *kb, int active)
- int setactive_mouse (usbdevice *kb, int active)
- int cmd_active_kb (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- int cmd_active_mouse (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- int cmd_idle_kb (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- int cmd_idle_mouse (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- int cmd_pollrate (usbdevice *kb, usbmode *dummy1, int dummy2, int rate, const char *dummy3)
- void setmodeindex_nrgb (usbdevice *kb, int index)

Variables

- usbdevice keyboard [9]

  remember all usb devices. Needed for closeusb().
- pthread_mutex_t devmutex [9]

  Mutex for handling the usbdevice structure.
- pthread_mutex_t inputmutex [9]

  Mutex for dealing with usb input frames.
- pthread_mutex_t macromutex [9]

  Protecting macros against lightning: Both use usb_send.
- pthread_mutex_t macromutex2 [9]

  Protecting the single link list of threads and the macrovar.
- pthread_cond_t macrovar [9]

  This variable is used to stop and wakeup all macro threads which have to wait.

13.8.1 Macro Definition Documentation

13.8.1.1 #define ACT_LIGHT 1

Definition at line 68 of file device.h.

Referenced by setactive_kb().

13.8.1.2 #define ACT_LOCK 8

Definition at line 71 of file device.h.

Referenced by setactive_kb().

13.8.1.3 #define ACT_M1 10

Definition at line 73 of file device.h.

Referenced by setactive_kb().

13.8.1.4 #define ACT_M2 11

Definition at line 74 of file device.h.

Referenced by setactive_kb().

13.8.1.5 #define ACT_M3 12

Definition at line 75 of file device.h.

Referenced by setactive_kb().

13.8.1.6 #define ACT_MR_RING 9

Definition at line 72 of file device.h.

Referenced by setactive_kb().

13.8.1.7 #define ACT_NEXT 3

Definition at line 69 of file device.h.

13.8.1.8 #define ACT_NEXT_NOWRAP 5

Definition at line 70 of file device.h.

13.8.1.9 #define DEV_MAX 9

Definition at line 8 of file device.h.

Referenced by _start_dev(), _updateconnected(), quitWithLock(), usb_rm_device(), and usbadd().

13.8.1.10 #define dmutex(*kb*) (devmutex + INDEX_OF(kb, keyboard))

Definition at line 18 of file device.h.

Referenced by _ledthread(), _setupusb(), closeusb(), devmain(), and usbadd().

13.8.1.11 #define imutex(*kb*) (inputmutex + INDEX_OF(kb, keyboard))

Definition at line 22 of file device.h.

Referenced by _setupusb(), closeusb(), cmd_bind(), cmd_erase(), cmd_eraseprofile(), cmd_get(), cmd_macro(), cmd_notify(), cmd_rebind(), cmd_unbind(), os_inputmain(), setactive_kb(), setactive_mouse(), and setupusb().

13.8.1.12 #define IN_CORSAIR 0x40

Definition at line 65 of file device.h.

Referenced by setactive_kb(), and setactive_mouse().

13.8.1.13 #define IN_HID 0x80

Definition at line 64 of file device.h.

Referenced by setactive_kb(), and setactive_mouse().

13.8.1.14 #define IS_CONNECTED(*kb*) ((kb) && (kb)->handle && (kb)->uinput_kb && (kb)->uinput_mouse)

Definition at line 12 of file device.h.

Referenced by _updateconnected(), devmain(), quitWithLock(), and usbadd().

13.8.1.15 #define mmutex(*kb*) (macromutex + INDEX_OF(kb, keyboard))

Definition at line 26 of file device.h.

Referenced by _usbrecv(), _usbsend(), and play_macro().

13.8.1.16 #define mmutex2(*kb*) (macromutex2 + INDEX_OF(kb, keyboard))

Definition at line 28 of file device.h.

Referenced by play_macro().

13.8.1.17 #define mvar(*kb*) (macrovar + INDEX_OF(kb, keyboard))

Definition at line 30 of file device.h.

Referenced by play_macro().

13.8.1.18 #define setactive(*kb, makeactive*) ((makeactive) ? (kb)->vtable->active((kb), 0, 0, 0, 0) : (kb)->vtable->idle((kb), 0, 0, 0, 0))

Definition at line 44 of file device.h.

Referenced by _start_dev(), and revertusb().

13.8.2 Function Documentation

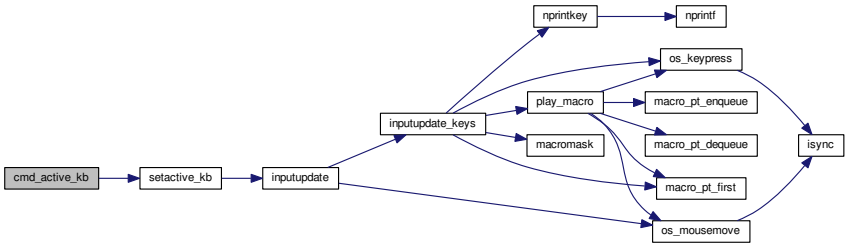
13.8.2.1 int cmd_active_kb (usbdevice * *kb,* usbmode * *dummy1,* int *dummy2,* int *dummy3,* const char * *dummy4*)

Definition at line 114 of file device_keyboard.c.

References setactive_kb().

```
114                                                                                                 {
115     (void)dummy1;
116     (void)dummy2;
117     (void)dummy3;
118     (void)dummy4;
119
120     return setactive_kb(kb, 1);
121 }
```

Here is the call graph for this function:



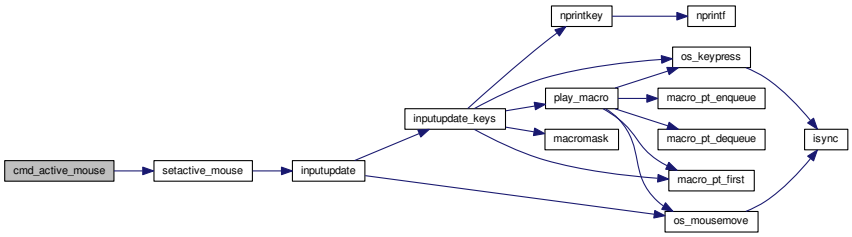
13.8.2.2 int cmd_active_mouse (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 44 of file device_mouse.c.

References setactive_mouse().

```
44                                                                                  {
45     (void)dummy1;
46     (void)dummy2;
47     (void)dummy3;
48     (void)dummy4;
49
50     return setactive_mouse(kb, 1);
51 }
```

Here is the call graph for this function:



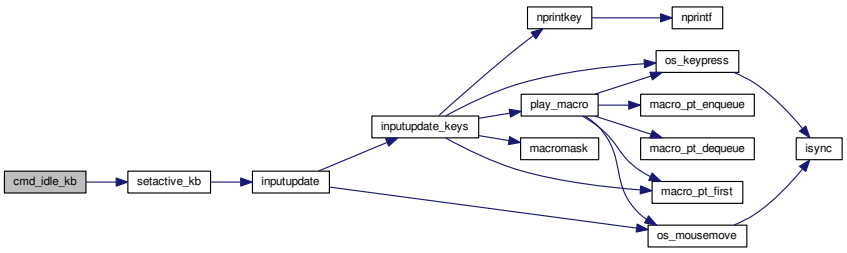
13.8.2.3 int cmd_idle_kb (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 123 of file device_keyboard.c.

References setactive_kb().

```
123                                                                                 {
124     (void)dummy1;
125     (void)dummy2;
126     (void)dummy3;
127     (void)dummy4;
128
129     return setactive_kb(kb, 0);
130 }
```

Here is the call graph for this function:



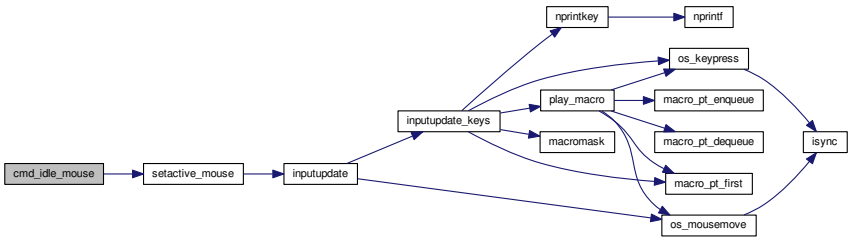
13.8.2.4 int cmd_idle_mouse (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 53 of file device_mouse.c.

References setactive_mouse().

```
53                                                                                                  {
54      (void)dummy1;
55      (void)dummy2;
56      (void)dummy3;
57      (void)dummy4;
58
59      return setactive_mouse(kb, 0);
60  }
```

Here is the call graph for this function:



13.8.2.5 int cmd_pollrate (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *rate*, const char * *dummy3*)

Definition at line 62 of file device_mouse.c.

References MSG_SIZE, usbdevice::pollrate, and usbsend.

```
62                                                                                                  {
63      (void)dummy1;
64      (void)dummy2;
65      (void)dummy3;
66
67      uchar msg[MSG_SIZE] = {
68          0x07, 0x0a, 0, 0, (uchar)rate
69      };
70      if(!usbsend(kb, msg, 1))
71          return -1;
72      // Device should disconnect+reconnect, but update the poll rate field in case it doesn't
73      kb->pollrate = rate;
74      return 0;
75  }
```

13.8.2.6 int setactive_kb (usbdevice * *kb*, int *active*)

Definition at line 20 of file device_keyboard.c.

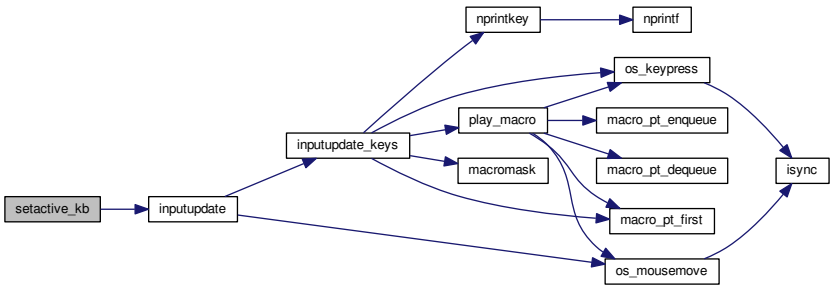
References ACT_LIGHT, ACT_LOCK, ACT_M1, ACT_M2, ACT_M3, ACT_MR_RING, usbdevice::active, DELAY_MEDIUM, lighting::forceupdate, imutex, IN_CORSAIR, IN_HID, usbdevice::input, inputupdate(), keymap, usbinput::keys, usbprofile::lastlight, MSG_SIZE, N_KEYS_HW, NEEDS_FW_UPDATE, usbdevice::profile, usbsend, and usbdevice::vtable.

Referenced by cmd_active_kb(), and cmd_idle_kb().

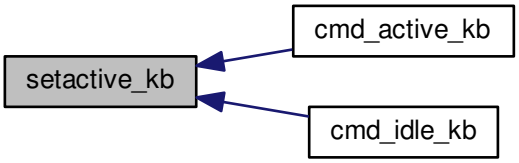
```
20                                          {
21     if(NEEDS_FW_UPDATE(kb))
22         return 0;
23
24     pthread_mutex_lock(imutex(kb));
25     kb->active = !!active;
26     kb->profile->lastlight.forceupdate = 1;
27     // Clear input
28     memset(&kb->input.keys, 0, sizeof(kb->input.keys));
29     inputupdate(kb);
30     pthread_mutex_unlock(imutex(kb));
31
32     uchar msg[3][MSG_SIZE] = {
33         { 0x07, 0x04, 0 },                  // Disables or enables HW control for top row
34         { 0x07, 0x40, 0 },                  // Selects key input
35         { 0x07, 0x05, 2, 0, 0x03, 0x00 }    // Commits key input selection
36     };
37     if(active){
38         // Put the M-keys (K95) as well as the Brightness/Lock keys into software-controlled mode.
39         msg[0][2] = 2;
40         if(!usbsend(kb, msg[0], 1))
41             return -1;
42         DELAY_MEDIUM(kb);
43         // Set input mode on the keys. They must be grouped into packets of 60 bytes (+ 4 bytes header)
44         // Keys are referenced in byte pairs, with the first byte representing the key and the second byte
       representing the mode.
45         for(int key = 0; key < N_KEYS_HW; ){
46             int pair;
47             for(pair = 0; pair < 30 && key < N_KEYS_HW; pair++, key++){
48                 // Select both standard and Corsair input. The standard input will be ignored except in
       BIOS mode.
49                 uchar action = IN_HID | IN_CORSAIR;
50                 // Additionally, make MR activate the MR ring (this is disabled for now, may be back later)
51                 //if(keymap[key].name && !strcmp(keymap[key].name, "mr"))
52                 //    action |= ACT_MR_RING;
53                 msg[1][4 + pair * 2] = key;
54                 msg[1][5 + pair * 2] = action;
55             }
56             // Byte 2 = pair count (usually 30, less on final message)
57             msg[1][2] = pair;
58             if(!usbsend(kb, msg[1], 1))
59                 return -1;
60         }
61         // Commit new input settings
62         if(!usbsend(kb, msg[2], 1))
63             return -1;
64         DELAY_MEDIUM(kb);
65     } else {
66         // Set the M-keys back into hardware mode, restore hardware RGB profile. It has to be sent twice
       for some reason.
67         msg[0][2] = 1;
68         if(!usbsend(kb, msg[0], 1))
69             return -1;
70         DELAY_MEDIUM(kb);
71         if(!usbsend(kb, msg[0], 1))
72             return -1;
73         DELAY_MEDIUM(kb);
74 #ifdef OS_LINUX
75         // On OSX the default key mappings are fine. On Linux, the G keys will freeze the keyboard. Set the
       keyboard entirely to HID input.
76         for(int key = 0; key < N_KEYS_HW; ){
77             int pair;
78             for(pair = 0; pair < 30 && key < N_KEYS_HW; pair++, key++){
79                 uchar action = IN_HID;
80                 // Enable hardware actions
81                 if(keymap[key].name){
82                     if(!strcmp(keymap[key].name, "mr"))
83                         action = ACT_MR_RING;
84                     else if(!strcmp(keymap[key].name, "m1"))
85                         action = ACT_M1;
86                     else if(!strcmp(keymap[key].name, "m2"))
87                         action = ACT_M2;
88                     else if(!strcmp(keymap[key].name, "m3"))
```

```
89                      action = ACT_M3;
90                  else if(!strcmp(keymap[key].name, "light"))
91                      action = ACT_LIGHT;
92                  else if(!strcmp(keymap[key].name, "lock"))
93                      action = ACT_LOCK;
94              }
95              msg[1][4 + pair * 2] = key;
96              msg[1][5 + pair * 2] = action;
97          }
98          // Byte 2 = pair count (usually 30, less on final message)
99          msg[1][2] = pair;
100          if(!usbsend(kb, msg[1], 1))
101              return -1;
102      }
103      // Commit new input settings
104      if(!usbsend(kb, msg[2], 1))
105          return -1;
106      DELAY_MEDIUM(kb);
107 #endif
108     }
109     // Update indicator LEDs if the profile contains settings for them
110     kb->vtable->updateindicators(kb, 0);
111     return 0;
112 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.8.2.7 int setactive_mouse (usbdevice * *kb*, int *active*)

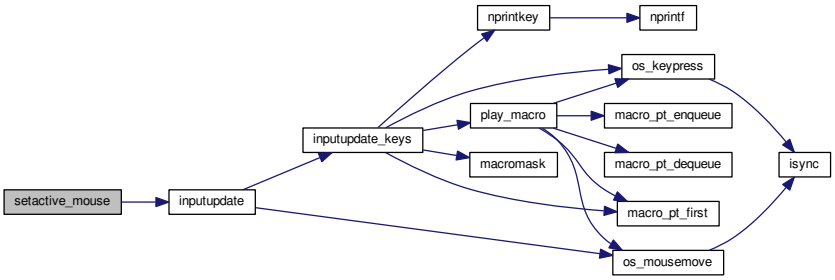
Definition at line 9 of file device_mouse.c.

References usbdevice::active, lighting::forceupdate, imutex, IN_CORSAIR, IN_HID, usbdevice::input, inputupdate(), usbinput::keys, usbprofile::lastlight, MSG_SIZE, NEEDS_FW_UPDATE, usbdevice::profile, and usbsend.

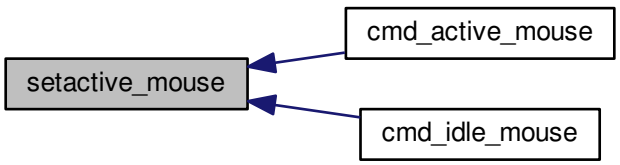
Referenced by cmd_active_mouse(), and cmd_idle_mouse().

```
9                                                   {
10      if(NEEDS_FW_UPDATE(kb))
11          return 0;
12      const int keycount = 20;
13      uchar msg[2][MSG_SIZE] = {
14          { 0x07, 0x04, 0 },                  // Disables or enables HW control for DPI and Sniper button
15          { 0x07, 0x40, keycount, 0 },        // Select button input (simlilar to the packet sent to
        keyboards, but lacks a commit packet)
16      };
17      if(active)
18          // Put the mouse into SW mode
19          msg[0][2] = 2;
20      else
21          // Restore HW mode
22          msg[0][2] = 1;
23      pthread_mutex_lock(imutex(kb));
24      kb->active = !!active;
25      kb->profile->lastlight.forceupdate = 1;
26      // Clear input
27      memset(&kb->input.keys, 0, sizeof(kb->input.keys));
28      inputupdate(kb);
29      pthread_mutex_unlock(imutex(kb));
30      if(!usbsend(kb, msg[0], 1))
31          return -1;
32      if(active){
33          // Set up key input
34          if(!usbsend(kb, msg[1], 1))
35              return -1;
36          for(int i = 0; i < keycount; i++){
37              msg[1][i * 2 + 4] = i + 1;
38              msg[1][i * 2 + 5] = (i < 6 ? IN_HID : IN_CORSAIR);
39          }
40      }
41      return 0;
42  }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.8.2.8 void setmodeindex_nrgb (usbdevice * kb, int index)

Definition at line 132 of file device_keyboard.c.

References NK95_M1, NK95_M2, NK95_M3, and nk95cmd.

```
132                                                 {
133     switch(index % 3){
134     case 0:
135         nk95cmd(kb, NK95_M1);
136         break;
137     case 1:
138         nk95cmd(kb, NK95_M2);
139         break;
140     case 2:
141         nk95cmd(kb, NK95_M3);
142         break;
143     }
144 }
```

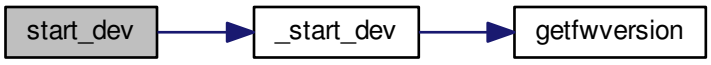
13.8.2.9 int start_dev (usbdevice * kb, int makeactive)

Definition at line 89 of file device.c.

References _start_dev(), USB_DELAY_DEFAULT, and usbdevice::usbdelay.

```
89                                              {
90     // Force USB interval to 10ms during initial setup phase; return to nominal 5ms after setup completes.
91     kb->usbdelay = 10;
92     int res = _start_dev(kb, makeactive);
93     kb->usbdelay = USB_DELAY_DEFAULT;
94     return res;
95 }
```

Here is the call graph for this function:



13.8.2.10 int start_kb_nrgb (usbdevice * kb, int makeactive)

Definition at line 9 of file device_keyboard.c.

References usbdevice::active, NK95_HWOFF, nk95cmd, and usbdevice::pollrate.

```
9                                                   {
10     (void)makeactive;
11
12     // Put the non-RGB K95 into software mode. Nothing else needs to be done hardware wise
13     nk95cmd(kb, NK95_HWOFF);
14     // Fill out RGB features for consistency, even though the keyboard doesn't have them
15     kb->active = 1;
16     kb->pollrate = -1;
17     return 0;
18 }
```

13.8.3 Variable Documentation

13.8.3.1 pthread_mutex_t devmutex[9]

Definition at line 12 of file device.c.

Referenced by _updateconnected(), quitWithLock(), and usb_rm_device().

13.8.3.2 pthread_mutex_t inputmutex[9]

Definition at line 13 of file device.c.

13.8.3.3 usbdevice keyboard[9]

Definition at line 10 of file device.c.

Referenced by _mkdevpath(), _mknotifynode(), _rmnotifynode(), _setupusb(), _updateconnected(), closeusb(), main(), mkfwnode(), os_closeusb(), os_inputmain(), os_inputopen(), os_setupusb(), quitWithLock(), rmdevpath(), usb_rm_device(), and usbadd().

13.8.3.4 pthread_mutex_t macromutex[9]

Definition at line 14 of file device.c.

13.8.3.5 pthread_mutex_t macromutex2[9]

Definition at line 15 of file device.c.

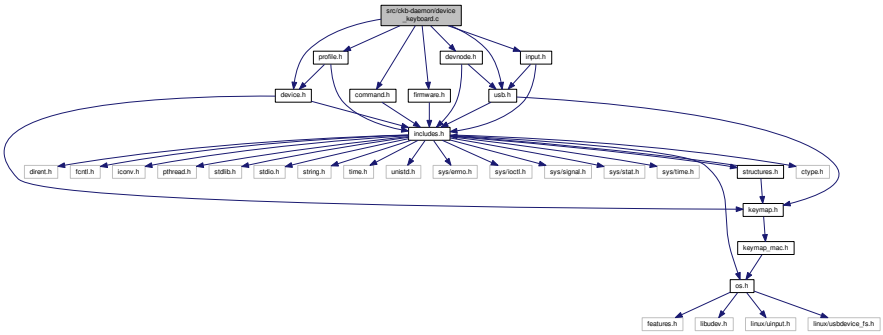
13.8.3.6 pthread_cond_t macrovar[9]

Definition at line 16 of file device.c.

13.9 src/ckb-daemon/device_keyboard.c File Reference

```
#include "command.h"
#include "device.h"
#include "devnode.h"
#include "firmware.h"
#include "input.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for device_keyboard.c:



Functions

- int start_kb_nrgb (usbdevice ∗kb, int makeactive)
- int setactive_kb (usbdevice ∗kb, int active)
- int cmd_active_kb (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)
- int cmd_idle_kb (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)
- void setmodeindex_nrgb (usbdevice ∗kb, int index)

13.9.1 Function Documentation

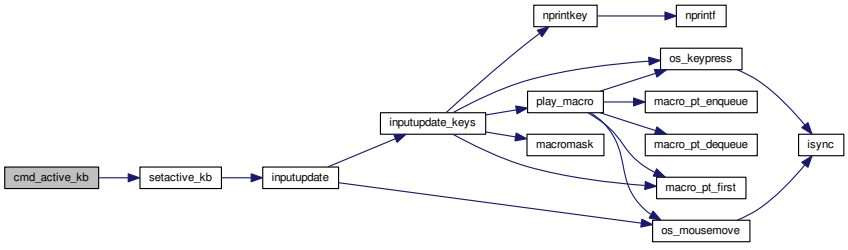
13.9.1.1 int cmd_active_kb (usbdevice ∗ *kb,* usbmode ∗ *dummy1,* int *dummy2,* int *dummy3,* const char ∗ *dummy4*)

Definition at line 114 of file device_keyboard.c.

References setactive_kb().

```
114                                                                                                 {
115     (void)dummy1;
116     (void)dummy2;
117     (void)dummy3;
118     (void)dummy4;
119
120     return setactive_kb(kb, 1);
121 }
```

Here is the call graph for this function:



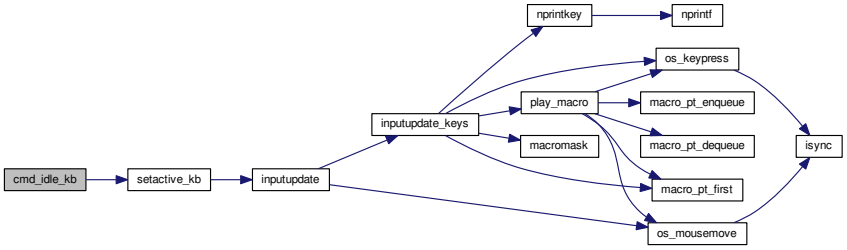
13.9.1.2 int cmd_idle_kb (usbdevice * kb, usbmode * dummy1, int dummy2, int dummy3, const char * dummy4)

Definition at line 123 of file device_keyboard.c.

References setactive_kb().

```
123                                                                                                 {
124     (void)dummy1;
125     (void)dummy2;
126     (void)dummy3;
127     (void)dummy4;
128
129     return setactive_kb(kb, 0);
130 }
```

Here is the call graph for this function:



13.9.1.3 int setactive_kb (usbdevice * kb, int active)

Definition at line 20 of file device_keyboard.c.

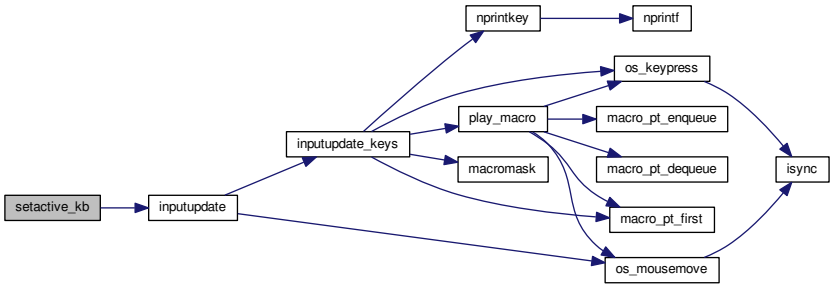
References ACT_LIGHT, ACT_LOCK, ACT_M1, ACT_M2, ACT_M3, ACT_MR_RING, usbdevice::active, DELAY_-MEDIUM, lighting::forceupdate, imutex, IN_CORSAIR, IN_HID, usbdevice::input, inputupdate(), keymap, usbinput-::keys, usbprofile::lastlight, MSG_SIZE, N_KEYS_HW, NEEDS_FW_UPDATE, usbdevice::profile, usbsend, and usbdevice::vtable.

Referenced by cmd_active_kb(), and cmd_idle_kb().

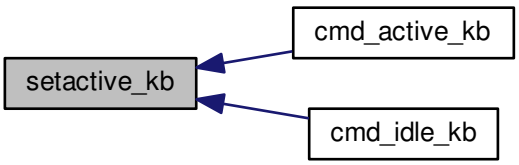
```
20                                          {
21     if(NEEDS_FW_UPDATE(kb))
22         return 0;
23
24     pthread_mutex_lock(imutex(kb));
25     kb->active = !!active;
26     kb->profile->lastlight.forceupdate = 1;
27     // Clear input
28     memset(&kb->input.keys, 0, sizeof(kb->input.keys));
29     inputupdate(kb);
30     pthread_mutex_unlock(imutex(kb));
31
32     uchar msg[3][MSG_SIZE] = {
33         { 0x07, 0x04, 0 },                  // Disables or enables HW control for top row
34         { 0x07, 0x40, 0 },                  // Selects key input
35         { 0x07, 0x05, 2, 0, 0x03, 0x00 }    // Commits key input selection
36     };
37     if(active){
38         // Put the M-keys (K95) as well as the Brightness/Lock keys into software-controlled mode.
39         msg[0][2] = 2;
40         if(!usbsend(kb, msg[0], 1))
41             return -1;
42         DELAY_MEDIUM(kb);
43         // Set input mode on the keys. They must be grouped into packets of 60 bytes (+ 4 bytes header)
44         // Keys are referenced in byte pairs, with the first byte representing the key and the second byte
       representing the mode.
45         for(int key = 0; key < N_KEYS_HW; ){
46             int pair;
47             for(pair = 0; pair < 30 && key < N_KEYS_HW; pair++, key++){
48                 // Select both standard and Corsair input. The standard input will be ignored except in
```

```
    BIOS mode.
49                  uchar action = IN_HID | IN_CORSAIR;
50                  // Additionally, make MR activate the MR ring (this is disabled for now, may be back later)
51                  //if(keymap[key].name && !strcmp(keymap[key].name, "mr"))
52                  //    action |= ACT_MR_RING;
53                  msg[1][4 + pair * 2] = key;
54                  msg[1][5 + pair * 2] = action;
55              }
56              // Byte 2 = pair count (usually 30, less on final message)
57              msg[1][2] = pair;
58              if(!usbsend(kb, msg[1], 1))
59                  return -1;
60          }
61          // Commit new input settings
62          if(!usbsend(kb, msg[2], 1))
63              return -1;
64          DELAY_MEDIUM(kb);
65      } else {
66          // Set the M-keys back into hardware mode, restore hardware RGB profile. It has to be sent twice
    for some reason.
67          msg[0][2] = 1;
68          if(!usbsend(kb, msg[0], 1))
69              return -1;
70          DELAY_MEDIUM(kb);
71          if(!usbsend(kb, msg[0], 1))
72              return -1;
73          DELAY_MEDIUM(kb);
74 #ifdef OS_LINUX
75          // On OSX the default key mappings are fine. On Linux, the G keys will freeze the keyboard. Set the
    keyboard entirely to HID input.
76          for(int key = 0; key < N_KEYS_HW; ){
77              int pair;
78              for(pair = 0; pair < 30 && key < N_KEYS_HW; pair++, key++){
79                  uchar action = IN_HID;
80                  // Enable hardware actions
81                  if(keymap[key].name){
82                      if(!strcmp(keymap[key].name, "mr"))
83                          action = ACT_MR_RING;
84                      else if(!strcmp(keymap[key].name, "m1"))
85                          action = ACT_M1;
86                      else if(!strcmp(keymap[key].name, "m2"))
87                          action = ACT_M2;
88                      else if(!strcmp(keymap[key].name, "m3"))
89                          action = ACT_M3;
90                      else if(!strcmp(keymap[key].name, "light"))
91                          action = ACT_LIGHT;
92                      else if(!strcmp(keymap[key].name, "lock"))
93                          action = ACT_LOCK;
94                  }
95                  msg[1][4 + pair * 2] = key;
96                  msg[1][5 + pair * 2] = action;
97              }
98              // Byte 2 = pair count (usually 30, less on final message)
99              msg[1][2] = pair;
100              if(!usbsend(kb, msg[1], 1))
101                  return -1;
102          }
103          // Commit new input settings
104          if(!usbsend(kb, msg[2], 1))
105              return -1;
106          DELAY_MEDIUM(kb);
107 #endif
108      }
109      // Update indicator LEDs if the profile contains settings for them
110      kb->vtable->updateindicators(kb, 0);
111      return 0;
112 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.9.1.4 void setmodeindex_nrgb (usbdevice * *kb*, int *index*)

Definition at line 132 of file device_keyboard.c.

References NK95_M1, NK95_M2, NK95_M3, and nk95cmd.

```
132                                                 {
133     switch(index % 3){
134     case 0:
135         nk95cmd(kb, NK95_M1);
136         break;
137     case 1:
138         nk95cmd(kb, NK95_M2);
139         break;
140     case 2:
141         nk95cmd(kb, NK95_M3);
142         break;
143     }
144 }
```

13.9.1.5 int start_kb_nrgb (usbdevice * *kb*, int *makeactive*)

Definition at line 9 of file device_keyboard.c.

References usbdevice::active, NK95_HWOFF, nk95cmd, and usbdevice::pollrate.

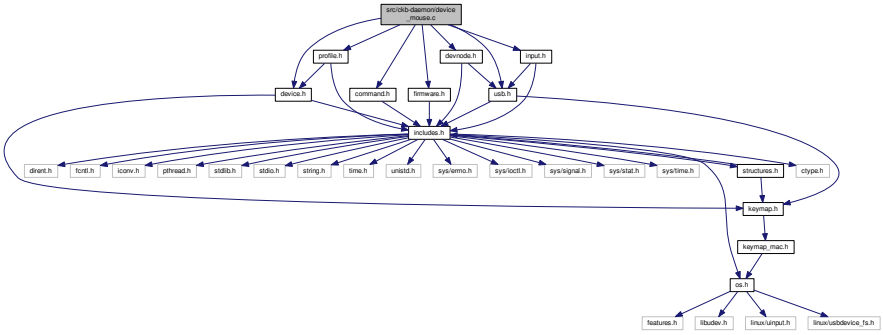
```
9                                                   {
10     (void)makeactive;
11
```

```
12     // Put the non-RGB K95 into software mode. Nothing else needs to be done hardware wise
13     nk95cmd(kb, NK95_HWOFF);
14     // Fill out RGB features for consistency, even though the keyboard doesn't have them
15     kb->active = 1;
16     kb->pollrate = -1;
17     return 0;
18 }
```

13.10 src/ckb-daemon/device_mouse.c File Reference

```
#include "command.h"
#include "device.h"
#include "devnode.h"
#include "firmware.h"
#include "input.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for device_mouse.c:



Functions

- int setactive_mouse (usbdevice ∗kb, int active)
- int cmd_active_mouse (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)
- int cmd_idle_mouse (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)
- int cmd_pollrate (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int rate, const char ∗dummy3)

13.10.1 Function Documentation

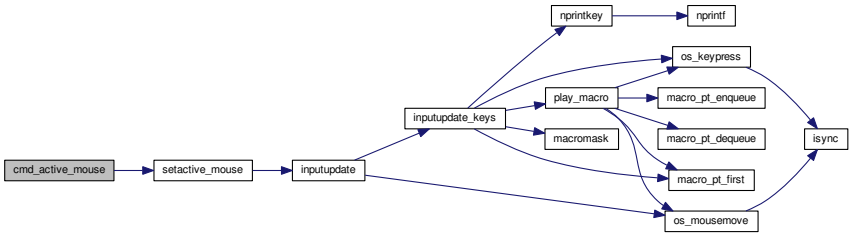
13.10.1.1 int cmd_active_mouse (usbdevice ∗ *kb,* usbmode ∗ *dummy1,* int *dummy2,* int *dummy3,* const char ∗ *dummy4*)

Definition at line 44 of file device_mouse.c.

References setactive_mouse().

```
44                                                                                                  {
45     (void)dummy1;
46     (void)dummy2;
47     (void)dummy3;
48     (void)dummy4;
49
50     return setactive_mouse(kb, 1);
51 }
```

Here is the call graph for this function:



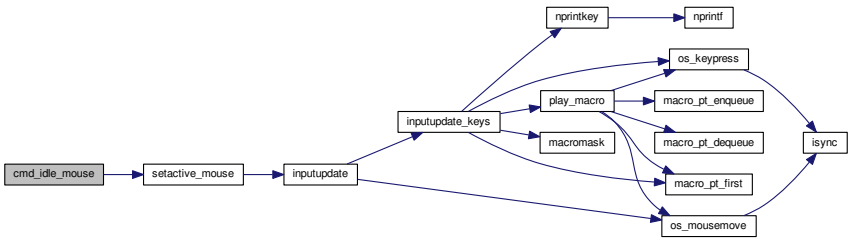
13.10.1.2 int cmd_idle_mouse (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 53 of file device_mouse.c.

References setactive_mouse().

```
53                                                                                                  {
54     (void)dummy1;
55     (void)dummy2;
56     (void)dummy3;
57     (void)dummy4;
58
59     return setactive_mouse(kb, 0);
60 }
```

Here is the call graph for this function:



13.10.1.3 int cmd_pollrate (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *rate*, const char * *dummy3*)

Definition at line 62 of file device_mouse.c.

References MSG_SIZE, usbdevice::pollrate, and usbsend.

```
62                                                                                                  {
63     (void)dummy1;
64     (void)dummy2;
65     (void)dummy3;
66
67     uchar msg[MSG_SIZE] = {
68         0x07, 0x0a, 0, 0, (uchar)rate
69     };
70     if(!usbsend(kb, msg, 1))
71         return -1;
72     // Device should disconnect+reconnect, but update the poll rate field in case it doesn't
73     kb->pollrate = rate;
74     return 0;
75 }
```

13.10.1.4 int setactive_mouse (usbdevice * *kb*, int *active*)

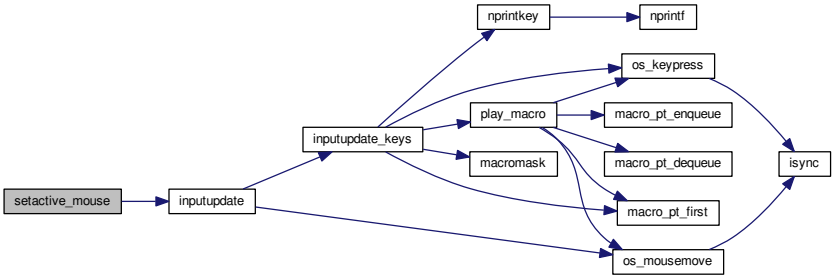
Definition at line 9 of file device_mouse.c.

References usbdevice::active, lighting::forceupdate, imutex, IN_CORSAIR, IN_HID, usbdevice::input, inputupdate(), usbinput::keys, usbprofile::lastlight, MSG_SIZE, NEEDS_FW_UPDATE, usbdevice::profile, and usbsend.

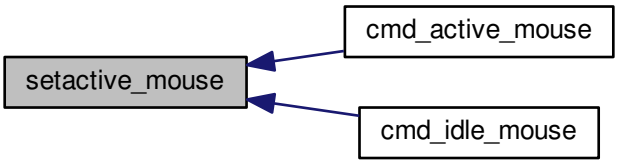
Referenced by cmd_active_mouse(), and cmd_idle_mouse().

```
9                                                {
10      if(NEEDS_FW_UPDATE(kb))
11          return 0;
12      const int keycount = 20;
13      uchar msg[2][MSG_SIZE] = {
14          { 0x07, 0x04, 0 },                  // Disables or enables HW control for DPI and Sniper button
15          { 0x07, 0x40, keycount, 0 },        // Select button input (simlilar to the packet sent to
        keyboards, but lacks a commit packet)
16      };
17      if(active)
18          // Put the mouse into SW mode
19          msg[0][2] = 2;
20      else
21          // Restore HW mode
22          msg[0][2] = 1;
23      pthread_mutex_lock(imutex(kb));
24      kb->active = !!active;
25      kb->profile->lastlight.forceupdate = 1;
26      // Clear input
27      memset(&kb->input.keys, 0, sizeof(kb->input.keys));
28      inputupdate(kb);
29      pthread_mutex_unlock(imutex(kb));
30      if(!usbsend(kb, msg[0], 1))
31          return -1;
32      if(active){
33          // Set up key input
34          if(!usbsend(kb, msg[1], 1))
35              return -1;
36          for(int i = 0; i < keycount; i++){
37              msg[1][i * 2 + 4] = i + 1;
38              msg[1][i * 2 + 5] = (i < 6 ? IN_HID : IN_CORSAIR);
39          }
40      }
41      return 0;
42  }
```

Here is the call graph for this function:



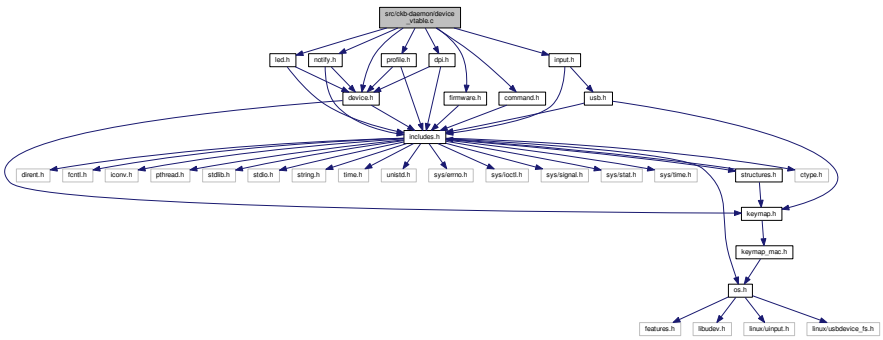
Here is the caller graph for this function:



13.11 src/ckb-daemon/device_vtable.c File Reference

```
#include "command.h"
#include "device.h"
#include "dpi.h"
#include "firmware.h"
#include "input.h"
#include "led.h"
#include "notify.h"
#include "profile.h"
```

Include dependency graph for device_vtable.c:



Functions

- static void cmd_none (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)
- static int cmd_io_none (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)
- static void cmd_macro_none (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, const char ∗dummy3, const char ∗dummy4)
- static int loadprofile_none (usbdevice ∗kb)
- static void int1_void_none (usbdevice ∗kb, int dummy)
- static int int1_int_none (usbdevice ∗kb, int dummy)

Variables

- const devcmd vtable_keyboard

RGB keyboard vtable holds functions for each device type.

- const devcmd vtable_keyboard_nonrgb
- const devcmd vtable_mouse

13.11.1 Function Documentation

13.11.1.1 static int cmd_io_none (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*) [static]

Definition at line 18 of file device_vtable.c.

```
18                                                                                                  {
19     (void)kb;
20     (void)dummy1;
21     (void)dummy2;
22     (void)dummy3;
23     (void)dummy4;
24
25     return 0;
26 }
```

13.11.1.2 static void cmd_macro_none (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, const char * *dummy3*, const char * *dummy4*) [static]

Definition at line 27 of file device_vtable.c.

```
27
        {
28     (void)kb;
29     (void)dummy1;
30     (void)dummy2;
31     (void)dummy3;
32     (void)dummy4;
33 }
```

13.11.1.3 static void cmd_none (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*) [static]

Definition at line 11 of file device_vtable.c.

```
11                                                                                                  {
12     (void)kb;
13     (void)dummy1;
14     (void)dummy2;
15     (void)dummy3;
16     (void)dummy4;
17 }
```

13.11.1.4 static int int1_int_none (usbdevice * *kb*, int *dummy*) [static]

Definition at line 43 of file device_vtable.c.

```
43                                                  {
44     (void)kb;
45     (void)dummy;
46
47     return 0;
48 }
```

13.11.1.5 static void int1_void_none (usbdevice * kb, int dummy) [static]

Definition at line 39 of file device_vtable.c.

```
39                                                  {
40     (void)kb;
41     (void)dummy;
42 }
```

13.11.1.6 static int loadprofile_none (usbdevice * kb) [static]

Definition at line 34 of file device_vtable.c.

```
34                                          {
35     (void)kb;
36
37     return 0;
38 }
```

13.11.2 Variable Documentation

13.11.2.1 const devcmd vtable_keyboard

Definition at line 52 of file device_vtable.c.

Referenced by get_vtable().

13.11.2.2 const devcmd vtable_keyboard_nonrgb

Definition at line 99 of file device_vtable.c.

Referenced by get_vtable().

13.11.2.3 const devcmd vtable_mouse

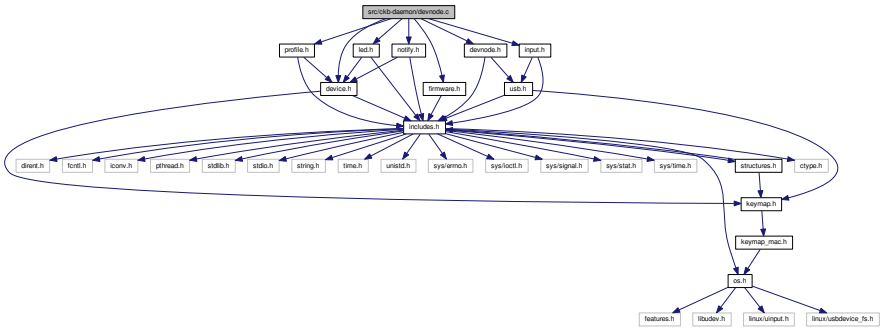
Definition at line 146 of file device_vtable.c.

Referenced by get_vtable().

13.12 src/ckb-daemon/devnode.c File Reference

```
#include "device.h"
#include "devnode.h"
#include "firmware.h"
#include "input.h"
#include "led.h"
#include "notify.h"
#include "profile.h"
```

Include dependency graph for devnode.c:



Data Structures

- struct _readlines_ctx

Macros

- #define S_GID_READ (gid >= 0 ? S_CUSTOM_R : S_READ)
- #define MAX_BUFFER (1024 * 1024 - 1)

Functions

- int rm_recursive (const char *path)
- void _updateconnected ()

  _updateconnected Update the list of connected devices.
- void updateconnected ()

  Update the list of connected devices.
- int _mknotifynode (usbdevice *kb, int notify)
- int mknotifynode (usbdevice *kb, int notify)

  Creates a notification node for the specified keyboard.
- int _rmnotifynode (usbdevice *kb, int notify)
- int rmnotifynode (usbdevice *kb, int notify)

  Removes a notification node for the specified keyboard.
- static int _mkdevpath (usbdevice *kb)
- int mkdevpath (usbdevice *kb)

  Create a dev path for the keyboard at index. Returns 0 on success.
- int rmdevpath (usbdevice *kb)

  Remove the dev path for the keyboard at index. Returns 0 on success.
- int mkfwnode (usbdevice *kb)

  Writes a keyboard's firmware version and poll rate to its device node.
- void readlines_ctx_init (readlines_ctx *ctx)
- void readlines_ctx_free (readlines_ctx ctx)
- unsigned readlines (int fd, readlines_ctx ctx, const char **input)

Variables

- const char *const devpath = "/dev/input/ckb"
- long gid = -1

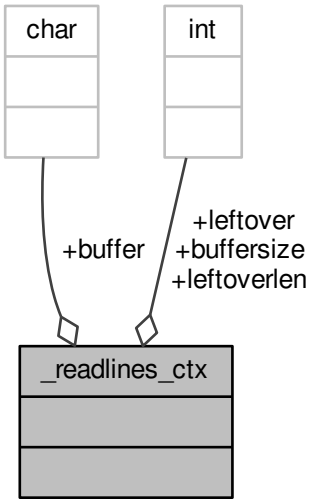
  Group ID for the control nodes. -1 to give read/write access to everybody.

13.12.1 Data Structure Documentation

13.12.1.1 struct _readlines_ctx

Definition at line 335 of file devnode.c.

Collaboration diagram for _readlines_ctx:



Data Fields

char *	buffer	
int	buffersize	
int	leftover	
int	leftoverlen	

13.12.2 Macro Definition Documentation

13.12.2.1 #define MAX_BUFFER (1024 * 1024 - 1)

Definition at line 334 of file devnode.c.

Referenced by readlines().

13.12.2.2 #define S_GID_READ (gid >= 0 ? S_CUSTOM_R : S_READ)

Definition at line 17 of file devnode.c.

Referenced by _mkdevpath(), _mknotifynode(), _updateconnected(), and mkfwnode().

13.12.3 Function Documentation

13.12.3.1 static int _mkdevpath (usbdevice * kb) [static]

Definition at line 136 of file devnode.c.

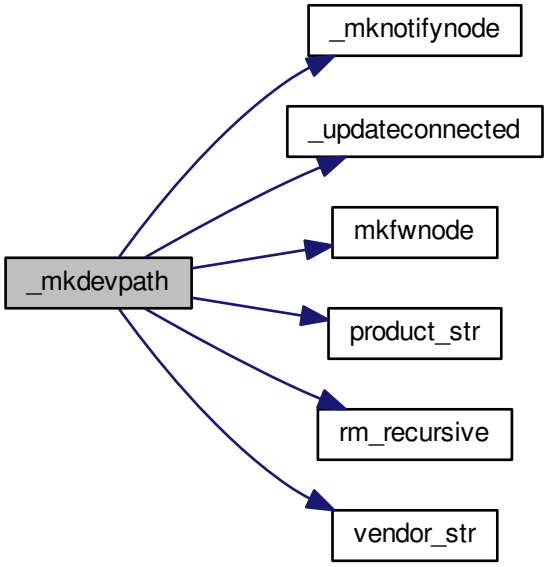
References _mknotifynode(), _updateconnected(), ckb_err, ckb_warn, devpath, FEAT_ADJRATE, FEAT_BIND, FEAT_FWUPDATE, FEAT_FWVERSION, FEAT_MONOCHROME, FEAT_NOTIFY, FEAT_POLLRATE, FEAT_RGB, gid, HAS_FEATURES, INDEX_OF, usbdevice::infifo, keyboard, mkfwnode(), usbdevice::name, usbdevice::product, product_str(), rm_recursive(), S_CUSTOM, S_GID_READ, S_READ, S_READDIR, S_READWRITE, usbdevice::serial, usbdevice::vendor, and vendor_str().

Referenced by mkdevpath().

```
136                                     {
137     int index = INDEX_OF(kb, keyboard);
138     // Create the control path
139     char path[strlen(devpath) + 2];
140     snprintf(path, sizeof(path), "%s%d", devpath, index);
141     if(rm_recursive(path) != 0 && errno != ENOENT){
142         ckb_err("Unable to delete %s: %s\n", path, strerror(errno));
143         return -1;
144     }
145     if(mkdir(path, S_READDIR) != 0){
146         ckb_err("Unable to create %s: %s\n", path, strerror(errno));
147         rm_recursive(path);
148         return -1;
149     }
150     if(gid >= 0)
151         chown(path, 0, gid);
152 
153     if(kb == keyboard + 0){
154         // Root keyboard: write a list of devices
155         _updateconnected();
156         // Write version number
157         char vpath[sizeof(path) + 8];
158         snprintf(vpath, sizeof(vpath), "%s/version", path);
159         FILE* vfile = fopen(vpath, "w");
160         if(vfile){
161             fprintf(vfile, "%s\n", CKB_VERSION_STR);
162             fclose(vfile);
163             chmod(vpath, S_GID_READ);
164             if(gid >= 0)
165                 chown(vpath, 0, gid);
166         } else {
167             ckb_warn("Unable to create %s: %s\n", vpath, strerror(errno));
168             remove(vpath);
169         }
170         // Write PID
171         char ppath[sizeof(path) + 4];
172         snprintf(ppath, sizeof(ppath), "%s/pid", path);
173         FILE* pfile = fopen(ppath, "w");
174         if(pfile){
175             fprintf(pfile, "%u\n", getpid());
176             fclose(pfile);
177             chmod(ppath, S_READ);
178             if(gid >= 0)
179                 chown(vpath, 0, gid);
180         } else {
181             ckb_warn("Unable to create %s: %s\n", ppath, strerror(errno));
182             remove(ppath);
183         }
184     } else {
185         // Create command FIFO
186         char inpath[sizeof(path) + 4];
187         snprintf(inpath, sizeof(inpath), "%s/cmd", path);
188         if(mkfifo(inpath, gid >= 0 ? S_CUSTOM : S_READWRITE) != 0
189                 // Open the node in RDWR mode because RDONLY will lock the thread
190                 || (kb->infifo = open(inpath, O_RDWR) + 1) == 0){
191             // Add one to the FD because 0 is a valid descriptor, but ckb uses 0 for uninitialized devices
192             ckb_err("Unable to create %s: %s\n", inpath, strerror(errno));
193             rm_recursive(path);
194             kb->infifo = 0;
195             return -1;
196         }
197         if(gid >= 0)
198             fchown(kb->infifo - 1, 0, gid);
199 
200         // Create notification FIFO
201         _mknotifynode(kb, 0);
202 
203         // Write the model and serial to files
204         char mpath[sizeof(path) + 6], spath[sizeof(path) + 7];
205         snprintf(mpath, sizeof(mpath), "%s/model", path);
206         snprintf(spath, sizeof(spath), "%s/serial", path);
```

```
207         FILE* mfile = fopen(mpath, "w");
208         if(mfile){
209             fputs(kb->name, mfile);
210             fputc('\n', mfile);
211             fclose(mfile);
212             chmod(mpath, S_GID_READ);
213             if(gid >= 0)
214                 chown(mpath, 0, gid);
215         } else {
216             ckb_warn("Unable to create %s: %s\n", mpath, strerror(errno));
217             remove(mpath);
218         }
219         FILE* sfile = fopen(spath, "w");
220         if(sfile){
221             fputs(kb->serial, sfile);
222             fputc('\n', sfile);
223             fclose(sfile);
224             chmod(spath, S_GID_READ);
225             if(gid >= 0)
226                 chown(spath, 0, gid);
227         } else {
228             ckb_warn("Unable to create %s: %s\n", spath, strerror(errno));
229             remove(spath);
230         }
231         // Write the keyboard's features
232         char fpath[sizeof(path) + 9];
233         snprintf(fpath, sizeof(fpath), "%s/features", path);
234         FILE* ffile = fopen(fpath, "w");
235         if(ffile){
236             fprintf(ffile, "%s %s", vendor_str(kb->vendor),
    product_str(kb->product));
237             if(HAS_FEATURES(kb, FEAT_MONOCHROME))
238                 fputs(" monochrome", ffile);
239             if(HAS_FEATURES(kb, FEAT_RGB))
240                 fputs(" rgb", ffile);
241             if(HAS_FEATURES(kb, FEAT_POLLRATE))
242                 fputs(" pollrate", ffile);
243             if(HAS_FEATURES(kb, FEAT_ADJRATE))
244                 fputs(" adjrate", ffile);
245             if(HAS_FEATURES(kb, FEAT_BIND))
246                 fputs(" bind", ffile);
247             if(HAS_FEATURES(kb, FEAT_NOTIFY))
248                 fputs(" notify", ffile);
249             if(HAS_FEATURES(kb, FEAT_FWVERSION))
250                 fputs(" fwversion", ffile);
251             if(HAS_FEATURES(kb, FEAT_FWUPDATE))
252                 fputs(" fwupdate", ffile);
253             fputc('\n', ffile);
254             fclose(ffile);
255             chmod(fpath, S_GID_READ);
256             if(gid >= 0)
257                 chown(fpath, 0, gid);
258         } else {
259             ckb_warn("Unable to create %s: %s\n", fpath, strerror(errno));
260             remove(fpath);
261         }
262         // Write firmware version and poll rate
263         mkfwnode(kb);
264     }
265     return 0;
266 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.12.3.2 int _mknotifynode (usbdevice * *kb*, int *notify*)

Definition at line 87 of file devnode.c.

References ckb_warn, devpath, gid, INDEX_OF, keyboard, usbdevice::outfifo, OUTFIFO_MAX, and S_GID_READ.

Referenced by _mkdevpath(), and mknotifynode().

```
87                                          {
88     if(notify < 0 || notify >= OUTFIFO_MAX)
89         return -1;
90     if(kb->outfifo[notify] != 0)
91         return 0;
92     // Create the notification node
93     int index = INDEX_OF(kb, keyboard);
94     char outpath[strlen(devpath) + 10];
95     snprintf(outpath, sizeof(outpath), "%s%d/notify%d", devpath, index, notify);
96     if(mkfifo(outpath, S_GID_READ) != 0 || (kb->outfifo[notify] = open(outpath, O_RDWR |
       O_NONBLOCK) + 1) == 0){
97         // Add one to the FD because 0 is a valid descriptor, but ckb uses 0 for uninitialized devices
98         ckb_warn("Unable to create %s: %s\n", outpath, strerror(errno));
99         kb->outfifo[notify] = 0;
100         remove(outpath);
101         return -1;
102     }
103     if(gid >= 0)
104         fchown(kb->outfifo[notify] - 1, 0, gid);
105     return 0;
106 }
```

Here is the caller graph for this function:

13.12.3.3 int _rmnotifynode (usbdevice * kb, int notify)

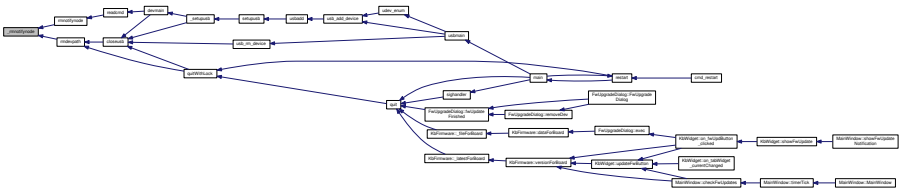
Definition at line 115 of file devnode.c.

References devpath, INDEX_OF, keyboard, usbdevice::outfifo, and OUTFIFO_MAX.

Referenced by rmdevpath(), and rmnotifynode().

```
115                                             {
116     if(notify < 0 || notify >= OUTFIFO_MAX || !kb->outfifo[notify])
117         return -1;
118     int index = INDEX_OF(kb, keyboard);
119     char outpath[strlen(devpath) + 10];
120     snprintf(outpath, sizeof(outpath), "%s%d/notify%d", devpath, index, notify);
121     // Close FIFO
122     close(kb->outfifo[notify] - 1);
123     kb->outfifo[notify] = 0;
124     // Delete node
125     int res = remove(outpath);
126     return res;
127 }
```

Here is the caller graph for this function:



13.12.3.4 void _updateconnected ( )

<devicepath> normally is /dev/input/ckb or /input/ckb.

Open the normal file under <devicepath>0/connected for writing. For each device connected, print its devicepath+number, the serial number of the usb device and the usb name of the device connected to that usb interface.

eg:

/dev/input/ckb1 0F022014ABABABABABABABABABABA999 Corsair K95 RGB Gaming Keyboard

/dev/input/ckb2 0D02303DBACBACBACBACBACBACBAC998 Corsair M65 RGB Gaming Mouse

Set the file ownership to root. If the glob var gid is explicitly set to something different from -1 (the initial value), set file permission to 640, else to 644. This is used if you start the daemon with –gid=<GID> Parameter.

Because several independent threads may call updateconnected(), protect that procedure with locking/unlocking of **devmutex**.

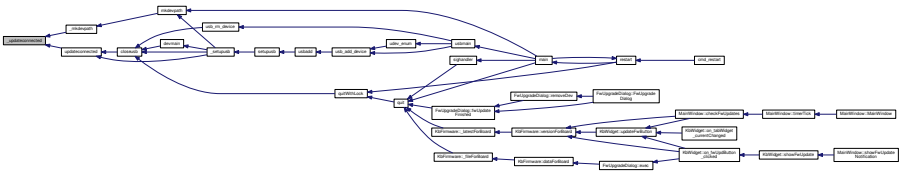
Definition at line 55 of file devnode.c.

References ckb_warn, DEV_MAX, devmutex, devpath, gid, IS_CONNECTED, keyboard, and S_GID_READ.

Referenced by _mkdevpath(), and updateconnected().

```
55                      {
56     pthread_mutex_lock(devmutex);
57     char cpath[strlen(devpath) + 12];
58     snprintf(cpath, sizeof(cpath), "%s0/connected", devpath);
59     FILE* cfile = fopen(cpath, "w");
60     if(!cfile){
61         ckb_warn("Unable to update %s: %s\n", cpath, strerror(errno));
62         pthread_mutex_unlock(devmutex);
63         return;
64     }
65     int written = 0;
66     for(int i = 1; i < DEV_MAX; i++){
67         if(IS_CONNECTED(keyboard + i)){
68             written = 1;
69             fprintf(cfile, "%s%d %s %s\n", devpath, i, keyboard[i].serial,
     keyboard[i].name);
70         }
71     }
72     if(!written)
73         fputc('\n', cfile);
74     fclose(cfile);
75     chmod(cpath, S_GID_READ);
76     if(gid >= 0)
77         chown(cpath, 0, gid);
78     pthread_mutex_unlock(devmutex);
79 }
```

Here is the caller graph for this function:



13.12.3.5 int mkdevpath (usbdevice * *kb*)

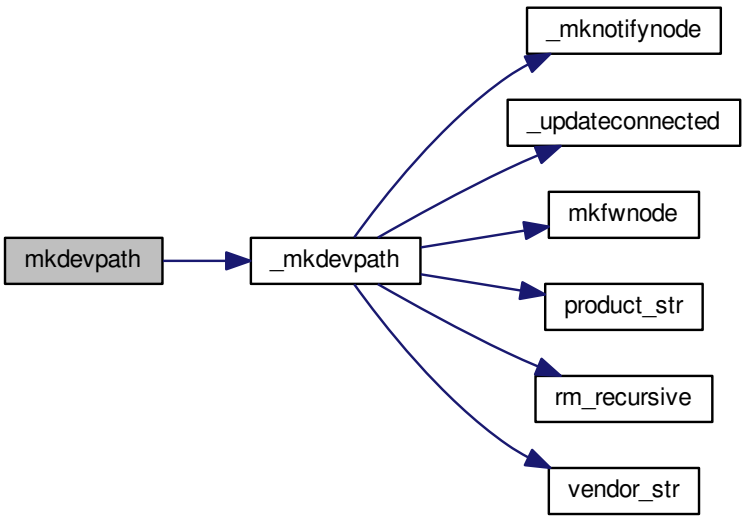
Definition at line 268 of file devnode.c.

References _mkdevpath(), euid_guard_start, and euid_guard_stop.

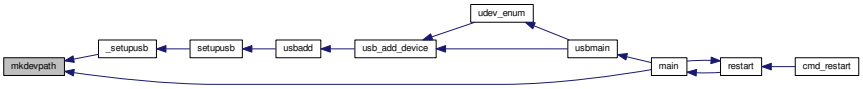
Referenced by _setupusb(), and main().

```
268                             {
269     euid_guard_start;
270     int res = _mkdevpath(kb);
271     euid_guard_stop;
272     return res;
273 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.12.3.6 int mkfwnode (usbdevice * *kb*)

Definition at line 299 of file devnode.c.

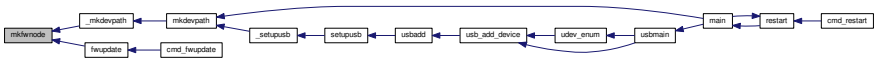
References ckb_warn, devpath, usbdevice::fwversion, gid, INDEX_OF, keyboard, usbdevice::pollrate, and S_GID_-READ.

Referenced by _mkdevpath(), and fwupdate().

```
299                          {
300     int index = INDEX_OF(kb, keyboard);
301     char fwpath[strlen(devpath) + 12];
302     snprintf(fwpath, sizeof(fwpath), "%s%d/fwversion", devpath, index);
303     FILE* fwfile = fopen(fwpath, "w");
304     if(fwfile){
305         fprintf(fwfile, "%04x", kb->fwversion);
306         fputc('\n', fwfile);
307         fclose(fwfile);
308         chmod(fwpath, S_GID_READ);
309         if(gid >= 0)
310             chown(fwpath, 0, gid);
311     } else {
312         ckb_warn("Unable to create %s: %s\n", fwpath, strerror(errno));
313         remove(fwpath);
314         return -1;
315     }
316     char ppath[strlen(devpath) + 11];
317     snprintf(ppath, sizeof(ppath), "%s%d/pollrate", devpath, index);
318     FILE* pfile = fopen(ppath, "w");
```

```
319     if(pfile){
320         fprintf(pfile, "%d ms", kb->pollrate);
321         fputc('\n', pfile);
322         fclose(pfile);
323         chmod(ppath, S_GID_READ);
324         if(gid >= 0)
325             chown(ppath, 0, gid);
326     } else {
327         ckb_warn("Unable to create %s: %s\n", fwpath, strerror(errno));
328         remove(ppath);
329         return -2;
330     }
331     return 0;
332 }
```

Here is the caller graph for this function:



13.12.3.7 int mknotifynode (usbdevice * kb, int notify)

Definition at line 108 of file devnode.c.

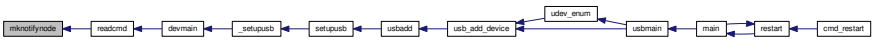
References _mknotifynode(), euid_guard_start, and euid_guard_stop.

Referenced by readcmd().

```
108                                         {
109     euid_guard_start;
110     int res = _mknotifynode(kb, notify);
111     euid_guard_stop;
112     return res;
113 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.12.3.8 unsigned readlines (int fd, readlines_ctx ctx, const char ** input)

Definition at line 353 of file devnode.c.

References _readlines_ctx::buffer, _readlines_ctx::buffersize, ckb_warn, _readlines_ctx::leftover, _readlines_ctx::leftoverlen, and MAX_BUFFER.

Referenced by devmain().

```
353                                                                 {
354     // Move any data left over from a previous read to the start of the buffer
355     char* buffer = ctx->buffer;
356     int buffersize = ctx->buffersize;
357     int leftover = ctx->leftover, leftoverlen = ctx->leftoverlen;
358     memcpy(buffer, buffer + leftover, leftoverlen);
359     // Read data from the file
360     ssize_t length = read(fd, buffer + leftoverlen, buffersize - leftoverlen);
361     length = (length < 0 ? 0 : length) + leftoverlen;
362     leftover = ctx->leftover = leftoverlen = ctx->leftoverlen = 0;
363     if(length <= 0){
364         *input = 0;
365         return 0;
366     }
367     // Continue buffering until all available input is read or there's no room left
368     while(length == buffersize){
369         if(buffersize == MAX_BUFFER)
370             break;
371         int oldsize = buffersize;
372         buffersize += 4096;
373         ctx->buffersize = buffersize;
374         buffer = ctx->buffer = realloc(buffer, buffersize + 1);
375         ssize_t length2 = read(fd, buffer + oldsize, buffersize - oldsize);
376         if(length2 <= 0)
377             break;
378         length += length2;
379     }
380     buffer[length] = 0;
381     // Input should be issued one line at a time and should end with a newline.
382     char* lastline = memrchr(buffer, '\n', length);
383     if(lastline == buffer + length - 1){
384         // If the buffer ends in a newline, process the whole string
385         *input = buffer;
386         return length;
387     } else if(lastline){
388         // Otherwise, chop off the last line but process everything else
389         *lastline = 0;
390         leftover = ctx->leftover = lastline + 1 - buffer;
391         leftoverlen = ctx->leftoverlen = length - leftover;
392         *input = buffer;
393         return leftover - 1;
394     } else {
395         // If a newline wasn't found at all, process the whole buffer next time
396         *input = 0;
397         if(length == MAX_BUFFER){
398             // Unless the buffer is completely full, in which case discard it
399             ckb_warn("Too much input (1MB). Dropping.\n");
400             return 0;
401         }
402         leftoverlen = ctx->leftoverlen = length;
403         return 0;
404     }
405 }
```

Here is the caller graph for this function:

13.12.3.9 void readlines_ctx_free (readlines_ctx *ctx*)

Definition at line 348 of file devnode.c.

References _readlines_ctx::buffer.

Referenced by devmain().

```
348                                                 {
```

```
349     free(ctx->buffer);
350     free(ctx);
351 }
```

Here is the caller graph for this function:

13.12.3.10 void readlines_ctx_init (readlines_ctx * *ctx*)

Definition at line 341 of file devnode.c.

Referenced by devmain().

```
341                                         {
342     // Allocate buffers to store data
343     *ctx = calloc(1, sizeof(struct _readlines_ctx));
344     int buffersize = (*ctx)->buffersize = 4095;
345     (*ctx)->buffer = malloc(buffersize + 1);
346 }
```

Here is the caller graph for this function:

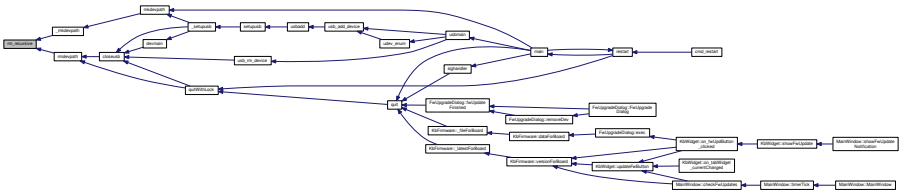
13.12.3.11 int rm_recursive (const char * *path*)

Definition at line 19 of file devnode.c.

Referenced by _mkdevpath(), and rmdevpath().

```
19                                      {
20     DIR* dir = opendir(path);
21     if(!dir)
22         return remove(path);
23     struct dirent* file;
24     while((file = readdir(dir)))
25     {
26         if(!strcmp(file->d_name, ".") || !strcmp(file->d_name, ".."))
27             continue;
28         char path2[FILENAME_MAX];
29         snprintf(path2, FILENAME_MAX, "%s/%s", path, file->d_name);
30         int stat = rm_recursive(path2);
31         if(stat != 0)
32             return stat;
33     }
34     closedir(dir);
35     return remove(path);
36 }
```

Here is the caller graph for this function:



13.12.3.12 int rmdevpath (usbdevice * *kb*)

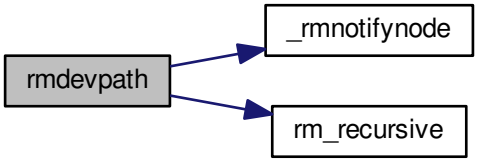
Definition at line 275 of file devnode.c.

References _rmnotifynode(), ckb_info, ckb_warn, devpath, euid_guard_start, euid_guard_stop, INDEX_OF, usbdevice::infifo, keyboard, OUTFIFO_MAX, and rm_recursive().

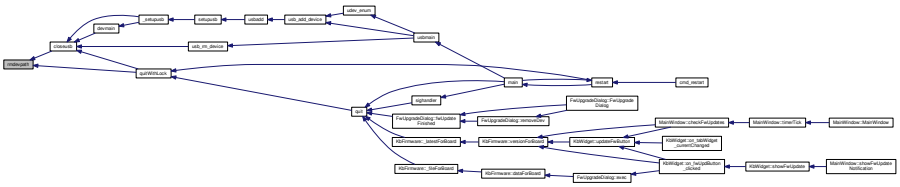
Referenced by closeusb(), and quitWithLock().

```
275                             {
276     euid_guard_start;
277     int index = INDEX_OF(kb, keyboard);
278     if(kb->infifo != 0){
279 #ifdef OS_LINUX
280         write(kb->infifo - 1, "\n", 1); // hack to prevent the FIFO thread from perma-blocking
281 #endif
282         close(kb->infifo - 1);
283         kb->infifo = 0;
284     }
285     for(int i = 0; i < OUTFIFO_MAX; i++)
286         _rmnotifynode(kb, i);
287     char path[strlen(devpath) + 2];
288     snprintf(path, sizeof(path), "%s%d", devpath, index);
289     if(rm_recursive(path) != 0 && errno != ENOENT){
290         ckb_warn("Unable to delete %s: %s\n", path, strerror(errno));
291         euid_guard_stop;
292         return -1;
293     }
294     ckb_info("Removed device path %s\n", path);
295     euid_guard_stop;
296     return 0;
297 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.12.3.13 int rmnotifynode (usbdevice * *kb*, int *notify*)

Definition at line 129 of file devnode.c.

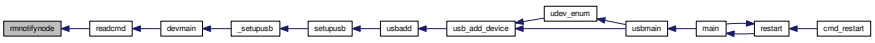
References _rmnotifynode(), euid_guard_start, and euid_guard_stop.

Referenced by readcmd().

```
129                                                {
130     euid_guard_start;
131     int res = _rmnotifynode(kb, notify);
132     euid_guard_stop;
133     return res;
134 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.12.3.14 void updateconnected ( )

Definition at line 81 of file devnode.c.

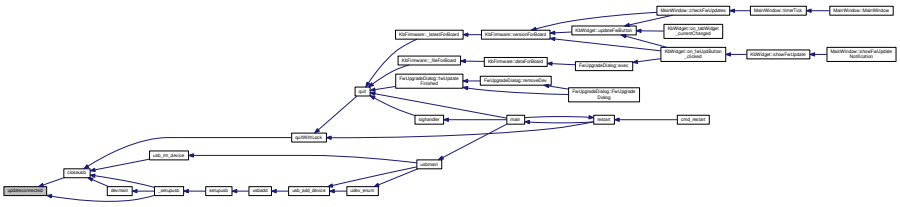
References _updateconnected(), euid_guard_start, and euid_guard_stop.

Referenced by _setupusb(), and closeusb().

```
81                          {
82     euid_guard_start;
83     _updateconnected();
84     euid_guard_stop;
85 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.12.4 Variable Documentation

13.12.4.1 const char* const devpath = "/dev/input/ckb"

Definition at line 11 of file devnode.c.

Referenced by _mkdevpath(), _mknotifynode(), _rmnotifynode(), _updateconnected(), mkfwnode(), and rmdevpath().

13.12.4.2 long gid = -1

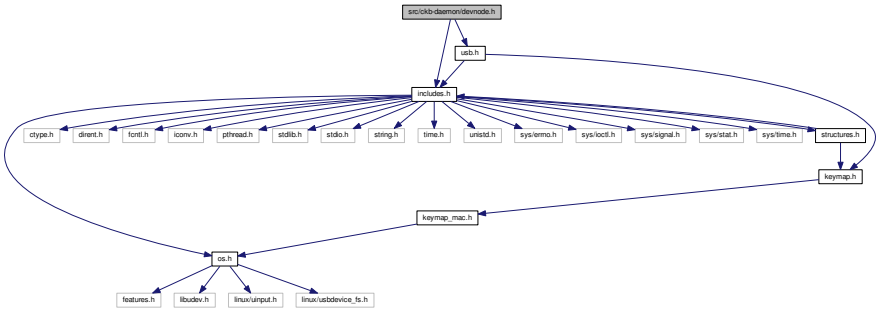
Definition at line 16 of file devnode.c.

Referenced by _mkdevpath(), _mknotifynode(), _updateconnected(), main(), and mkfwnode().

13.13 src/ckb-daemon/devnode.h File Reference

```
#include "includes.h"
#include "usb.h"
```

Include dependency graph for devnode.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define S_READDIR (S_IRWXU | S_IRGRP | S_IROTH | S_IXGRP | S_IXOTH)
- #define S_READ (S_IRUSR | S_IRGRP | S_IROTH | S_IWUSR)
- #define S_READWRITE (S_IRUSR | S_IRGRP | S_IROTH | S_IWUSR | S_IWGRP | S_IWOTH)
- #define S_CUSTOM (S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP)
- #define S_CUSTOM_R (S_IRUSR | S_IWUSR | S_IRGRP)

Typedefs

- typedef struct _readlines_ctx * readlines_ctx

  Custom readline is needed for FIFOs. fopen()/getline() will die if the data is sent in too fast.

Functions

- void updateconnected ()

  Update the list of connected devices.
- int mkdevpath (usbdevice *kb)

  Create a dev path for the keyboard at index. Returns 0 on success.
- int rmdevpath (usbdevice *kb)

  Remove the dev path for the keyboard at index. Returns 0 on success.
- int mknotifynode (usbdevice *kb, int notify)

  Creates a notification node for the specified keyboard.
- int rmnotifynode (usbdevice *kb, int notify)

  Removes a notification node for the specified keyboard.
- int mkfwnode (usbdevice *kb)

  Writes a keyboard's firmware version and poll rate to its device node.
- void readlines_ctx_init (readlines_ctx *ctx)
- void readlines_ctx_free (readlines_ctx ctx)
- unsigned readlines (int fd, readlines_ctx ctx, const char **input)

Variables

- const char *const devpath

  Device path base ("/dev/input/ckb" or "/var/run/ckb")
- long gid

  Group ID for the control nodes. -1 to give read/write access to everybody.

13.13.1 Macro Definition Documentation

13.13.1.1 #define S_CUSTOM (S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP)

Definition at line 17 of file devnode.h.

Referenced by _mkdevpath().

13.13.1.2 #define S_CUSTOM_R (S_IRUSR | S_IWUSR | S_IRGRP)

Definition at line 18 of file devnode.h.

13.13.1.3 #define S_READ (S_IRUSR | S_IRGRP | S_IROTH | S_IWUSR)

Definition at line 15 of file devnode.h.

Referenced by _mkdevpath().

13.13.1.4 #define S_READDIR (S_IRWXU | S_IRGRP | S_IROTH | S_IXGRP | S_IXOTH)

Definition at line 14 of file devnode.h.

Referenced by _mkdevpath().

13.13.1.5 #define S_READWRITE (S_IRUSR | S_IRGRP | S_IROTH | S_IWUSR | S_IWGRP | S_IWOTH)

Definition at line 16 of file devnode.h.

Referenced by _mkdevpath().

13.13.2 Typedef Documentation

13.13.2.1 typedef struct _readlines_ctx* readlines_ctx

Definition at line 39 of file devnode.h.

13.13.3 Function Documentation

13.13.3.1 int mkdevpath (usbdevice * *kb*)

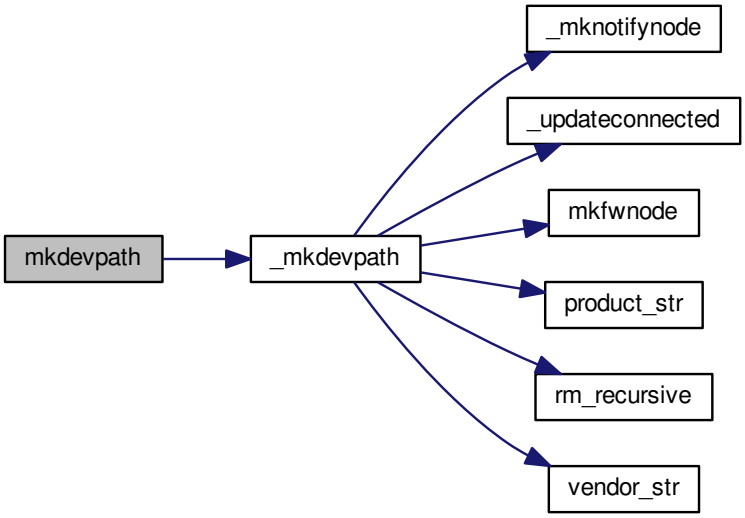
Definition at line 268 of file devnode.c.

References _mkdevpath(), euid_guard_start, and euid_guard_stop.

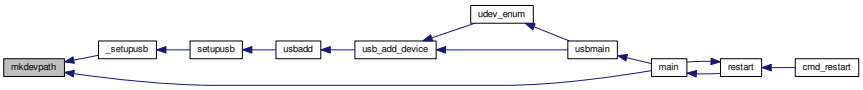
Referenced by _setupusb(), and main().

```
268                             {
269     euid_guard_start;
270     int res = _mkdevpath(kb);
271     euid_guard_stop;
272     return res;
273 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.13.3.2 int mkfwnode (usbdevice * *kb*)

Definition at line 299 of file devnode.c.

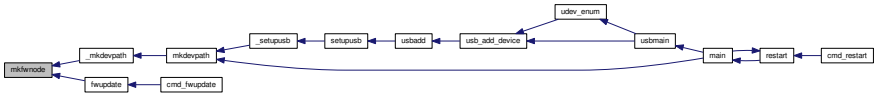
References ckb_warn, devpath, usbdevice::fwversion, gid, INDEX_OF, keyboard, usbdevice::pollrate, and S_GID_-READ.

Referenced by _mkdevpath(), and fwupdate().

```
299                             {
300     int index = INDEX_OF(kb, keyboard);
301     char fwpath[strlen(devpath) + 12];
302     snprintf(fwpath, sizeof(fwpath), "%s%d/fwversion", devpath, index);
303     FILE* fwfile = fopen(fwpath, "w");
304     if(fwfile){
305         fprintf(fwfile, "%04x", kb->fwversion);
306         fputc('\n', fwfile);
307         fclose(fwfile);
308         chmod(fwpath, S_GID_READ);
309         if(gid >= 0)
```

```
310             chown(fwpath, 0, gid);
311     } else {
312         ckb_warn("Unable to create %s: %s\n", fwpath, strerror(errno));
313         remove(fwpath);
314         return -1;
315     }
316     char ppath[strlen(devpath) + 11];
317     snprintf(ppath, sizeof(ppath), "%s%d/pollrate", devpath, index);
318     FILE* pfile = fopen(ppath, "w");
319     if(pfile){
320         fprintf(pfile, "%d ms", kb->pollrate);
321         fputc('\n', pfile);
322         fclose(pfile);
323         chmod(ppath, S_GID_READ);
324         if(gid >= 0)
325             chown(ppath, 0, gid);
326     } else {
327         ckb_warn("Unable to create %s: %s\n", fwpath, strerror(errno));
328         remove(ppath);
329         return -2;
330     }
331     return 0;
332 }
```

Here is the caller graph for this function:



13.13.3.3 int mknotifynode (usbdevice * kb, int notify)

Definition at line 108 of file devnode.c.

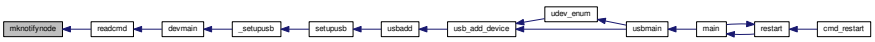
References _mknotifynode(), euid_guard_start, and euid_guard_stop.

Referenced by readcmd().

```
108                                             {
109     euid_guard_start;
110     int res = _mknotifynode(kb, notify);
111     euid_guard_stop;
112     return res;
113 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.13.3.4 unsigned readlines (int *fd*, readlines_ctx *ctx*, const char ** *input*)

Definition at line 353 of file devnode.c.

References _readlines_ctx::buffer, _readlines_ctx::buffersize, ckb_warn, _readlines_ctx::leftover, _readlines_ctx::leftoverlen, and MAX_BUFFER.

Referenced by devmain().

```
353                                                                 {
354     // Move any data left over from a previous read to the start of the buffer
355     char* buffer = ctx->buffer;
356     int buffersize = ctx->buffersize;
357     int leftover = ctx->leftover, leftoverlen = ctx->leftoverlen;
358     memcpy(buffer, buffer + leftover, leftoverlen);
359     // Read data from the file
360     ssize_t length = read(fd, buffer + leftoverlen, buffersize - leftoverlen);
361     length = (length < 0 ? 0 : length) + leftoverlen;
362     leftover = ctx->leftover = leftoverlen = ctx->leftoverlen = 0;
363     if(length <= 0){
364         *input = 0;
365         return 0;
366     }
367     // Continue buffering until all available input is read or there's no room left
368     while(length == buffersize){
369         if(buffersize == MAX_BUFFER)
370             break;
371         int oldsize = buffersize;
372         buffersize += 4096;
373         ctx->buffersize = buffersize;
374         buffer = ctx->buffer = realloc(buffer, buffersize + 1);
375         ssize_t length2 = read(fd, buffer + oldsize, buffersize - oldsize);
376         if(length2 <= 0)
377             break;
378         length += length2;
379     }
380     buffer[length] = 0;
381     // Input should be issued one line at a time and should end with a newline.
382     char* lastline = memrchr(buffer, '\n', length);
383     if(lastline == buffer + length - 1){
384         // If the buffer ends in a newline, process the whole string
385         *input = buffer;
386         return length;
387     } else if(lastline){
388         // Otherwise, chop off the last line but process everything else
389         *lastline = 0;
390         leftover = ctx->leftover = lastline + 1 - buffer;
391         leftoverlen = ctx->leftoverlen = length - leftover;
392         *input = buffer;
393         return leftover - 1;
394     } else {
395         // If a newline wasn't found at all, process the whole buffer next time
396         *input = 0;
397         if(length == MAX_BUFFER){
398             // Unless the buffer is completely full, in which case discard it
399             ckb_warn("Too much input (1MB). Dropping.\n");
400             return 0;
401         }
402         leftoverlen = ctx->leftoverlen = length;
403         return 0;
404     }
405 }
```

Here is the caller graph for this function:

13.13.3.5 void readlines_ctx_free (readlines_ctx *ctx*)

Definition at line 348 of file devnode.c.

References _readlines_ctx::buffer.

Referenced by devmain().

```
348                                         {
349     free(ctx->buffer);
350     free(ctx);
351 }
```

Here is the caller graph for this function:

13.13.3.6 void readlines_ctx_init (readlines_ctx * *ctx*)

Definition at line 341 of file devnode.c.

Referenced by devmain().

```
341                                          {
342     // Allocate buffers to store data
343     *ctx = calloc(1, sizeof(struct _readlines_ctx));
344     int buffersize = (*ctx)->buffersize = 4095;
345     (*ctx)->buffer = malloc(buffersize + 1);
346 }
```

Here is the caller graph for this function:

13.13.3.7 int rmdevpath (usbdevice * *kb*)

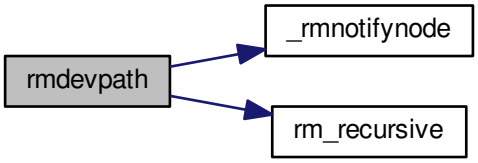
Definition at line 275 of file devnode.c.

References _rmnotifynode(), ckb_info, ckb_warn, devpath, euid_guard_start, euid_guard_stop, INDEX_OF, usbdevice::infifo, keyboard, OUTFIFO_MAX, and rm_recursive().

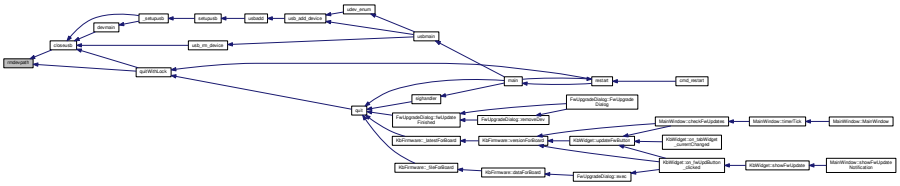
Referenced by closeusb(), and quitWithLock().

```
275                                {
276     euid_guard_start;
277     int index = INDEX_OF(kb, keyboard);
278     if(kb->infifo != 0){
279 #ifdef OS_LINUX
280         write(kb->infifo - 1, "\n", 1); // hack to prevent the FIFO thread from perma-blocking
281 #endif
282         close(kb->infifo - 1);
283         kb->infifo = 0;
284     }
285     for(int i = 0; i < OUTFIFO_MAX; i++)
286         _rmnotifynode(kb, i);
287     char path[strlen(devpath) + 2];
288     snprintf(path, sizeof(path), "%s%d", devpath, index);
289     if(rm_recursive(path) != 0 && errno != ENOENT){
290         ckb_warn("Unable to delete %s: %s\n", path, strerror(errno));
291         euid_guard_stop;
292         return -1;
293     }
294     ckb_info("Removed device path %s\n", path);
295     euid_guard_stop;
296     return 0;
297 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.13.3.8 int rmnotifynode (usbdevice * *kb*, int *notify*)

Definition at line 129 of file devnode.c.

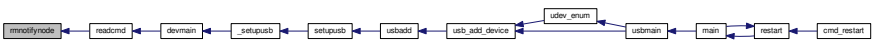
References _rmnotifynode(), euid_guard_start, and euid_guard_stop.

Referenced by readcmd().

```
129                                             {
130     euid_guard_start;
131     int res = _rmnotifynode(kb, notify);
132     euid_guard_stop;
133     return res;
134 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.13.3.9 void updateconnected ( )

Definition at line 81 of file devnode.c.

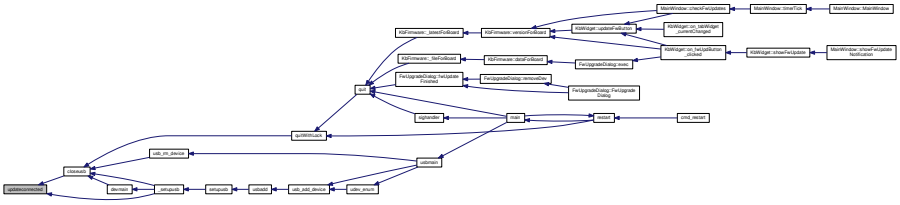
References _updateconnected(), euid_guard_start, and euid_guard_stop.

Referenced by _setupusb(), and closeusb().

```
81                          {
82     euid_guard_start;
83     _updateconnected();
84     euid_guard_stop;
85 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.13.4 Variable Documentation

13.13.4.1 const char* const devpath

Definition at line 8 of file devnode.h.

Referenced by MainWindow::MainWindow().

13.13.4.2 long gid

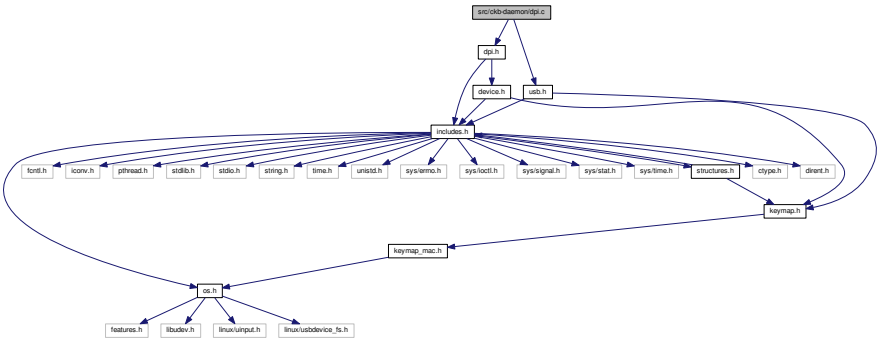
Definition at line 16 of file devnode.c.

Referenced by _mkdevpath(), _mknotifynode(), _updateconnected(), main(), and mkfwnode().

13.14 src/ckb-daemon/dpi.c File Reference

```
#include "dpi.h"
#include "usb.h"
```

Include dependency graph for dpi.c:



Functions

- void cmd_dpi (usbdevice ∗kb, usbmode ∗mode, int dummy, const char ∗stages, const char ∗values)
- void cmd_dpisel (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗stage)
- void cmd_lift (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗height)
- void cmd_snap (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗enable)
- char ∗ printdpi (const dpiset ∗dpi, const usbdevice ∗kb)
- int updatedpi (usbdevice ∗kb, int force)
- int savedpi (usbdevice ∗kb, dpiset ∗dpi, lighting ∗light)
- int loaddpi (usbdevice ∗kb, dpiset ∗dpi, lighting ∗light)

13.14.1 Function Documentation

13.14.1.1 void cmd_dpi (usbdevice ∗ *kb*, usbmode ∗ *mode*, int *dummy*, const char ∗ *stages*, const char ∗ *values*)

Definition at line 4 of file dpi.c.

References usbmode::dpi, DPI_COUNT, dpiset::enabled, left, dpiset::x, x, dpiset::y, and y.

```
4                                                                                                                {
5       (void)kb;
6       (void)dummy;
7
8       int disable = 0;
9       ushort x, y;
10       // Try to scan X,Y values
11       if(sscanf(values, "%hu,%hu", &x, &y) != 2){
12           // If that doesn't work, scan single number
13           if(sscanf(values, "%hu", &x) == 1)
14               y = x;
15           else if(!strncmp(values, "off", 3))
16               // If the right side says "off", disable the level(s)
17               disable = 1;
18           else
19               // Otherwise, quit
20               return;
21       }
22       if((x == 0 || y == 0) && !disable)
23           return;
24       // Scan the left side for stage numbers (comma-separated)
25       int left = strlen(stages);
26       int position = 0, field = 0;
27       char stagename[3];
28       while(position < left && sscanf(stages + position, "%2[^,]%n", stagename, &field) == 1){
29           uchar stagenum;
30           if(sscanf(stagename, "%hhu", &stagenum) && stagenum < DPI_COUNT){
31               // Set DPI for this stage
32               if(disable){
33                   mode->dpi.enabled &= ~(1 << stagenum);
34                   mode->dpi.x[stagenum] = 0;
35                   mode->dpi.y[stagenum] = 0;
```

```
36              } else {
37                  mode->dpi.enabled |= 1 << stagenum;
38                  mode->dpi.x[stagenum] = x;
39                  mode->dpi.y[stagenum] = y;
40              }
41          }
42          if(stages[position += field] == ',')
43              position++;
44      }
45  }
```

13.14.1.2 void cmd_dpisel (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *stage*)

Definition at line 47 of file dpi.c.

References dpiset::current, usbmode::dpi, and DPI_COUNT.

```
47                                                                                  {
48      (void)kb;
49      (void)dummy1;
50      (void)dummy2;
51
52      uchar stagenum;
53      if(sscanf(stage, "%hhu", &stagenum) != 1)
54          return;
55      if(stagenum > DPI_COUNT)
56          return;
57      mode->dpi.current = stagenum;
58  }
```

13.14.1.3 void cmd_lift (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *height*)

Definition at line 60 of file dpi.c.

References usbmode::dpi, dpiset::lift, LIFT_MAX, and LIFT_MIN.

```
60                                                                                  {
61      (void)kb;
62      (void)dummy1;
63      (void)dummy2;
64
65      uchar heightnum;
66      if(sscanf(height, "%hhu", &heightnum) != 1)
67          return;
68      if(heightnum > LIFT_MAX || heightnum < LIFT_MIN)
69          return;
70      mode->dpi.lift = heightnum;
71  }
```

13.14.1.4 void cmd_snap (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *enable*)

Definition at line 73 of file dpi.c.

References usbmode::dpi, and dpiset::snap.

```
73                                                                                  {
74      (void)kb;
75      (void)dummy1;
76      (void)dummy2;
77
78      if(!strcmp(enable, "on"))
79          mode->dpi.snap = 1;
80      if(!strcmp(enable, "off"))
81          mode->dpi.snap = 0;
82  }
```

13.14.1.5 int loaddpi (usbdevice * *kb,* dpiset * *dpi,* lighting * *light*)

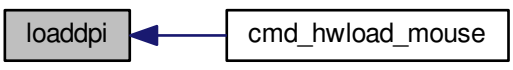
Definition at line 222 of file dpi.c.

References lighting::b, ckb_err, dpiset::current, DPI_COUNT, dpiset::enabled, lighting::g, LED_MOUSE, dpiset::lift, LIFT_MAX, LIFT_MIN, MSG_SIZE, N_MOUSE_ZONES, lighting::r, dpiset::snap, usbrecv, dpiset::x, and dpiset::y.

Referenced by cmd_hwload_mouse().

```
222                                                     {
223     // Ask for settings
224     uchar data_pkt[4][MSG_SIZE] = {
225         { 0x0e, 0x13, 0x05, 1, },
226         { 0x0e, 0x13, 0x02, 1, },
227         { 0x0e, 0x13, 0x03, 1, },
228         { 0x0e, 0x13, 0x04, 1, }
229     };
230     uchar in_pkt[4][MSG_SIZE];
231     for(int i = 0; i < 4; i++){
232         if(!usbrecv(kb, data_pkt[i], in_pkt[i]))
233             return -2;
234         if(memcmp(in_pkt[i], data_pkt[i], 4)){
235             ckb_err("Bad input header\n");
236             return -3;
237         }
238     }
239     // Copy data from device
240     dpi->enabled = in_pkt[0][4];
241     dpi->enabled &= (1 << DPI_COUNT) - 1;
242     dpi->current = in_pkt[1][4];
243     if(dpi->current >= DPI_COUNT)
244         dpi->current = 0;
245     dpi->lift = in_pkt[2][4];
246     if(dpi->lift < LIFT_MIN || dpi->lift > LIFT_MAX)
247         dpi->lift = LIFT_MIN;
248     dpi->snap = !!in_pkt[3][4];
249
250     // Get X/Y DPIs
251     for(int i = 0; i < DPI_COUNT; i++){
252         uchar data_pkt[MSG_SIZE] = { 0x0e, 0x13, 0xd0, 1 };
253         uchar in_pkt[MSG_SIZE];
254         data_pkt[2] |= i;
255         if(!usbrecv(kb, data_pkt, in_pkt))
256             return -2;
257         if(memcmp(in_pkt, data_pkt, 4)){
258             ckb_err("Bad input header\n");
259             return -3;
260         }
261         // Copy to profile
262         dpi->x[i] = *(ushort*)(in_pkt + 5);
263         dpi->y[i] = *(ushort*)(in_pkt + 7);
264         light->r[LED_MOUSE + N_MOUSE_ZONES + i] = in_pkt[9];
265         light->g[LED_MOUSE + N_MOUSE_ZONES + i] = in_pkt[10];
266         light->b[LED_MOUSE + N_MOUSE_ZONES + i] = in_pkt[11];
267     }
268     // Finished. Set SW DPI light to the current hardware level
269     light->r[LED_MOUSE + 2] = light->r[LED_MOUSE +
    N_MOUSE_ZONES + dpi->current];
270     light->g[LED_MOUSE + 2] = light->g[LED_MOUSE +
    N_MOUSE_ZONES + dpi->current];
271     light->b[LED_MOUSE + 2] = light->b[LED_MOUSE +
    N_MOUSE_ZONES + dpi->current];
272     return 0;
273 }
```

Here is the caller graph for this function:



13.14.1.6 char* printdpi (const dpiset * *dpi,* const usbdevice * *kb*)

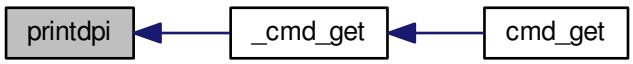
Definition at line 84 of file dpi.c.

References _readlines_ctx::buffer, DPI_COUNT, dpiset::enabled, dpiset::x, and dpiset::y.

Referenced by _cmd_get().

```
84                                                      {
85     (void)kb;
86
87     // Print all DPI settings
88     const int BUFFER_LEN = 100;
89     char* buffer = malloc(BUFFER_LEN);
90     int length = 0;
91     for(int i = 0; i < DPI_COUNT; i++){
92         // Print the stage number
93         int newlen = 0;
94         snprintf(buffer + length, BUFFER_LEN - length, length == 0 ? "%d%n" : " %d%n", i, &newlen);
95         length += newlen;
96         // Print the DPI settings
97         if(!(dpi->enabled & (1 << i)))
98             snprintf(buffer + length, BUFFER_LEN - length, ":off%n", &newlen);
99         else
100             snprintf(buffer + length, BUFFER_LEN - length, ":%u,%u%n", dpi->x[i], dpi->
      y[i], &newlen);
101         length += newlen;
102     }
103     return buffer;
104 }
```

Here is the caller graph for this function:



13.14.1.7 int savedpi (usbdevice * *kb,* dpiset * *dpi,* lighting * *light*)

Definition at line 194 of file dpi.c.

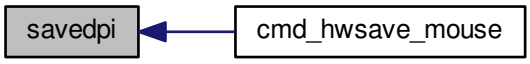
References lighting::b, dpiset::current, DPI_COUNT, dpiset::enabled, lighting::g, LED_MOUSE, dpiset::lift, MSG_-SIZE, N_MOUSE_ZONES, lighting::r, dpiset::snap, usbsend, dpiset::x, and dpiset::y.

Referenced by cmd_hwsave_mouse().

```
194                                                       {
195     // Send X/Y DPIs
196     for(int i = 0; i < DPI_COUNT; i++){
197         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0xd0, 1 };
198         data_pkt[2] |= i;
199         *(ushort*)(data_pkt + 5) = dpi->x[i];
200         *(ushort*)(data_pkt + 7) = dpi->y[i];
201         // Save the RGB value for this setting too
202         data_pkt[9] = light->r[LED_MOUSE + N_MOUSE_ZONES + i];
203         data_pkt[10] = light->g[LED_MOUSE + N_MOUSE_ZONES + i];
204         data_pkt[11] = light->b[LED_MOUSE + N_MOUSE_ZONES + i];
205         if(!usbsend(kb, data_pkt, 1))
206             return -1;
207     }
208
209     // Send settings
210     uchar data_pkt[4][MSG_SIZE] = {
211         { 0x07, 0x13, 0x05, 1, dpi->enabled },
212         { 0x07, 0x13, 0x02, 1, dpi->current },
213         { 0x07, 0x13, 0x03, 1, dpi->lift },
```

```
214         { 0x07, 0x13, 0x04, 1, dpi->snap, 0x05 }
215     };
216     if(!usbsend(kb, data_pkt[0], 4))
217         return -2;
218     // Finished
219     return 0;
220 }
```

Here is the caller graph for this function:



13.14.1.8 int updatedpi (usbdevice * *kb*, int *force*)

Definition at line 106 of file dpi.c.

References usbdevice::active, dpiset::current, usbprofile::currentmode, usbmode::dpi, DPI_COUNT, dpiset::enabled, dpiset::forceupdate, usbprofile::lastdpi, dpiset::lift, MSG_SIZE, usbdevice::profile, dpiset::snap, usbsend, dpiset::x, and dpiset::y.

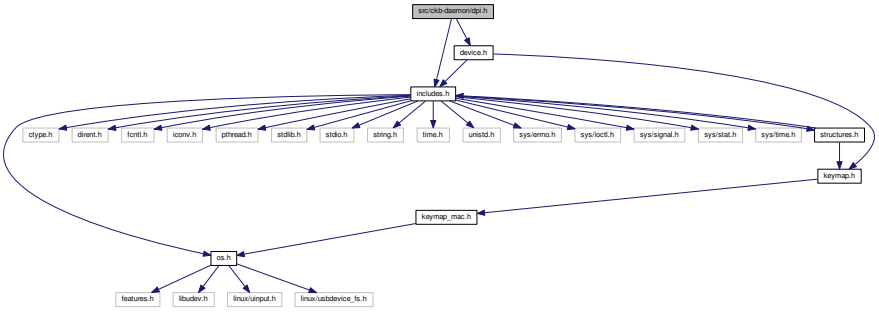
```
106                                     {
107     if(!kb->active)
108         return 0;
109     dpiset* lastdpi = &kb->profile->lastdpi;
110     dpiset* newdpi = &kb->profile->currentmode->dpi;
111     // Don't do anything if the settings haven't changed
112     if(!force && !lastdpi->forceupdate && !newdpi->forceupdate
113             && !memcmp(lastdpi, newdpi, sizeof(dpiset)))
114         return 0;
115     lastdpi->forceupdate = newdpi->forceupdate = 0;
116
117     if (newdpi->current != lastdpi->current) {
118         // Before we switch the current DPI stage, make sure the stage we are
119         // switching to is both enabled and configured to the correct DPI.
120
121         // Enable the stage if necessary.
122         if ((lastdpi->enabled & 1 << newdpi->current) == 0) {
123             uchar newenabled;
124             // If the new enabled flags contain both the current and previous
125             // stages, use it.
126             if (newdpi->enabled & 1 << newdpi->current &&
127                 newdpi->enabled & 1 << lastdpi->current) {
128                 newenabled = newdpi->enabled;
129             } else {
130                 // Otherwise just enable the new stage. We'll write the actual
131                 // requested flags after switching stages.
132                 newenabled = lastdpi->enabled | 1 << newdpi->current;
133             }
134             uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x05, 0, newenabled };
135             if(!usbsend(kb, data_pkt, 1))
136                 return -2;
137             // Cache the flags we wrote.
138             lastdpi->enabled = newenabled;
139         }
140         // Set the DPI for the new stage if necessary.
141         if (newdpi->x[newdpi->current] != lastdpi->x[newdpi->current] ||
142             newdpi->y[newdpi->current] != lastdpi->y[newdpi->current]) {
143             uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0xd0, 0 };
144             data_pkt[2] |= newdpi->current;
145             *(ushort*)(data_pkt + 5) = newdpi->x[newdpi->current];
146             *(ushort*)(data_pkt + 7) = newdpi->y[newdpi->current];
147             if(!usbsend(kb, data_pkt, 1))
148                 return -1;
149             // Set these values in the cache so we don't rewrite them.
150             lastdpi->x[newdpi->current] = newdpi->x[newdpi->current];
151             lastdpi->y[newdpi->current] = newdpi->y[newdpi->current];
```

```
152         }
153         // Set current DPI stage.
154         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x02, 0, newdpi->
    current };
155         if(!usbsend(kb, data_pkt, 1))
156             return -2;
157     }
158
159     // Send X/Y DPIs. We've changed to the new stage already so these can be set
160     // safely.
161     for(int i = 0; i < DPI_COUNT; i++){
162         if (newdpi->x[i] == lastdpi->x[i] && newdpi->y[i] == lastdpi->y[i])
163             continue;
164         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0xd0, 0 };
165         data_pkt[2] |= i;
166         *(ushort*)(data_pkt + 5) = newdpi->x[i];
167         *(ushort*)(data_pkt + 7) = newdpi->y[i];
168         if(!usbsend(kb, data_pkt, 1))
169             return -1;
170     }
171
172     // Send settings
173     if (newdpi->enabled != lastdpi->enabled) {
174         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x05, 0, newdpi->
    enabled };
175         if(!usbsend(kb, data_pkt, 1))
176             return -2;
177     }
178     if (newdpi->lift != lastdpi->lift) {
179         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x03, 0, newdpi->
    lift };
180         if(!usbsend(kb, data_pkt, 1))
181             return -2;
182     }
183     if (newdpi->snap != lastdpi->snap) {
184         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x04, 0, newdpi->
    snap, 0x05 };
185         if(!usbsend(kb, data_pkt, 1))
186             return -2;
187     }
188
189     // Finished
190     memcpy(lastdpi, newdpi, sizeof(dpiset));
191     return 0;
192 }
```

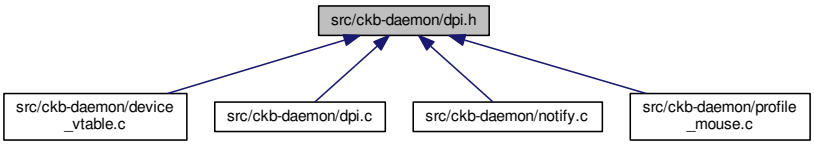
13.15 src/ckb-daemon/dpi.h File Reference

```
#include "includes.h"
#include "device.h"
```

Include dependency graph for dpi.h:



This graph shows which files directly or indirectly include this file:



Functions

- int updatedpi (usbdevice ∗kb, int force)
- int savedpi (usbdevice ∗kb, dpiset ∗dpi, lighting ∗light)
- int loaddpi (usbdevice ∗kb, dpiset ∗dpi, lighting ∗light)
- char ∗ printdpi (const dpiset ∗dpi, const usbdevice ∗kb)
- void cmd_dpi (usbdevice ∗kb, usbmode ∗mode, int dummy, const char ∗stages, const char ∗values)
- void cmd_dpisel (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗stage)
- void cmd_lift (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗height)
- void cmd_snap (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗enable)

13.15.1 Function Documentation

13.15.1.1 void cmd_dpi (usbdevice ∗ *kb*, usbmode ∗ *mode*, int *dummy*, const char ∗ *stages*, const char ∗ *values*)

Definition at line 4 of file dpi.c.

References usbmode::dpi, DPI_COUNT, dpiset::enabled, left, dpiset::x, x, dpiset::y, and y.

```
4                                                                                                {
5     (void)kb;
6     (void)dummy;
7
8     int disable = 0;
9     ushort x, y;
10     // Try to scan X,Y values
11     if(sscanf(values, "%hu,%hu", &x, &y) != 2){
12         // If that doesn't work, scan single number
13         if(sscanf(values, "%hu", &x) == 1)
14             y = x;
15         else if(!strncmp(values, "off", 3))
16             // If the right side says "off", disable the level(s)
17             disable = 1;
18         else
19             // Otherwise, quit
20             return;
21     }
22     if((x == 0 || y == 0) && !disable)
23         return;
24     // Scan the left side for stage numbers (comma-separated)
25     int left = strlen(stages);
26     int position = 0, field = 0;
27     char stagename[3];
28     while(position < left && sscanf(stages + position, "%2[^,]%n", stagename, &field) == 1){
29         uchar stagenum;
30         if(sscanf(stagename, "%hhu", &stagenum) && stagenum < DPI_COUNT){
31             // Set DPI for this stage
32             if(disable){
33                 mode->dpi.enabled &= ~(1 << stagenum);
34                 mode->dpi.x[stagenum] = 0;
35                 mode->dpi.y[stagenum] = 0;
36             } else {
37                 mode->dpi.enabled |= 1 << stagenum;
38                 mode->dpi.x[stagenum] = x;
39                 mode->dpi.y[stagenum] = y;
40             }
41         }
```

```
42          if(stages[position += field] == ',')
43              position++;
44      }
45  }
```

13.15.1.2 void cmd_dpisel (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *stage*)

Definition at line 47 of file dpi.c.

References dpiset::current, usbmode::dpi, and DPI_COUNT.

```
47                                                                                  {
48      (void)kb;
49      (void)dummy1;
50      (void)dummy2;
51
52      uchar stagenum;
53      if(sscanf(stage, "%hhu", &stagenum) != 1)
54          return;
55      if(stagenum > DPI_COUNT)
56          return;
57      mode->dpi.current = stagenum;
58  }
```

13.15.1.3 void cmd_lift (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *height*)

Definition at line 60 of file dpi.c.

References usbmode::dpi, dpiset::lift, LIFT_MAX, and LIFT_MIN.

```
60                                                                                  {
61      (void)kb;
62      (void)dummy1;
63      (void)dummy2;
64
65      uchar heightnum;
66      if(sscanf(height, "%hhu", &heightnum) != 1)
67          return;
68      if(heightnum > LIFT_MAX || heightnum < LIFT_MIN)
69          return;
70      mode->dpi.lift = heightnum;
71  }
```

13.15.1.4 void cmd_snap (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *enable*)

Definition at line 73 of file dpi.c.

References usbmode::dpi, and dpiset::snap.

```
73                                                                                  {
74      (void)kb;
75      (void)dummy1;
76      (void)dummy2;
77
78      if(!strcmp(enable, "on"))
79          mode->dpi.snap = 1;
80      if(!strcmp(enable, "off"))
81          mode->dpi.snap = 0;
82  }
```

13.15.1.5 int loaddpi (usbdevice * *kb*, dpiset * *dpi*, lighting * *light*)

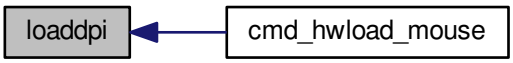
Definition at line 222 of file dpi.c.

References lighting::b, ckb_err, dpiset::current, DPI_COUNT, dpiset::enabled, lighting::g, LED_MOUSE, dpiset::lift, LIFT_MAX, LIFT_MIN, MSG_SIZE, N_MOUSE_ZONES, lighting::r, dpiset::snap, usbrecv, dpiset::x, and dpiset::y.

Referenced by cmd_hwload_mouse().

```
222                                                     {
223     // Ask for settings
224     uchar data_pkt[4][MSG_SIZE] = {
225         { 0x0e, 0x13, 0x05, 1, },
226         { 0x0e, 0x13, 0x02, 1, },
227         { 0x0e, 0x13, 0x03, 1, },
228         { 0x0e, 0x13, 0x04, 1, }
229     };
230     uchar in_pkt[4][MSG_SIZE];
231     for(int i = 0; i < 4; i++){
232         if(!usbrecv(kb, data_pkt[i], in_pkt[i]))
233             return -2;
234         if(memcmp(in_pkt[i], data_pkt[i], 4)){
235             ckb_err("Bad input header\n");
236             return -3;
237         }
238     }
239     // Copy data from device
240     dpi->enabled = in_pkt[0][4];
241     dpi->enabled &= (1 << DPI_COUNT) - 1;
242     dpi->current = in_pkt[1][4];
243     if(dpi->current >= DPI_COUNT)
244         dpi->current = 0;
245     dpi->lift = in_pkt[2][4];
246     if(dpi->lift < LIFT_MIN || dpi->lift > LIFT_MAX)
247         dpi->lift = LIFT_MIN;
248     dpi->snap = !!in_pkt[3][4];
249
250     // Get X/Y DPIs
251     for(int i = 0; i < DPI_COUNT; i++){
252         uchar data_pkt[MSG_SIZE] = { 0x0e, 0x13, 0xd0, 1 };
253         uchar in_pkt[MSG_SIZE];
254         data_pkt[2] |= i;
255         if(!usbrecv(kb, data_pkt, in_pkt))
256             return -2;
257         if(memcmp(in_pkt, data_pkt, 4)){
258             ckb_err("Bad input header\n");
259             return -3;
260         }
261         // Copy to profile
262         dpi->x[i] = *(ushort*)(in_pkt + 5);
263         dpi->y[i] = *(ushort*)(in_pkt + 7);
264         light->r[LED_MOUSE + N_MOUSE_ZONES + i] = in_pkt[9];
265         light->g[LED_MOUSE + N_MOUSE_ZONES + i] = in_pkt[10];
266         light->b[LED_MOUSE + N_MOUSE_ZONES + i] = in_pkt[11];
267     }
268     // Finished. Set SW DPI light to the current hardware level
269     light->r[LED_MOUSE + 2] = light->r[LED_MOUSE +
    N_MOUSE_ZONES + dpi->current];
270     light->g[LED_MOUSE + 2] = light->g[LED_MOUSE +
    N_MOUSE_ZONES + dpi->current];
271     light->b[LED_MOUSE + 2] = light->b[LED_MOUSE +
    N_MOUSE_ZONES + dpi->current];
272     return 0;
273 }
```

Here is the caller graph for this function:



13.15.1.6 char* printdpi (const dpiset * *dpi*, const usbdevice * *kb*)

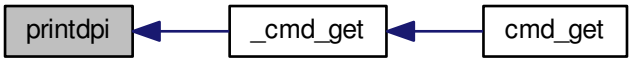
Definition at line 84 of file dpi.c.

References _readlines_ctx::buffer, DPI_COUNT, dpiset::enabled, dpiset::x, and dpiset::y.

Referenced by _cmd_get().

```
84                                                  {
85     (void)kb;
86
87     // Print all DPI settings
88     const int BUFFER_LEN = 100;
89     char* buffer = malloc(BUFFER_LEN);
90     int length = 0;
91     for(int i = 0; i < DPI_COUNT; i++){
92         // Print the stage number
93         int newlen = 0;
94         snprintf(buffer + length, BUFFER_LEN - length, length == 0 ? "%d%n" : " %d%n", i, &newlen);
95         length += newlen;
96         // Print the DPI settings
97         if(!(dpi->enabled & (1 << i)))
98             snprintf(buffer + length, BUFFER_LEN - length, ":off%n", &newlen);
99         else
100             snprintf(buffer + length, BUFFER_LEN - length, ":%u,%u%n", dpi->x[i], dpi->
      y[i], &newlen);
101         length += newlen;
102     }
103     return buffer;
104 }
```

Here is the caller graph for this function:



13.15.1.7 int savedpi (usbdevice * *kb,* dpiset * *dpi,* lighting * *light*)

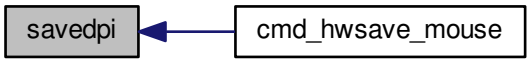
Definition at line 194 of file dpi.c.

References lighting::b, dpiset::current, DPI_COUNT, dpiset::enabled, lighting::g, LED_MOUSE, dpiset::lift, MSG_-SIZE, N_MOUSE_ZONES, lighting::r, dpiset::snap, usbsend, dpiset::x, and dpiset::y.

Referenced by cmd_hwsave_mouse().

```
194                                                  {
195     // Send X/Y DPIs
196     for(int i = 0; i < DPI_COUNT; i++){
197         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0xd0, 1 };
198         data_pkt[2] |= i;
199         *(ushort*)(data_pkt + 5) = dpi->x[i];
200         *(ushort*)(data_pkt + 7) = dpi->y[i];
201         // Save the RGB value for this setting too
202         data_pkt[9] = light->r[LED_MOUSE + N_MOUSE_ZONES + i];
203         data_pkt[10] = light->g[LED_MOUSE + N_MOUSE_ZONES + i];
204         data_pkt[11] = light->b[LED_MOUSE + N_MOUSE_ZONES + i];
205         if(!usbsend(kb, data_pkt, 1))
206             return -1;
207     }
208
209     // Send settings
210     uchar data_pkt[4][MSG_SIZE] = {
211         { 0x07, 0x13, 0x05, 1, dpi->enabled },
212         { 0x07, 0x13, 0x02, 1, dpi->current },
213         { 0x07, 0x13, 0x03, 1, dpi->lift },
214         { 0x07, 0x13, 0x04, 1, dpi->snap, 0x05 }
215     };
216     if(!usbsend(kb, data_pkt[0], 4))
217         return -2;
218     // Finished
219     return 0;
220 }
```

Here is the caller graph for this function:



13.15.1.8 int updatedpi (usbdevice * *kb*, int *force*)

Definition at line 106 of file dpi.c.

References usbdevice::active, dpiset::current, usbprofile::currentmode, usbmode::dpi, DPI_COUNT, dpiset::enabled, dpiset::forceupdate, usbprofile::lastdpi, dpiset::lift, MSG_SIZE, usbdevice::profile, dpiset::snap, usbsend, dpiset::x, and dpiset::y.

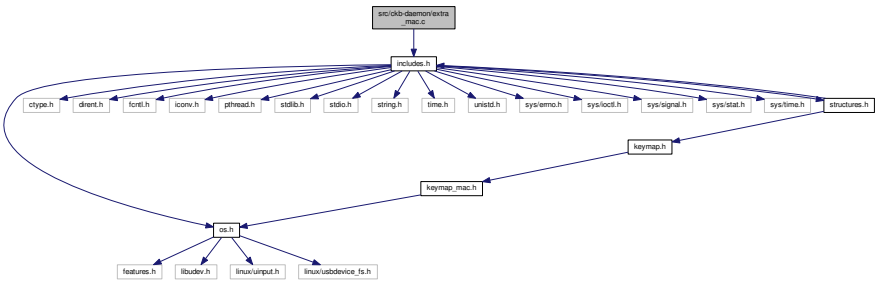
```
106                                     {
107     if(!kb->active)
108         return 0;
109     dpiset* lastdpi = &kb->profile->lastdpi;
110     dpiset* newdpi = &kb->profile->currentmode->dpi;
111     // Don't do anything if the settings haven't changed
112     if(!force && !lastdpi->forceupdate && !newdpi->forceupdate
113             && !memcmp(lastdpi, newdpi, sizeof(dpiset)))
114         return 0;
115     lastdpi->forceupdate = newdpi->forceupdate = 0;
116
117     if (newdpi->current != lastdpi->current) {
118         // Before we switch the current DPI stage, make sure the stage we are
119         // switching to is both enabled and configured to the correct DPI.
120
121         // Enable the stage if necessary.
122         if ((lastdpi->enabled & 1 << newdpi->current) == 0) {
123             uchar newenabled;
124             // If the new enabled flags contain both the current and previous
125             // stages, use it.
126             if (newdpi->enabled & 1 << newdpi->current &&
127                 newdpi->enabled & 1 << lastdpi->current) {
128                 newenabled = newdpi->enabled;
129             } else {
130                 // Otherwise just enable the new stage. We'll write the actual
131                 // requested flags after switching stages.
132                 newenabled = lastdpi->enabled | 1 << newdpi->current;
133             }
134             uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x05, 0, newenabled };
135             if(!usbsend(kb, data_pkt, 1))
136                 return -2;
137             // Cache the flags we wrote.
138             lastdpi->enabled = newenabled;
139         }
140         // Set the DPI for the new stage if necessary.
141         if (newdpi->x[newdpi->current] != lastdpi->x[newdpi->current] ||
142             newdpi->y[newdpi->current] != lastdpi->y[newdpi->current]) {
143             uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0xd0, 0 };
144             data_pkt[2] |= newdpi->current;
145             *(ushort*)(data_pkt + 5) = newdpi->x[newdpi->current];
146             *(ushort*)(data_pkt + 7) = newdpi->y[newdpi->current];
147             if(!usbsend(kb, data_pkt, 1))
148                 return -1;
149             // Set these values in the cache so we don't rewrite them.
150             lastdpi->x[newdpi->current] = newdpi->x[newdpi->current];
151             lastdpi->y[newdpi->current] = newdpi->y[newdpi->current];
152         }
153         // Set current DPI stage.
154         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x02, 0, newdpi->
    current };
155         if(!usbsend(kb, data_pkt, 1))
156             return -2;
157     }
158
159     // Send X/Y DPIs. We've changed to the new stage already so these can be set
```

```
160     // safely.
161     for(int i = 0; i < DPI_COUNT; i++){
162         if (newdpi->x[i] == lastdpi->x[i] && newdpi->y[i] == lastdpi->y[i])
163             continue;
164         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0xd0, 0 };
165         data_pkt[2] |= i;
166         *(ushort*)(data_pkt + 5) = newdpi->x[i];
167         *(ushort*)(data_pkt + 7) = newdpi->y[i];
168         if(!usbsend(kb, data_pkt, 1))
169             return -1;
170     }
171
172     // Send settings
173     if (newdpi->enabled != lastdpi->enabled) {
174         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x05, 0, newdpi->
    enabled };
175         if(!usbsend(kb, data_pkt, 1))
176             return -2;
177     }
178     if (newdpi->lift != lastdpi->lift) {
179         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x03, 0, newdpi->
    lift };
180         if(!usbsend(kb, data_pkt, 1))
181             return -2;
182     }
183     if (newdpi->snap != lastdpi->snap) {
184         uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x04, 0, newdpi->
    snap, 0x05 };
185         if(!usbsend(kb, data_pkt, 1))
186             return -2;
187     }
188
189     // Finished
190     memcpy(lastdpi, newdpi, sizeof(dpiset));
191     return 0;
192 }
```

13.16 src/ckb-daemon/extra_mac.c File Reference

```
#include "includes.h"
```

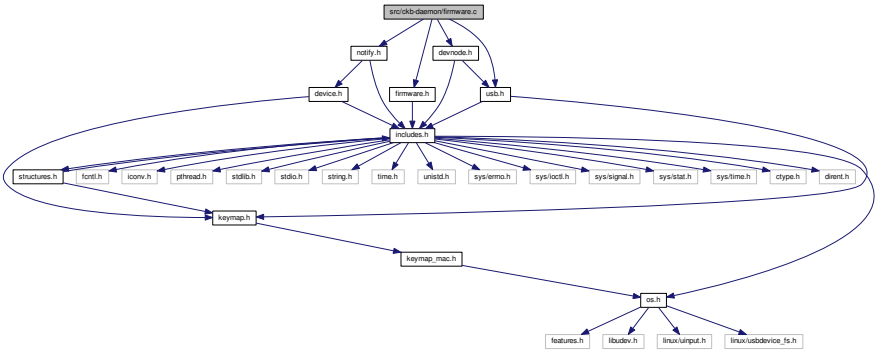
Include dependency graph for extra_mac.c:



13.17 src/ckb-daemon/firmware.c File Reference

```
#include "devnode.h"
#include "firmware.h"
#include "notify.h"
#include "usb.h"
```

Include dependency graph for firmware.c:



Macros

- #define FW_OK 0
- #define FW_NOFILE -1
- #define FW_WRONGDEV -2
- #define FW_USBFAIL -3
- #define FW_MAXSIZE (255 ∗ 256)

Functions

- int getfwversion (usbdevice ∗kb)
- int fwupdate (usbdevice ∗kb, const char ∗path, int nnumber)
- int cmd_fwupdate (usbdevice ∗kb, usbmode ∗dummy1, int nnumber, int dummy2, const char ∗path)

13.17.1 Macro Definition Documentation

13.17.1.1 #define FW_MAXSIZE (255 ∗ 256)

Definition at line 51 of file firmware.c.

Referenced by fwupdate().

13.17.1.2 #define FW_NOFILE -1

Definition at line 7 of file firmware.c.

Referenced by cmd_fwupdate(), and fwupdate().

13.17.1.3 #define FW_OK 0

Definition at line 6 of file firmware.c.

Referenced by cmd_fwupdate(), and fwupdate().

13.17.1.4 #define FW_USBFAIL -3

Definition at line 9 of file firmware.c.

Referenced by cmd_fwupdate(), and fwupdate().

13.17.1.5 #define FW_WRONGDEV -2

Definition at line 8 of file firmware.c.

Referenced by cmd_fwupdate(), and fwupdate().

13.17.2 Function Documentation

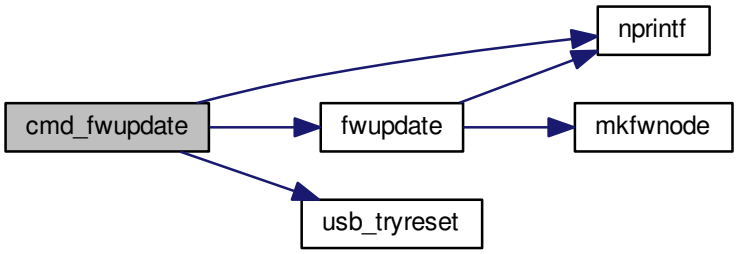
13.17.2.1 int cmd_fwupdate (usbdevice * kb, usbmode * dummy1, int nnumber, int dummy2, const char * path)

Definition at line 154 of file firmware.c.

References FEAT_FWUPDATE, FW_NOFILE, FW_OK, FW_USBFAIL, FW_WRONGDEV, fwupdate(), HAS_FEATURES, nprintf(), and usb_tryreset().

```
154                                                                                                 {
155     (void)dummy1;
156     (void)dummy2;
157
158     if(!HAS_FEATURES(kb, FEAT_FWUPDATE))
159         return 0;
160     // Update the firmware
161     int ret = fwupdate(kb, path, nnumber);
162     while(ret == FW_USBFAIL){
163         // Try to reset the device if it fails
164         if(usb_tryreset(kb))
165             break;
166         ret = fwupdate(kb, path, nnumber);
167     }
168     switch(ret){
169     case FW_OK:
170         nprintf(kb, nnumber, 0, "fwupdate %s ok\n", path);
171         break;
172     case FW_NOFILE:
173     case FW_WRONGDEV:
174         nprintf(kb, nnumber, 0, "fwupdate %s invalid\n", path);
175         break;
176     case FW_USBFAIL:
177         nprintf(kb, nnumber, 0, "fwupdate %s fail\n", path);
178         return -1;
179     }
180     return 0;
181 }
```

Here is the call graph for this function:



13.17.2.2 int fwupdate (usbdevice * kb, const char * path, int nnumber)

Definition at line 55 of file firmware.c.

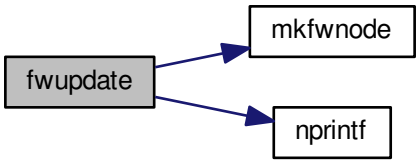
References ckb_err, ckb_info, FW_MAXSIZE, FW_NOFILE, FW_OK, FW_USBFAIL, FW_WRONGDEV, usbdevice::fwversion, mkfwnode(), MSG_SIZE, nprintf(), usbdevice::product, usbdevice::usbdelay, usbsend, and usbdevice::vendor.

Referenced by cmd_fwupdate().

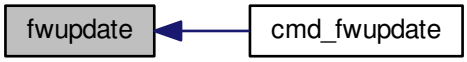
```
55                                                          {
56      // Read the firmware from the given path
57      char* fwdata = calloc(1, FW_MAXSIZE + 256);
58      int fd = open(path, O_RDONLY);
59      if(fd == -1){
60          ckb_err("Failed to open firmware file %s: %s\n", path, strerror(errno));
61          return FW_NOFILE;
62      }
63      ssize_t length = read(fd, fwdata, FW_MAXSIZE + 1);
64      if(length <= 0x108 || length > FW_MAXSIZE){
65          ckb_err("Failed to read firmware file %s: %s\n", path, length <= 0 ? strerror(errno) : "
    Wrong size");
66          close(fd);
67          return FW_NOFILE;
68      }
69      close(fd);
70
71      short vendor, product, version;
72      // Copy the vendor ID, product ID, and version from the firmware file
73      memcpy(&vendor, fwdata + 0x102, 2);
74      memcpy(&product, fwdata + 0x104, 2);
75      memcpy(&version, fwdata + 0x106, 2);
76      // Check against the actual device
77      if(vendor != kb->vendor || product != kb->product){
78          ckb_err("Firmware file %s doesn't match device (V: %04x P: %04x)\n", path, vendor, product);
79          return FW_WRONGDEV;
80      }
81      ckb_info("Loading firmware version %04x from %s\n", version, path);
82      nprintf(kb, nnumber, 0, "fwupdate %s 0/%d\n", path, (int)length);
83      // Force the device to 10ms delay (we need to deliver packets very slowly to make sure it doesn't get
    overwhelmed)
84      kb->usbdelay = 10;
85      // Send the firmware messages (256 bytes at a time)
86      uchar data_pkt[7][MSG_SIZE] = {
87          { 0x07, 0x0c, 0xf0, 0x01, 0 },
88          { 0x07, 0x0d, 0xf0, 0 },
89          { 0x7f, 0x01, 0x3c, 0 },
90          { 0x7f, 0x02, 0x3c, 0 },
91          { 0x7f, 0x03, 0x3c, 0 },
92          { 0x7f, 0x04, 0x3c, 0 },
93          { 0x7f, 0x05, 0x10, 0 }
94      };
95      int output = 0, last = 0;
96      int index = 0;
97      while(output < length){
98          int npackets = 1;
99          // Packet 1: data position
100         data_pkt[1][6] = index++;
101         while(output < length){
102             npackets++;
103             if(npackets != 6){
104                 // Packets 2-5: 60 bytes of data
105                 memcpy(data_pkt[npackets] + 4, fwdata + output, 60);
106                 last = output;
107                 output += 60;
108             } else {
109                 // Packet 6: 16 bytes
110                 memcpy(data_pkt[npackets] + 4, fwdata + output, 16);
111                 last = output;
112                 output += 16;
113                 break;
114             }
115         }
116         if(index == 1){
117             if(!usbsend(kb, data_pkt[0], 1)){
118                 ckb_err("Firmware update failed\n");
119                 return FW_USBFAIL;
120             }
121             // The above packet can take a lot longer to process, so wait for a while
122             sleep(3);
123             if(!usbsend(kb, data_pkt[2], npackets - 1)){
124                 ckb_err("Firmware update failed\n");
125                 return FW_USBFAIL;
126             }
127         } else {
128             // If the output ends here, set the length byte appropriately
129             if(output >= length)
130                 data_pkt[npackets][2] = length - last;
131             if(!usbsend(kb, data_pkt[1], npackets)){
132                 ckb_err("Firmware update failed\n");
```

```
133                 return FW_USBFAIL;
134             }
135         }
136         nprintf(kb, nnumber, 0, "fwupdate %s %d/%d\n", path, output, (int)length);
137     }
138     // Send the final pair of messages
139     uchar data_pkt2[2][MSG_SIZE] = {
140         { 0x07, 0x0d, 0xf0, 0x00, 0x00, 0x00, index },
141         { 0x07, 0x02, 0xf0, 0 }
142     };
143     if(!usbsend(kb, data_pkt2[0], 2)){
144         ckb_err("Firmware update failed\n");
145         return FW_USBFAIL;
146     }
147     // Updated successfully
148     kb->fwversion = version;
149     mkfwnode(kb);
150     ckb_info("Firmware update complete\n");
151     return FW_OK;
152 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.17.2.3 int getfwversion (usbdevice * *kb*)

Definition at line 11 of file firmware.c.

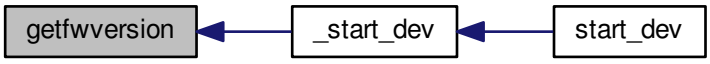
References ckb_err, ckb_warn, FEAT_POLLRATE, usbdevice::features, usbdevice::fwversion, MSG_SIZE, usbdevice::pollrate, usbdevice::product, usbrecv, and usbdevice::vendor.

Referenced by _start_dev().

```
11                              {
12     // Ask board for firmware info
13     uchar data_pkt[MSG_SIZE] = { 0x0e, 0x01, 0 };
14     uchar in_pkt[MSG_SIZE];
15     if(!usbrecv(kb, data_pkt, in_pkt))
16         return -1;
17     if(in_pkt[0] != 0x0e || in_pkt[1] != 0x01){
18         ckb_err("Bad input header\n");
```

```
19          return -1;
20      }
21      short vendor, product, version, bootloader;
22      // Copy the vendor ID, product ID, version, and poll rate from the firmware data
23      memcpy(&version, in_pkt + 8, 2);
24      memcpy(&bootloader, in_pkt + 10, 2);
25      memcpy(&vendor, in_pkt + 12, 2);
26      memcpy(&product, in_pkt + 14, 2);
27      char poll = in_pkt[16];
28      if(poll <= 0){
29          poll = -1;
30          kb->features &= ~FEAT_POLLRATE;
31      }
32      // Print a warning if the message didn't match the expected data
33      if(vendor != kb->vendor)
34          ckb_warn("Got vendor ID %04x (expected %04x)\n", vendor, kb->
    vendor);
35      if(product != kb->product)
36          ckb_warn("Got product ID %04x (expected %04x)\n", product, kb->
    product);
37      // Set firmware version and poll rate
38      if(version == 0 || bootloader == 0){
39          // Needs firmware update
40          kb->fwversion = 0;
41          kb->pollrate = -1;
42      } else {
43          if(version != kb->fwversion && kb->fwversion != 0)
44              ckb_warn("Got firmware version %04x (expected %04x)\n", version, kb->
    fwversion);
45          kb->fwversion = version;
46          kb->pollrate = poll;
47      }
48      return 0;
49  }
```

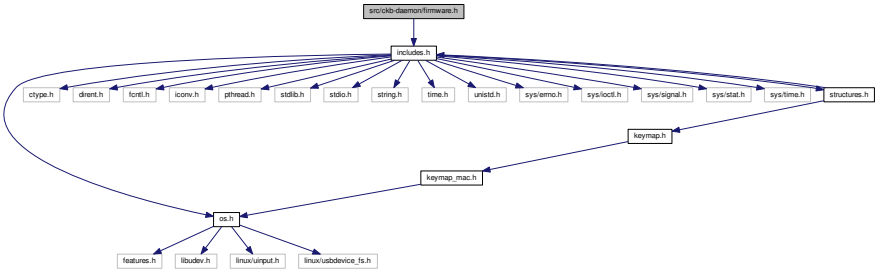
Here is the caller graph for this function:



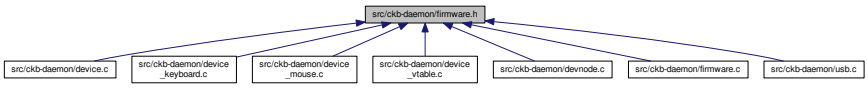
13.18 src/ckb-daemon/firmware.h File Reference

```
#include "includes.h"
```

Include dependency graph for firmware.h:



This graph shows which files directly or indirectly include this file:



Functions

- int getfwversion (usbdevice ∗kb)
- int cmd_fwupdate (usbdevice ∗kb, usbmode ∗dummy1, int nnumber, int dummy2, const char ∗path)

13.18.1 Function Documentation

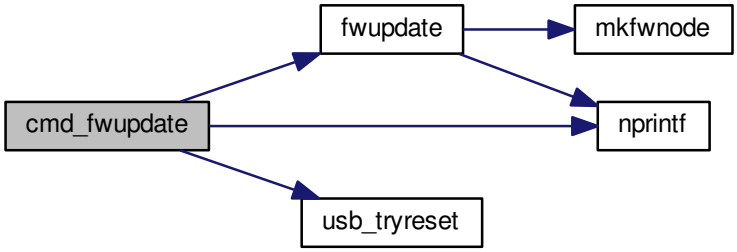
13.18.1.1 int cmd_fwupdate (usbdevice ∗ *kb,* usbmode ∗ *dummy1,* int *nnumber,* int *dummy2,* const char ∗ *path*)

Definition at line 154 of file firmware.c.

References FEAT_FWUPDATE, FW_NOFILE, FW_OK, FW_USBFAIL, FW_WRONGDEV, fwupdate(), HAS_FEATURES, nprintf(), and usb_tryreset().

```
154                                                                                 {
155     (void)dummy1;
156     (void)dummy2;
157
158     if(!HAS_FEATURES(kb, FEAT_FWUPDATE))
159         return 0;
160     // Update the firmware
161     int ret = fwupdate(kb, path, nnumber);
162     while(ret == FW_USBFAIL){
163         // Try to reset the device if it fails
164         if(usb_tryreset(kb))
165             break;
166         ret = fwupdate(kb, path, nnumber);
167     }
168     switch(ret){
169     case FW_OK:
170         nprintf(kb, nnumber, 0, "fwupdate %s ok\n", path);
171         break;
172     case FW_NOFILE:
173     case FW_WRONGDEV:
174         nprintf(kb, nnumber, 0, "fwupdate %s invalid\n", path);
175         break;
176     case FW_USBFAIL:
177         nprintf(kb, nnumber, 0, "fwupdate %s fail\n", path);
178         return -1;
179     }
180     return 0;
181 }
```

Here is the call graph for this function:



13.18.1.2 int getfwversion (usbdevice * *kb*)

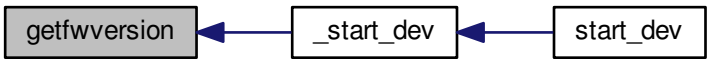
Definition at line 11 of file firmware.c.

References ckb_err, ckb_warn, FEAT_POLLRATE, usbdevice::features, usbdevice::fwversion, MSG_SIZE, usbdevice::pollrate, usbdevice::product, usbrecv, and usbdevice::vendor.

Referenced by _start_dev().

```
11                              {
12      // Ask board for firmware info
13      uchar data_pkt[MSG_SIZE] = { 0x0e, 0x01, 0 };
14      uchar in_pkt[MSG_SIZE];
15      if(!usbrecv(kb, data_pkt, in_pkt))
16          return -1;
17      if(in_pkt[0] != 0x0e || in_pkt[1] != 0x01){
18          ckb_err("Bad input header\n");
19          return -1;
20      }
21      short vendor, product, version, bootloader;
22      // Copy the vendor ID, product ID, version, and poll rate from the firmware data
23      memcpy(&version, in_pkt + 8, 2);
24      memcpy(&bootloader, in_pkt + 10, 2);
25      memcpy(&vendor, in_pkt + 12, 2);
26      memcpy(&product, in_pkt + 14, 2);
27      char poll = in_pkt[16];
28      if(poll <= 0){
29          poll = -1;
30          kb->features &= ~FEAT_POLLRATE;
31      }
32      // Print a warning if the message didn't match the expected data
33      if(vendor != kb->vendor)
34          ckb_warn("Got vendor ID %04x (expected %04x)\n", vendor, kb->
    vendor);
35      if(product != kb->product)
36          ckb_warn("Got product ID %04x (expected %04x)\n", product, kb->
    product);
37      // Set firmware version and poll rate
38      if(version == 0 || bootloader == 0){
39          // Needs firmware update
40          kb->fwversion = 0;
41          kb->pollrate = -1;
42      } else {
43          if(version != kb->fwversion && kb->fwversion != 0)
44              ckb_warn("Got firmware version %04x (expected %04x)\n", version, kb->
    fwversion);
45          kb->fwversion = version;
46          kb->pollrate = poll;
47      }
48      return 0;
49  }
```

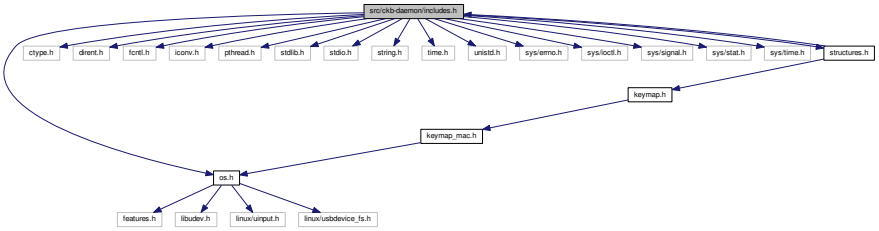
Here is the caller graph for this function:



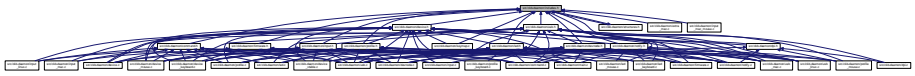
13.19 src/ckb-daemon/includes.h File Reference

```
#include "os.h"
#include <ctype.h>
#include <dirent.h>
#include <fcntl.h>
#include <iconv.h>
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <sys/errno.h>
#include <sys/ioctl.h>
#include <sys/signal.h>
#include <sys/stat.h>
#include <sys/time.h>
#include "structures.h"
```

Include dependency graph for includes.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define INDEX_OF(entry, array) (int)(entry - array)
- #define ckb_s_out stdout

- #define ckb_s_err stdout
- #define __FILE_NOPATH__ (strrchr(__FILE__, '/') ? strrchr(__FILE__, '/') + 1 : __FILE__)
- #define ckb_fatal_nofile(fmt, args...) fprintf(ckb_s_err, "[F] " fmt, ## args)
- #define ckb_fatal_fn(fmt, file, line, args...) fprintf(ckb_s_err, "[F] %s (via %s:%d): " fmt, __func__, file, line, ## args)
- #define ckb_fatal(fmt, args...) fprintf(ckb_s_err, "[F] %s (%s:%d): " fmt, __func__, __FILE_NOPATH__, __LINE__, ## args)
- #define ckb_err_nofile(fmt, args...) fprintf(ckb_s_err, "[E] " fmt, ## args)
- #define ckb_err_fn(fmt, file, line, args...) fprintf(ckb_s_err, "[E] %s (via %s:%d): " fmt, __func__, file, line, ## args)
- #define ckb_err(fmt, args...) fprintf(ckb_s_err, "[E] %s (%s:%d): " fmt, __func__, __FILE_NOPATH__, __LINE__, ## args)
- #define ckb_warn_nofile(fmt, args...) fprintf(ckb_s_out, "[W] " fmt, ## args)
- #define ckb_warn_fn(fmt, file, line, args...) fprintf(ckb_s_out, "[W] %s (via %s:%d): " fmt, __func__, file, line, ## args)
- #define ckb_warn(fmt, args...) fprintf(ckb_s_out, "[W] %s (%s:%d): " fmt, __func__, __FILE_NOPATH__, __LINE__, ## args)
- #define ckb_info_nofile(fmt, args...) fprintf(ckb_s_out, "[I] " fmt, ## args)
- #define ckb_info_fn(fmt, file, line, args...) fprintf(ckb_s_out, "[I] " fmt, ## args)
- #define ckb_info(fmt, args...) fprintf(ckb_s_out, "[I] " fmt, ## args)
- #define timespec_gt(left, right) ((left).tv_sec > (right).tv_sec || ((left).tv_sec == (right).tv_sec && (left).tv_nsec > (right).tv_nsec))
- #define timespec_eq(left, right) ((left).tv_sec == (right).tv_sec && (left).tv_nsec == (right).tv_nsec)
- #define timespec_ge(left, right) ((left).tv_sec > (right).tv_sec || ((left).tv_sec == (right).tv_sec && (left).tv_nsec >= (right).tv_nsec))
- #define timespec_lt(left, right) (!timespec_ge(left, right))
- #define timespec_le(left, right) (!timespec_gt(left, right))

Typedefs

- typedef unsigned char uchar
- typedef unsigned short ushort

Functions

- void timespec_add (struct timespec *timespec, long nanoseconds)

13.19.1 Macro Definition Documentation

13.19.1.1 #define __FILE_NOPATH__ (strrchr(__FILE__, '/') ? strrchr(__FILE__, '/') + 1 : __FILE__)

Definition at line 40 of file includes.h.

13.19.1.2 #define ckb_err(*fmt, args...*) fprintf(ckb_s_err, "[E] %s (%s:%d): " fmt, __func__, __FILE_NOPATH__, __LINE__, ## args)

Definition at line 49 of file includes.h.

Referenced by _mkdevpath(), fwupdate(), getfwversion(), loaddpi(), loadrgb_kb(), loadrgb_mouse(), macro_pt_dequeue(), os_inputmain(), os_sendindicators(), os_setupusb(), restart(), setupusb(), uinputopen(), usb_tryreset(), usbadd(), and usbclaim().

13.19.1.3 #define ckb_err_fn(*fmt, file, line, args...*) fprintf(ckb_s_err, "[E] %s (via %s:%d): " fmt, __func__, file, line, ## args)

Definition at line 48 of file includes.h.

Referenced by _nk95cmd(), _usbrecv(), os_usbrecv(), and os_usbsend().

13.19.1.4 #define ckb_err_nofile(*fmt, args...*) fprintf(ckb_s_err, "[E] " fmt, ## args)

Definition at line 47 of file includes.h.

13.19.1.5 #define ckb_fatal(*fmt, args...*) fprintf(ckb_s_err, "[F] %s (%s:%d): " fmt, __func__, __FILE_NOPATH__, __LINE__, ## args)

Definition at line 46 of file includes.h.

Referenced by usbmain().

13.19.1.6 #define ckb_fatal_fn(*fmt, file, line, args...*) fprintf(ckb_s_err, "[F] %s (via %s:%d): " fmt, __func__, file, line, ## args)

Definition at line 45 of file includes.h.

13.19.1.7 #define ckb_fatal_nofile(*fmt, args...*) fprintf(ckb_s_err, "[F] " fmt, ## args)

Definition at line 44 of file includes.h.

Referenced by main().

13.19.1.8 #define ckb_info(*fmt, args...*) fprintf(ckb_s_out, "[I] " fmt, ## args)

Definition at line 55 of file includes.h.

Referenced by _setupusb(), _start_dev(), closeusb(), cmd_restart(), fwupdate(), os_inputmain(), os_setupusb(), rmdevpath(), usb_tryreset(), usbadd(), and usbclaim().

13.19.1.9 #define ckb_info_fn(*fmt, file, line, args...*) fprintf(ckb_s_out, "[I] " fmt, ## args)

Definition at line 54 of file includes.h.

13.19.1.10 #define ckb_info_nofile(*fmt, args...*) fprintf(ckb_s_out, "[I] " fmt, ## args)

Definition at line 53 of file includes.h.

Referenced by main().

13.19.1.11 #define ckb_s_err stdout

Definition at line 36 of file includes.h.

13.19.1.12 #define ckb_s_out stdout

Definition at line 35 of file includes.h.

13.19.1.13 #define ckb_warn(*fmt, args...*) fprintf(ckb_s_out, "[W] %s (%s:%d): " fmt, __func__, __FILE_NOPATH__, __LINE__, ## args)

Definition at line 52 of file includes.h.

Referenced by _mkdevpath(), _mknotifynode(), _start_dev(), _updateconnected(), getfwversion(), hid_kb_translate(), isync(), mkfwnode(), os_inputclose(), os_keypress(), os_mousemove(), readlines(), rmdevpath(), uinputopen(), and usbmain().

13.19.1.14 #define ckb_warn_fn(*fmt, file, line, args...*) fprintf(ckb_s_out, "[W] %s (via %s:%d): " fmt, __func__, file, line, ## args)

Definition at line 51 of file includes.h.

Referenced by os_usbrecv(), and os_usbsend().

13.19.1.15 #define ckb_warn_nofile(*fmt, args...*) fprintf(ckb_s_out, "[W] " fmt, ## args)

Definition at line 50 of file includes.h.

Referenced by main().

13.19.1.16 #define INDEX_OF(*entry, array*) (int)(entry - array)

Definition at line 27 of file includes.h.

Referenced by _mkdevpath(), _mknotifynode(), _rmnotifynode(), _setupusb(), closeusb(), mkfwnode(), nprintf(), os_closeusb(), os_inputmain(), os_inputopen(), os_setupusb(), readcmd(), and rmdevpath().

13.19.1.17 #define timespec_eq(left, right) ((left).tv_sec == (right).tv_sec && (left).tv_nsec == (right).tv_nsec)

Definition at line 60 of file includes.h.

13.19.1.18 #define timespec_ge(left, right) ((left).tv_sec > (right).tv_sec || ((left).tv_sec == (right).tv_sec && (left).tv_nsec >= (right).tv_nsec))

Definition at line 61 of file includes.h.

13.19.1.19 #define timespec_gt(left, right) ((left).tv_sec > (right).tv_sec || ((left).tv_sec == (right).tv_sec && (left).tv_nsec > (right).tv_nsec))

Definition at line 59 of file includes.h.

13.19.1.20 #define timespec_le(left, right) (!timespec_gt(left, right))

Definition at line 63 of file includes.h.

13.19.1.21 #define timespec_lt(left, right) (!timespec_ge(left, right))

Definition at line 62 of file includes.h.

13.19.2 Typedef Documentation

13.19.2.1 typedef unsigned char uchar

Definition at line 24 of file includes.h.

13.19.2.2 typedef unsigned short ushort

Definition at line 25 of file includes.h.

13.19.3 Function Documentation

13.19.3.1 void timespec_add (struct timespec * *timespec,* long *nanoseconds*)

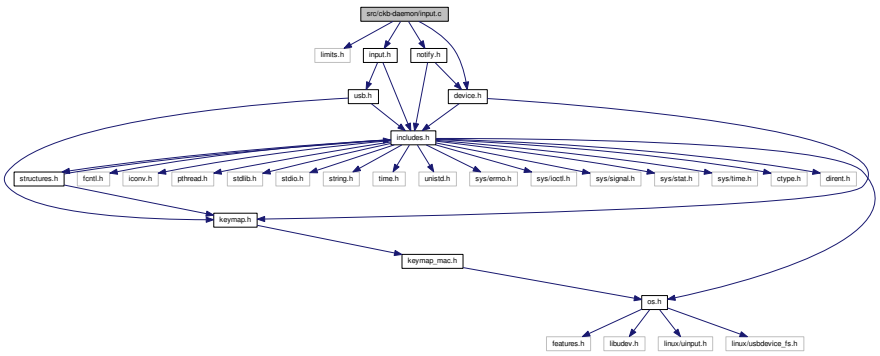
Definition at line 19 of file main.c.

```
19                                                              {
20     nanoseconds += timespec->tv_nsec;
21     timespec->tv_sec += nanoseconds / 1000000000;
22     timespec->tv_nsec = nanoseconds % 1000000000;
23 }
```

13.20 src/ckb-daemon/input.c File Reference

```
#include <limits.h>
#include "device.h"
#include "input.h"
#include "notify.h"
```

Include dependency graph for input.c:



Macros

- #define IS_WHEEL(scan, kb) (((scan) == KEY_VOLUMEUP || (scan) == KEY_VOLUMEDOWN || (scan) == BTN_WHEELUP || (scan) == BTN_WHEELDOWN) && (!IS_K65(kb) && !IS_K63(kb)))

Functions

- int macromask (const uchar *key1, const uchar *key2)
- static void macro_pt_enqueue ()

  macro_pt_enqueue Save the new thread in the single linked list (FIFO).

- static pthread_t macro_pt_dequeue ()

  macro_pt_dequeue gets the first thread id of the list and returns the thread_id stored in it.
- static pthread_t macro_pt_first ()

  macro_pt_first returns the first pthread_id but does not remove the first entry.
- static void * play_macro (void *param)

  play_macro is the code for all threads started to play a macro.
- static void inputupdate_keys (usbdevice *kb)

  inputupdate_keys Handle input from Keyboard or mouse; start Macrof if detected.
- void inputupdate (usbdevice *kb)
- void updateindicators_kb (usbdevice *kb, int force)
- void initbind (binding *bind)
- void freebind (binding *bind)
- void cmd_bind (usbdevice *kb, usbmode *mode, int dummy, int keyindex, const char *to)
- void cmd_unbind (usbdevice *kb, usbmode *mode, int dummy, int keyindex, const char *to)
- void cmd_rebind (usbdevice *kb, usbmode *mode, int dummy, int keyindex, const char *to)
- static void _cmd_macro (usbmode *mode, const char *keys, const char *assignment)
- void cmd_macro (usbdevice *kb, usbmode *mode, const int notifynumber, const char *keys, const char *assignment)

Variables

- static ptlist_t * pt_head = 0

  pt_head is the head pointer for the single linked thread list managed by macro_pt_en/dequeue().
- static ptlist_t * pt_tail = 0

  pt_tail is the tail pointer for the single linked thread list managed by macro_pt_en/dequeue().

13.20.1 Macro Definition Documentation

13.20.1.1 #define IS_WHEEL(*scan, kb*) (((scan) == KEY_VOLUMEUP || (scan) == KEY_VOLUMEDOWN || (scan) == BTN_WHEELUP || (scan) == BTN_WHEELDOWN) && (!IS_K65(kb) && !IS_K63(kb)))

Referenced by inputupdate_keys().

13.20.2 Function Documentation

13.20.2.1 static void _cmd_macro (usbmode * *mode,* const char * *keys,* const char * *assignment*) [static]

Definition at line 353 of file input.c.

References keymacro::actioncount, keymacro::actions, usbmode::bind, keymacro::combo, macroaction::delay, macroaction::down, keymap, left, MACRO_MAX, binding::macrocap, binding::macrocount, binding::macros, N_KEYBYTES_INPUT, N_KEYS_INPUT, right(), macroaction::scan, key::scan, and SET_KEYBIT.

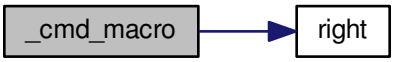
Referenced by cmd_macro().

```
353                                                                         {
354     binding* bind = &mode->bind;
355     if(!keys && !assignment){
356         // Null strings = "macro clear" -> erase the whole thing
357         for(int i = 0; i < bind->macrocount; i++)
358             free(bind->macros[i].actions);
359         bind->macrocount = 0;
360         return;
361     }
362     if(bind->macrocount >= MACRO_MAX)
363         return;
364     // Create a key macro
365     keymacro macro;
366     memset(&macro, 0, sizeof(macro));
```

```
367     // Scan the left side for key names, separated by +
368     int empty = 1;
369     int left = strlen(keys), right = strlen(assignment);
370     int position = 0, field = 0;
371     char keyname[24];
372     while(position < left && sscanf(keys + position, "%10[^+]%n", keyname, &field) == 1){
373         int keycode;
374         if((sscanf(keyname, "#%d", &keycode) && keycode >= 0 && keycode <
    N_KEYS_INPUT)
375                 || (sscanf(keyname, "#x%x", &keycode) && keycode >= 0 && keycode <
    N_KEYS_INPUT)){
376             // Set a key numerically
377             SET_KEYBIT(macro.combo, keycode);
378             empty = 0;
379         } else {
380             // Find this key in the keymap
381             for(unsigned i = 0; i < N_KEYS_INPUT; i++){
382                 if(keymap[i].name && !strcmp(keyname, keymap[i].name)){
383                     macro.combo[i / 8] |= 1 << (i % 8);
384                     empty = 0;
385                     break;
386                 }
387             }
388         }
389         if(keys[position += field] == '+')
390             position++;
391     }
392     if(empty)
393         return;
394     // Count the number of actions (comma separated)
395     int count = 1;
396     for(const char* c = assignment; *c != 0; c++){
397         if(*c == ',')
398             count++;
399     }
400     // Allocate a buffer for them
401     macro.actions = calloc(count, sizeof(macroaction));
402     macro.actioncount = 0;
403     // Scan the actions
404     position = 0;
405     field = 0;
406     // max action = old 11 chars plus 12 chars which is the max 32-bit int 4294967295 size
407     while(position < right && sscanf(assignment + position, "%23[^,]%n", keyname, &field) == 1){
408         if(!strcmp(keyname, "clear"))
409             break;
410 
411         // Check for local key delay of the form '[+-]<key>=<delay>'
412         long int long_delay;    // scanned delay value, used to keep delay in range.
413         unsigned int delay = UINT_MAX; // computed delay value. UINT_MAX means use global delay value.
414         char real_keyname[12];  // temp to hold the left side (key) of the <key>=<delay>
415         int scan_matches = sscanf(keyname, "%11[^=]=%ld", real_keyname, &long_delay);
416         if (scan_matches == 2) {
417             if (0 <= long_delay && long_delay < UINT_MAX) {
418                 delay = (unsigned int)long_delay;
419                 strcpy(keyname, real_keyname); // keyname[24], real_keyname[12]
420             }
421         }
422 
423         int down = (keyname[0] == '+');
424         if(down || keyname[0] == '-'){
425             int keycode;
426             if((sscanf(keyname + 1, "#%d", &keycode) && keycode >= 0 && keycode < N_KEYS_INPUT)
427                     || (sscanf(keyname + 1, "#x%x", &keycode) && keycode >= 0 && keycode <
    N_KEYS_INPUT)){
428                 // Set a key numerically
429                 macro.actions[macro.actioncount].scan =
    keymap[keycode].scan;
430                 macro.actions[macro.actioncount].down = down;
431                 macro.actions[macro.actioncount].delay = delay;
432                 macro.actioncount++;
433             } else {
434                 // Find this key in the keymap
435                 for(unsigned i = 0; i < N_KEYS_INPUT; i++){
436                     if(keymap[i].name && !strcmp(keyname + 1, keymap[i].name)){
437                         macro.actions[macro.actioncount].scan =
    keymap[i].scan;
438                         macro.actions[macro.actioncount].down = down;
439                         macro.actions[macro.actioncount].delay = delay;
440                         macro.actioncount++;
441                         break;
442                     }
443                 }
444             }
445         }
446         if(assignment[position += field] == ',')
447             position++;
448     }
```

```
449
450     // See if there's already a macro with this trigger
451     keymacro* macros = bind->macros;
452     for(int i = 0; i < bind->macrocount; i++){
453         if(!memcmp(macros[i].combo, macro.combo, N_KEYBYTES_INPUT)){
454             free(macros[i].actions);
455             // If the new macro has no actions, erase the existing one
456             if(!macro.actioncount){
457                 for(int j = i + 1; j < bind->macrocount; j++)
458                     memcpy(macros + j - 1, macros + j, sizeof(keymacro));
459                 bind->macrocount--;
460             } else
461                 // If there are actions, replace the existing with the new
462                 memcpy(macros + i, &macro, sizeof(keymacro));
463             return;
464         }
465     }
466
467     // Add the macro to the device settings if not empty
468     if(macro.actioncount < 1)
469         return;
470     memcpy(bind->macros + (bind->macrocount++), &macro, sizeof(
    keymacro));
471     if(bind->macrocount >= bind->macrocap)
472         bind->macros = realloc(bind->macros, (bind->macrocap += 16) * sizeof(
    keymacro));
473 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.20.2.2 void cmd_bind (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *to*)

Definition at line 307 of file input.c.

References binding::base, usbmode::bind, imutex, keymap, N_KEYS_INPUT, and key::scan.

```
307                                                                         {
308     (void)dummy;
309
310     if(keyindex >= N_KEYS_INPUT)
311         return;
312     // Find the key to bind to
313     int tocode = 0;
314     if(sscanf(to, "#x%ux", &tocode) != 1 && sscanf(to, "#%u", &tocode) == 1 && tocode <
    N_KEYS_INPUT){
```

```
315         pthread_mutex_lock(imutex(kb));
316         mode->bind.base[keyindex] = tocode;
317         pthread_mutex_unlock(imutex(kb));
318         return;
319     }
320     // If not numeric, look it up
321     for(int i = 0; i < N_KEYS_INPUT; i++){
322         if(keymap[i].name && !strcmp(to, keymap[i].name)){
323             pthread_mutex_lock(imutex(kb));
324             mode->bind.base[keyindex] = keymap[i].scan;
325             pthread_mutex_unlock(imutex(kb));
326             return;
327         }
328     }
329 }
```

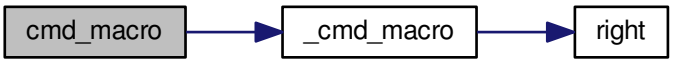
13.20.2.3 void cmd_macro (usbdevice * *kb*, usbmode * *mode*, const int *notifynumber*, const char * *keys*, const char * *assignment*)

Definition at line 475 of file input.c.

References _cmd_macro(), and imutex.

```
475
         {
476     (void)notifynumber;
477
478     pthread_mutex_lock(imutex(kb));
479     _cmd_macro(mode, keys, assignment);
480     pthread_mutex_unlock(imutex(kb));
481 }
```

Here is the call graph for this function:



13.20.2.4 void cmd_rebind (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *to*)

Definition at line 342 of file input.c.

References binding::base, usbmode::bind, imutex, keymap, N_KEYS_INPUT, and key::scan.

```
342                                                                                 {
343     (void)dummy;
344     (void)to;
345
346     if(keyindex >= N_KEYS_INPUT)
347         return;
348     pthread_mutex_lock(imutex(kb));
349     mode->bind.base[keyindex] = keymap[keyindex].scan;
350     pthread_mutex_unlock(imutex(kb));
351 }
```

13.20.2.5 void cmd_unbind (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *to*)

Definition at line 331 of file input.c.

References binding::base, usbmode::bind, imutex, KEY_UNBOUND, and N_KEYS_INPUT.

```
331                                                                                 {
332     (void)dummy;
333     (void)to;
334
335     if(keyindex >= N_KEYS_INPUT)
336         return;
337     pthread_mutex_lock(imutex(kb));
338     mode->bind.base[keyindex] = KEY_UNBOUND;
339     pthread_mutex_unlock(imutex(kb));
340 }
```

13.20.2.6 void freebind (binding * *bind*)

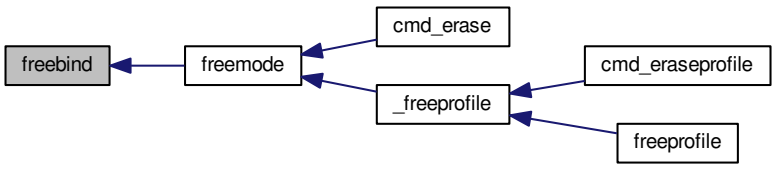
Definition at line 300 of file input.c.

References keymacro::actions, binding::macrocount, and binding::macros.

Referenced by freemode().

```
300                             {
301     for(int i = 0; i < bind->macrocount; i++)
302         free(bind->macros[i].actions);
303     free(bind->macros);
304     memset(bind, 0, sizeof(*bind));
305 }
```

Here is the caller graph for this function:



13.20.2.7 void initbind (binding * *bind*)

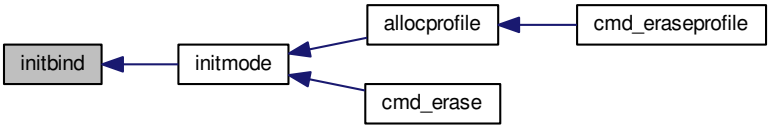
Definition at line 292 of file input.c.

References binding::base, keymap, binding::macrocap, binding::macrocount, binding::macros, N_KEYS_INPUT, and key::scan.

Referenced by initmode().

```
292                             {
293     for(int i = 0; i < N_KEYS_INPUT; i++)
294         bind->base[i] = keymap[i].scan;
295     bind->macros = calloc(32, sizeof(keymacro));
296     bind->macrocap = 32;
297     bind->macrocount = 0;
298 }
```

Here is the caller graph for this function:



13.20.2.8 void inputupdate (usbdevice * kb)

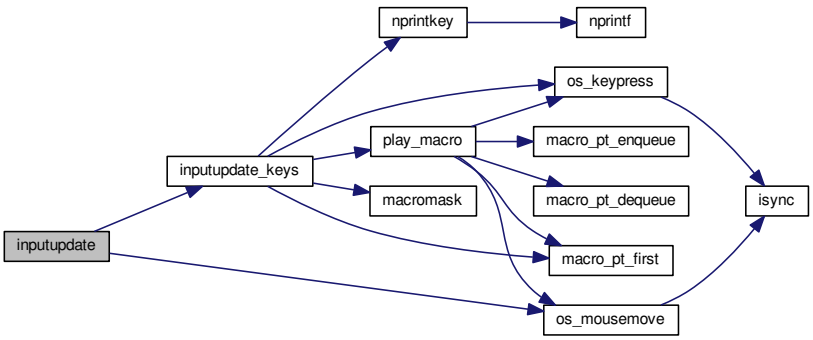
Definition at line 241 of file input.c.

References usbdevice::input, inputupdate_keys(), os_mousemove(), usbdevice::profile, usbinput::rel_x, usbinput::rel_y, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

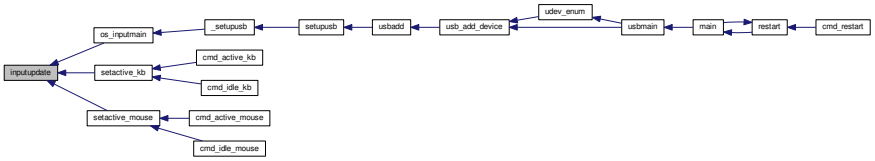
Referenced by os_inputmain(), setactive_kb(), and setactive_mouse().

```
241                                {
242 #ifdef OS_LINUX
243     if((!kb->uinput_kb || !kb->uinput_mouse)
244 #else
245     if(!kb->event
246 #endif
247             || !kb->profile)
248         return;
249     // Process key/button input
250     inputupdate_keys(kb);
251     // Process mouse movement
252     usbinput* input = &kb->input;
253     if(input->rel_x != 0 || input->rel_y != 0){
254         os_mousemove(kb, input->rel_x, input->rel_y);
255         input->rel_x = input->rel_y = 0;
256     }
257     // Finish up
258     memcpy(input->prevkeys, input->keys, N_KEYBYTES_INPUT);
259 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.20.2.9 static void inputupdate_keys (usbdevice * *kb*) [static]

Parameters

kb	

Process all queued keypresses if no macro is running yet.

Todo If we want to get all keys typed while a macro is played, add the code for it here.

Definition at line 134 of file input.c.

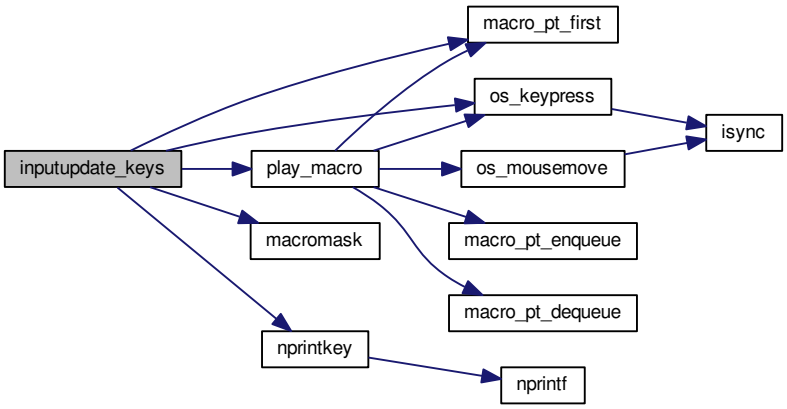
References usbdevice::active, binding::base, usbmode::bind, keymacro::combo, usbprofile::currentmode, usbdevice::input, IS_MOD, IS_WHEEL, parameter::kb, keymap, usbinput::keys, parameter::macro, macro_pt_first(), binding::macrocount, macromask(), binding::macros, N_KEYBYTES_INPUT, N_KEYS_INPUT, usbmode::notify, nprintkey(), os_keypress(), OUTFIFO_MAX, play_macro(), usbinput::prevkeys, usbdevice::profile, key::scan, SCAN_SILENT, and keymacro::triggered.

Referenced by inputupdate().

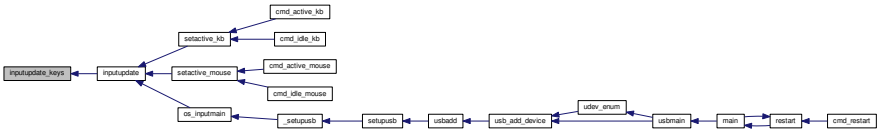
```
134                                             {
135     usbmode* mode = kb->profile->currentmode;
136     binding* bind = &mode->bind;
137     usbinput* input = &kb->input;
138 
139     // Don't do anything if the state hasn't changed
140     if(!memcmp(input->prevkeys, input->keys, N_KEYBYTES_INPUT))
141         return;
142     // Look for macros matching the current state
143     if (kb->active) {
144         for (int i = 0; i < bind->macrocount; i++) {
145             keymacro* macro = &bind->macros[i];
146             if (macromask(input->keys, macro->combo)) {
147                 if (!macro->triggered) {
148                     parameter_t* params = malloc(sizeof(parameter_t));
149                     if (params == 0) {
150                         perror("inputupdate_keys got no more mem:");
151                     } else {
152                         pthread_t thread = 0;
153                         params->kb = kb;
154                         params->macro = macro;
155                         int retval = pthread_create(&thread, 0, play_macro, (void*)params);
156                         if (retval) {
157                             perror("inputupdate_keys: Creating thread returned not null");
158                         } else {
159                             macro->triggered = 1;
160                         }
161                     }
162                 }
163             } else macro->triggered = 0;
164         }
165     }
166     // Make a list of keycodes to send. Rearrange them so that modifier keydowns always come first
167     // and modifier keyups always come last. This ensures that shortcut keys will register properly
168     // even if both keydown events happen at once.
169     // N_KEYS + 4 is used because the volume wheel generates keydowns and keyups at the same time
170     // (it's currently impossible to press all four at once, but safety first)
171     int events[N_KEYS_INPUT + 4];
172     int modcount = 0, keycount = 0, rmodcount = 0;
173     for(int byte = 0; byte < N_KEYBYTES_INPUT; byte++){
174         char oldb = input->prevkeys[byte], newb = input->keys[byte];
175         if(oldb == newb)
```

```
176             continue;
177         for(int bit = 0; bit < 8; bit++){
178             int keyindex = byte * 8 + bit;
179             if(keyindex >= N_KEYS_INPUT)
180                 break;
181             const key* map = keymap + keyindex;
182             int scancode = (kb->active) ? bind->base[keyindex] : map->
    scan;
183             char mask = 1 << bit;
184             char old = oldb & mask, new = newb & mask;
185             // If the key state changed, send it to the input device
186             if(old != new){
187                 // Don't echo a key press if there's no scancode associated
188                 if(!(scancode & SCAN_SILENT)){
189                     if(IS_MOD(scancode)){
190                         if(new){
191                             // Modifier down: Add to the end of modifier keys
192                             for(int i = keycount + rmodcount; i > 0; i--)
193                                 events[modcount + i] = events[modcount + i - 1];
194                             // Add 1 to the scancode because A is zero on OSX
195                             // Positive code = keydown, negative code = keyup
196                             events[modcount++] = scancode + 1;
197                         } else {
198                             // Modifier up: Add to the end of everything
199                             events[modcount + keycount + rmodcount++] = -(scancode + 1);
200                         }
201                     } else {
202                         // Regular keypress: add to the end of regular keys
203                         for(int i = rmodcount; i > 0; i--)
204                             events[modcount + keycount + i] = events[modcount + keycount + i - 1];
205                         events[modcount + keycount++] = new ? (scancode + 1) : -(scancode + 1);
206                         // The volume wheel and the mouse wheel don't generate keyups, so create them
    automatically
207 #define IS_WHEEL(scan, kb)  (((scan) == KEY_VOLUMEUP || (scan) == KEY_VOLUMEDOWN || (scan) == BTN_WHEELUP
    || (scan) == BTN_WHEELDOWN) && (!IS_K65(kb) && !IS_K63(kb)))
208                         if(new && IS_WHEEL(map->scan, kb)){
209                             for(int i = rmodcount; i > 0; i--)
210                                 events[modcount + keycount + i] = events[modcount + keycount + i - 1];
211                             events[modcount + keycount++] = -(scancode + 1);
212                             input->keys[byte] &= ~mask;
213                         }
214                     }
215                 }
216                 // Print notifications if desired
217                 if(kb->active){
218                     for(int notify = 0; notify < OUTFIFO_MAX; notify++){
219                         if(mode->notify[notify][byte] & mask){
220                             nprintkey(kb, notify, keyindex, new);
221                             // Wheels doesn't generate keyups
222                             if(new && IS_WHEEL(map->scan, kb))
223                                 nprintkey(kb, notify, keyindex, 0);
224                         }
225                     }
226                 }
227             }
228         }
229     }
232     if (!macro_pt_first()) {
233         int totalkeys = modcount + keycount + rmodcount;
234         for(int i = 0; i < totalkeys; i++){
235             int scancode = events[i];
236             os_keypress(kb, (scancode < 0 ? -scancode : scancode) - 1, scancode > 0);
237         }
238     }
239 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.20.2.10 static pthread_t macro_pt_dequeue ( ) [static]

Returns

the ptread_id of the first element. If list is empty, return 0.

Attention

Becuase multiple threads may use this function in parallel, save the critical section with a mutex.

< why are we called?

< Was last element in the list, so clear tail.

< save the return value before deleting element

Definition at line 49 of file input.c.

References ckb_err, ptlist::next, pt_head, and ptlist::thread_id.

Referenced by play_macro().

```
49                                      {
50     pthread_t retval = 0;
51     ptlist_t* elem = 0;
52     if (pt_head == 0 && pt_tail == 0) {
53         ckb_err("macro_pt_dequeue: called on empty list.\n");
54         return 0;
55     }
56     elem = pt_head;
57     pt_head = pt_head->next;
```

```
58      if (pt_head == 0) pt_tail = 0;
59      retval = elem->thread_id;
60      free(elem);
61      return retval;
62  }
```

Here is the caller graph for this function:

13.20.2.11 static void macro_pt_enqueue ( ) [static]

Attention

Becuase multiple threads may use this function in parallel, save the critical section with a mutex.

< exit on critical situation;

Todo find a better exit strategy if no more mem available.

< The element knows its ID byself

< new list, first element

< existing list, append on last element (FIFO)

Definition at line 27 of file input.c.

References ptlist::next, and ptlist::thread_id.

Referenced by play_macro().

```
27                                  {
28      ptlist_t* new_elem = malloc(sizeof(ptlist_t));
29      if (!new_elem) {
30          perror("macro_pt_enqueue: ");
31          exit (-1);
32      }
33      new_elem->next = 0;
34      new_elem->thread_id = pthread_self();
35      if (pt_head == 0) {
36          pt_head = pt_tail = new_elem;
37      } else {
38          pt_tail->next = new_elem;
39          pt_tail = new_elem;
40      }
41  }
```

Here is the caller graph for this function:

13.20.2.12 static pthread_t macro_pt_first ( ) [static]

Returns

the pthread_id of the first element in the list or 0 if list is empty.

Attention

Becuase multiple threads may use this function in parallel, save the critical section with a mutex (avoid NIL-ptr)

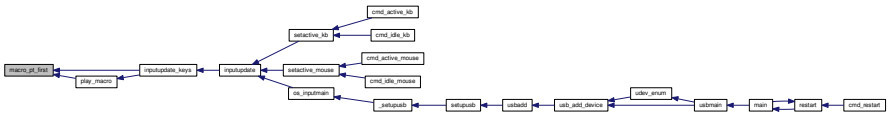
Definition at line 70 of file input.c.

References ptlist::thread_id.

Referenced by inputupdate_keys(), and play_macro().

```
70                                      {
71     return pt_head? pt_head->thread_id : 0;
72 }
```

Here is the caller graph for this function:



13.20.2.13 int macromask (const uchar * *key1*, const uchar * *key2*)

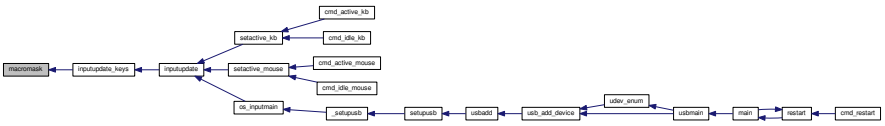
Definition at line 6 of file input.c.

References N_KEYBYTES_INPUT.

Referenced by inputupdate_keys().

```
6                                                       {
7     // Scan a macro against key input. Return 0 if any of them don't match
8     for(int i = 0; i < N_KEYBYTES_INPUT; i++){
9         // if((key1[i] & key2[i]) != key2[i])
10         if(key1[i] != key2[i])  // Changed to detect G-keys + modifiers
11             return 0;
12     }
13     return 1;
14 }
```

Here is the caller graph for this function:



13.20.2.14 static void* play_macro (void * *param*) [static]

Parameters

param	*parameter_t* to store Kb-ptr and macro-ptr (thread may get only one user-parameter)

Returns

0 on success, -1 else (no one is interested in it except the kernel...)

First have a look if we are the first and only macro-thread to run. If not, wait. So enqueue our thread first, so it is remembered for us and can be seen by all others.

< If the first thread in the list is not our, another one is running

< Give all new threads the chance to enter the block.

Send events for each keypress in the macro

< Synchonization between macro output and color information

< use this unlock / relock for enablling the parallel running colorization

< local delay set

< use default global delay

< use delays depending on macro length

< protect the linked list and the mvar

< Wake up all waiting threads

< for the linked list and the mvar

< Sync keyboard input/output and colorization

Definition at line 79 of file input.c.

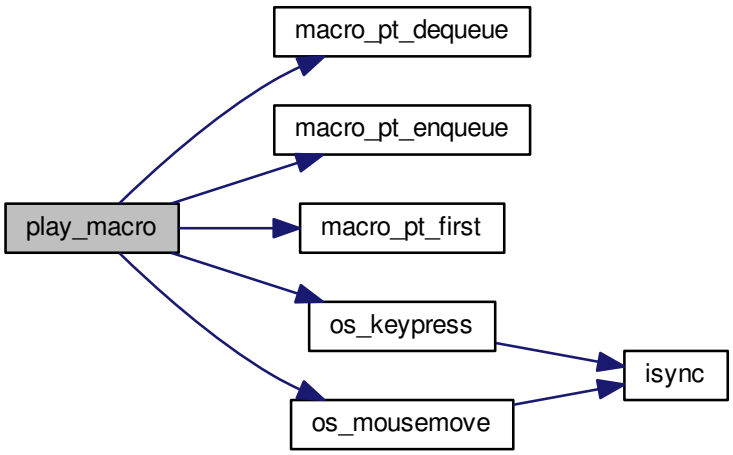
References keymacro::actioncount, keymacro::actions, macroaction::delay, usbdevice::delay, macroaction::down, parameter::kb, parameter::macro, macro_pt_dequeue(), macro_pt_enqueue(), macro_pt_first(), mmutex, mmutex2, mvar, os_keypress(), os_mousemove(), macroaction::rel_x, macroaction::rel_y, and macroaction::scan.

Referenced by inputupdate_keys().

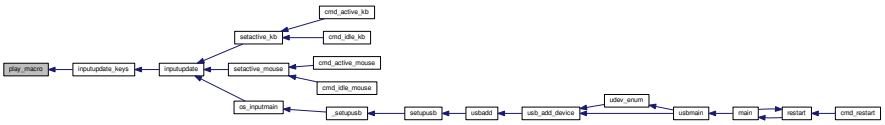
```
79                                      {
80     parameter_t* ptr = (parameter_t*) param;
81     usbdevice* kb = ptr->kb;
82     keymacro* macro = ptr->macro;
83 
86     pthread_mutex_lock(mmutex2(kb));
87     macro_pt_enqueue();
88     // ckb_info("Entering critical section with 0x%lx. Queue head is 0x%lx\n",  (unsigned long
       int)pthread_self(), (unsigned long int)macro_pt_first());
89     while (macro_pt_first() != pthread_self()) {
90         // ckb_info("Now waiting with 0x%lx because of 0x%lx\n", (unsigned long int)pthread_self(),
       (unsigned long int)macro_pt_first());
91         pthread_cond_wait(mvar(kb), mmutex2(kb));
92         // ckb_info("Waking up with 0x%lx\n", (unsigned long int)pthread_self());
93     }
94     pthread_mutex_unlock(mmutex2(kb));
95 
97     pthread_mutex_lock(mmutex(kb));
98     for (int a = 0; a < macro->actioncount; a++) {
99         macroaction* action = macro->actions + a;
100         if (action->rel_x != 0 || action->rel_y != 0)
101             os_mousemove(kb, action->rel_x, action->rel_y);
102         else {
103             os_keypress(kb, action->scan, action->down);
104             pthread_mutex_unlock(mmutex(kb));
105             if (action->delay != UINT_MAX && action->delay) {
106                 clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = action->
       delay * 1000}, NULL);
107             } else if (kb->delay != UINT_MAX && kb->delay) {
108                 clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = kb->
       delay * 1000}, NULL);
109             } else if (a < (macro->actioncount - 1)) {
110                 if (a > 200) {
111                     clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = action->
       delay * 100000}, NULL);
```

```
112                 } else if (a > 20) {
113                     clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = 30000}, NULL);
114                 }
115             }
116             pthread_mutex_lock(mmutex(kb));
117         }
118     }
119
120     pthread_mutex_lock(mmutex2(kb));
121     // ckb_info("Now leaving 0x%lx and waking up all others\n", (unsigned long int)pthread_self());
122     macro_pt_dequeue();
123     pthread_cond_broadcast(mvar(kb));
124     pthread_mutex_unlock(mmutex2(kb));
125
126     pthread_mutex_unlock(mmutex(kb));
127     return 0;
128 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.20.2.15 void updateindicators_kb (usbdevice * *kb*, int *force*)

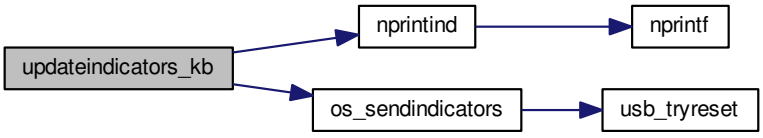
Definition at line 261 of file input.c.

References usbdevice::active, usbprofile::currentmode, DELAY_SHORT, usbdevice::hw_ileds, usbdevice::hw_ileds_old, I_CAPS, I_NUM, I_SCROLL, usbdevice::ileds, usbmode::inotify, usbmode::ioff, usbmode::ion, nprintind(), os_sendindicators(), OUTFIFO_MAX, and usbdevice::profile.

```
261                                                 {
262     // Read current hardware indicator state (set externally)
263     uchar old = kb->ileds, hw_old = kb->hw_ileds_old;
```

```
264     uchar new = kb->hw_ileds, hw_new = new;
265     // Update them if needed
266     if(kb->active){
267         usbmode* mode = kb->profile->currentmode;
268         new = (new & ~mode->ioff) | mode->ion;
269     }
270     kb->ileds = new;
271     kb->hw_ileds_old = hw_new;
272     if(old != new || force){
273         DELAY_SHORT(kb);
274         os_sendindicators(kb);
275     }
276     // Print notifications if desired
277     if(!kb->active)
278         return;
279     usbmode* mode = kb->profile->currentmode;
280     uchar indicators[] = { I_NUM, I_CAPS, I_SCROLL };
281     for(unsigned i = 0; i < sizeof(indicators) / sizeof(uchar); i++){
282         uchar mask = indicators[i];
283         if((hw_old & mask) == (hw_new & mask))
284             continue;
285         for(int notify = 0; notify < OUTFIFO_MAX; notify++){
286             if(mode->inotify[notify] & mask)
287                 nprintind(kb, notify, mask, hw_new & mask);
288         }
289     }
290 }
```

Here is the call graph for this function:



13.20.3 Variable Documentation

13.20.3.1 ptlist_t* pt_head = 0 [static]

Definition at line 18 of file input.c.

Referenced by macro_pt_dequeue().

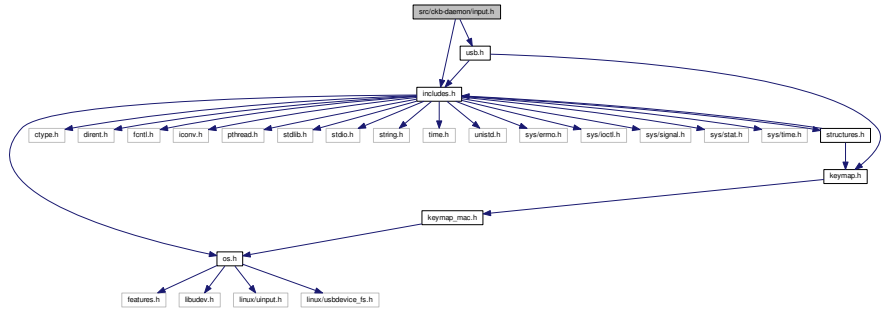
13.20.3.2 ptlist_t* pt_tail = 0 [static]

Definition at line 20 of file input.c.

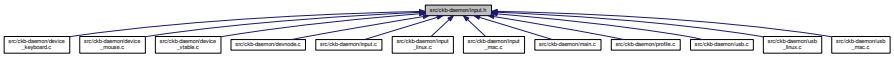
13.21 src/ckb-daemon/input.h File Reference

```
#include "includes.h"
#include "usb.h"
```

Include dependency graph for input.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct parameter

  struct parameter contains the values for a fresh started macro_play thread. parameter_t is the typedef for it. More...
- struct ptlist

  struct ptlist is one element in the single linked list to store macro_play threads waiting for their execution ptlist_t is the typedef for it. More...

Macros

- #define IS_MOD(s) ((s) == KEY_CAPSLOCK || (s) == KEY_NUMLOCK || (s) == KEY_SCROLLLOCK || (s) == KEY_LEFTSHIFT || (s) == KEY_RIGHTSHIFT || (s) == KEY_LEFTCTRL || (s) == KEY_RIGHTCTRL || (s) == KEY_LEFTMETA || (s) == KEY_RIGHTMETA || (s) == KEY_LEFTALT || (s) == KEY_RIGHTALT || (s) == KEY_FN)

Typedefs

- typedef struct parameter parameter_t

  struct parameter contains the values for a fresh started macro_play thread. parameter_t is the typedef for it.
- typedef struct ptlist ptlist_t

  struct ptlist is one element in the single linked list to store macro_play threads waiting for their execution ptlist_t is the typedef for it.

Functions

- int os_inputopen (usbdevice *kb)

  os_inputopen
- void os_inputclose (usbdevice *kb)
- void inputupdate (usbdevice *kb)
- void updateindicators_kb (usbdevice *kb, int force)
- void initbind (binding *bind)

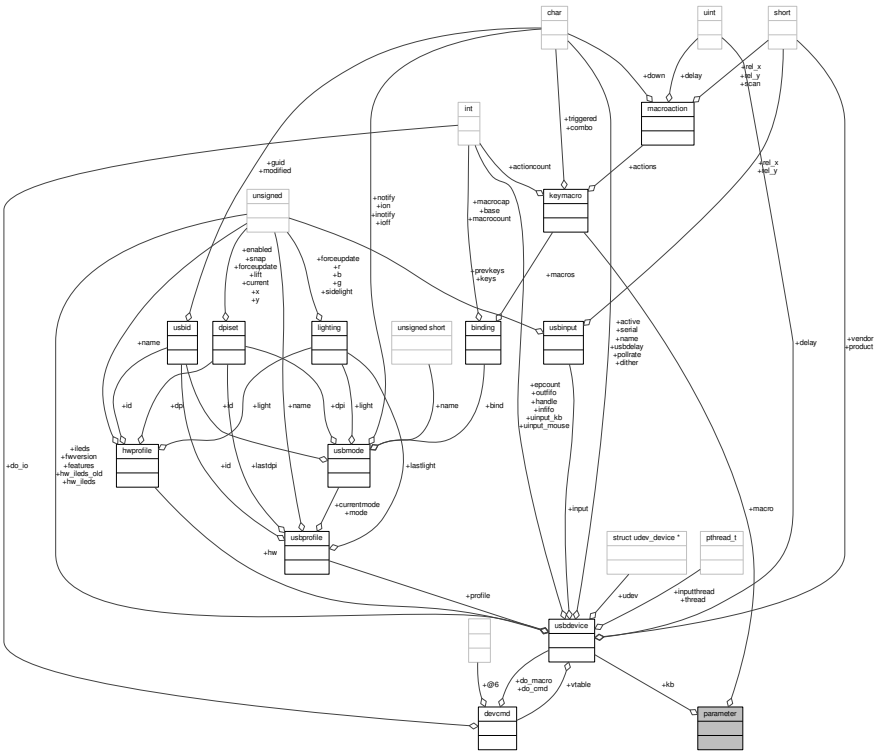
- void freebind (binding ∗bind)
- void cmd_bind (usbdevice ∗kb, usbmode ∗mode, int dummy, int keyindex, const char ∗to)
- void cmd_unbind (usbdevice ∗kb, usbmode ∗mode, int dummy, int keyindex, const char ∗ignored)
- void cmd_rebind (usbdevice ∗kb, usbmode ∗mode, int dummy, int keyindex, const char ∗ignored)
- void cmd_macro (usbdevice ∗kb, usbmode ∗mode, const int notifynumber, const char ∗keys, const char ∗assignment)
- void os_keypress (usbdevice ∗kb, int scancode, int down)
- void os_mousemove (usbdevice ∗kb, int x, int y)
- int os_setupindicators (usbdevice ∗kb)

13.21.1 Data Structure Documentation

13.21.1.1 struct parameter

Definition at line 54 of file input.h.

Collaboration diagram for parameter:



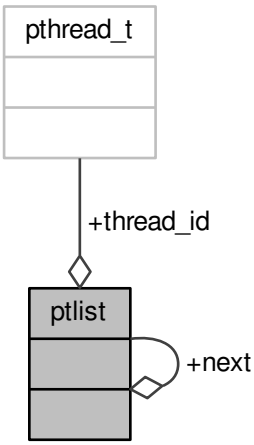
Data Fields

usbdevice ∗	kb	
keymacro ∗	macro	

13.21.1.2 struct ptlist

Definition at line 62 of file input.h.

Collaboration diagram for ptlist:



Data Fields

struct ptlist *	next	
pthread_t	thread_id	

13.21.2 Macro Definition Documentation

13.21.2.1 #define IS_MOD(*s*) ((s) == KEY_CAPSLOCK || (s) == KEY_NUMLOCK || (s) == KEY_SCROLLLOCK || (s) == KEY_LEFTSHIFT || (s) == KEY_RIGHTSHIFT || (s) == KEY_LEFTCTRL || (s) == KEY_RIGHTCTRL || (s) == KEY_LEFTMETA || (s) == KEY_RIGHTMETA || (s) == KEY_LEFTALT || (s) == KEY_RIGHTALT || (s) == KEY_FN)

Definition at line 34 of file input.h.

Referenced by inputupdate_keys().

13.21.3 Typedef Documentation

13.21.3.1 typedef struct parameter parameter_t

13.21.3.2 typedef struct ptlist ptlist_t

13.21.4 Function Documentation

13.21.4.1 void cmd_bind (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *to*)

Definition at line 307 of file input.c.

References binding::base, usbmode::bind, imutex, keymap, N_KEYS_INPUT, and key::scan.

```
307                                                                         {
308     (void)dummy;
309
310     if(keyindex >= N_KEYS_INPUT)
311         return;
```

```
312     // Find the key to bind to
313     int tocode = 0;
314     if(sscanf(to, "#x%ux", &tocode) != 1 && sscanf(to, "#%u", &tocode) == 1 && tocode <
      N_KEYS_INPUT){
315         pthread_mutex_lock(imutex(kb));
316         mode->bind.base[keyindex] = tocode;
317         pthread_mutex_unlock(imutex(kb));
318         return;
319     }
320     // If not numeric, look it up
321     for(int i = 0; i < N_KEYS_INPUT; i++){
322         if(keymap[i].name && !strcmp(to, keymap[i].name)){
323             pthread_mutex_lock(imutex(kb));
324             mode->bind.base[keyindex] = keymap[i].scan;
325             pthread_mutex_unlock(imutex(kb));
326             return;
327         }
328     }
329 }
```

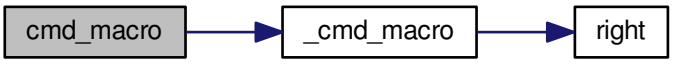
13.21.4.2 void cmd_macro (usbdevice * *kb*, usbmode * *mode*, const int *notifynumber*, const char * *keys*, const char * *assignment*)

Definition at line 475 of file input.c.

References _cmd_macro(), and imutex.

```
475
         {
476     (void)notifynumber;
477
478     pthread_mutex_lock(imutex(kb));
479     _cmd_macro(mode, keys, assignment);
480     pthread_mutex_unlock(imutex(kb));
481 }
```

Here is the call graph for this function:



13.21.4.3 void cmd_rebind (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *ignored*)

Definition at line 342 of file input.c.

References binding::base, usbmode::bind, imutex, keymap, N_KEYS_INPUT, and key::scan.

```
342                                                                                 {
343     (void)dummy;
344     (void)to;
345
346     if(keyindex >= N_KEYS_INPUT)
347         return;
348     pthread_mutex_lock(imutex(kb));
349     mode->bind.base[keyindex] = keymap[keyindex].scan;
350     pthread_mutex_unlock(imutex(kb));
351 }
```

13.21.4.4 void cmd_unbind (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *ignored*)

Definition at line 331 of file input.c.

References binding::base, usbmode::bind, imutex, KEY_UNBOUND, and N_KEYS_INPUT.

```
331                                                                                                 {
332     (void)dummy;
333     (void)to;
334
335     if(keyindex >= N_KEYS_INPUT)
336         return;
337     pthread_mutex_lock(imutex(kb));
338     mode->bind.base[keyindex] = KEY_UNBOUND;
339     pthread_mutex_unlock(imutex(kb));
340 }
```

13.21.4.5 void freebind (binding * *bind*)

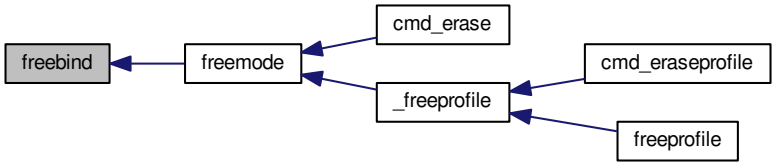
Definition at line 300 of file input.c.

References keymacro::actions, binding::macrocount, and binding::macros.

Referenced by freemode().

```
300                             {
301     for(int i = 0; i < bind->macrocount; i++)
302         free(bind->macros[i].actions);
303     free(bind->macros);
304     memset(bind, 0, sizeof(*bind));
305 }
```

Here is the caller graph for this function:



13.21.4.6 void initbind (binding * *bind*)

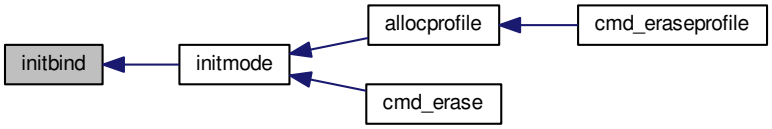
Definition at line 292 of file input.c.

References binding::base, keymap, binding::macrocap, binding::macrocount, binding::macros, N_KEYS_INPUT, and key::scan.

Referenced by initmode().

```
292                             {
293     for(int i = 0; i < N_KEYS_INPUT; i++)
294         bind->base[i] = keymap[i].scan;
295     bind->macros = calloc(32, sizeof(keymacro));
296     bind->macrocap = 32;
297     bind->macrocount = 0;
298 }
```

Here is the caller graph for this function:



13.21.4.7 void inputupdate (usbdevice * *kb*)

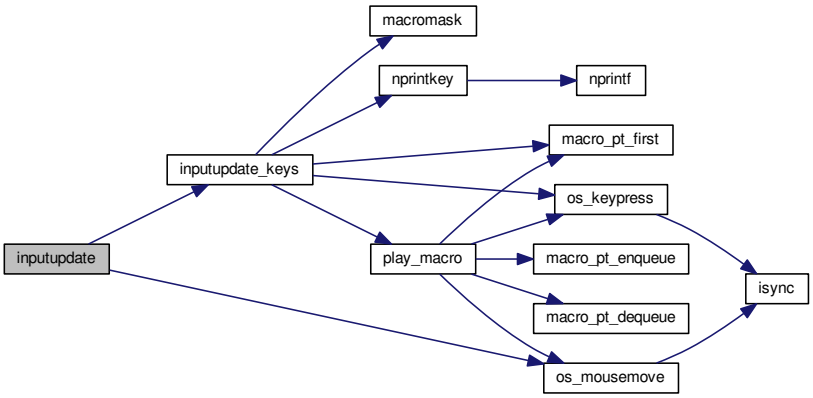
Definition at line 241 of file input.c.

References usbdevice::input, inputupdate_keys(), os_mousemove(), usbdevice::profile, usbinput::rel_x, usbinput::rel_y, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

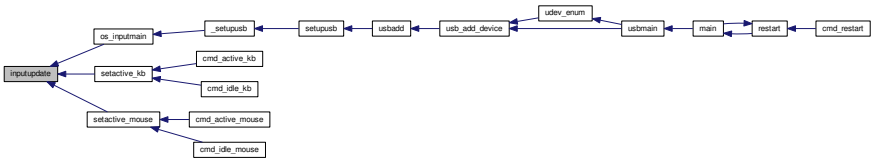
Referenced by os_inputmain(), setactive_kb(), and setactive_mouse().

```
241                                {
242 #ifdef OS_LINUX
243     if((!kb->uinput_kb || !kb->uinput_mouse)
244 #else
245     if(!kb->event
246 #endif
247             || !kb->profile)
248         return;
249     // Process key/button input
250     inputupdate_keys(kb);
251     // Process mouse movement
252     usbinput* input = &kb->input;
253     if(input->rel_x != 0 || input->rel_y != 0){
254         os_mousemove(kb, input->rel_x, input->rel_y);
255         input->rel_x = input->rel_y = 0;
256     }
257     // Finish up
258     memcpy(input->prevkeys, input->keys, N_KEYBYTES_INPUT);
259 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.21.4.8 void os_inputclose (usbdevice * kb)

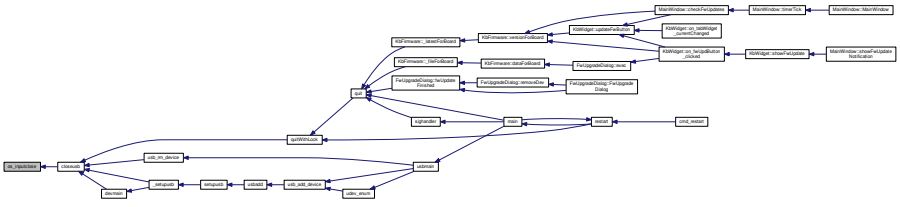
Definition at line 76 of file input_linux.c.

References ckb_warn, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

Referenced by closeusb().

```
76                                  {
77     if(kb->uinput_kb <= 0 || kb->uinput_mouse <= 0)
78         return;
79     // Set all keys released
80     struct input_event event;
81     memset(&event, 0, sizeof(event));
82     event.type = EV_KEY;
83     for(int key = 0; key < KEY_CNT; key++){
84         event.code = key;
85         if(write(kb->uinput_kb - 1, &event, sizeof(event)) <= 0)
86             ckb_warn("uinput write failed: %s\n", strerror(errno));
87         if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
88             ckb_warn("uinput write failed: %s\n", strerror(errno));
89     }
90     event.type = EV_SYN;
91     event.code = SYN_REPORT;
92     if(write(kb->uinput_kb - 1, &event, sizeof(event)) <= 0)
93         ckb_warn("uinput write failed: %s\n", strerror(errno));
94     if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
95         ckb_warn("uinput write failed: %s\n", strerror(errno));
96     // Close the keyboard
97     ioctl(kb->uinput_kb - 1, UI_DEV_DESTROY);
98     close(kb->uinput_kb - 1);
99     kb->uinput_kb = 0;
100     // Close the mouse
101     ioctl(kb->uinput_mouse - 1, UI_DEV_DESTROY);
102     close(kb->uinput_mouse - 1);
103     kb->uinput_mouse = 0;
104 }
```

Here is the caller graph for this function:



13.21.4.9 int os_inputopen (usbdevice * kb)

Parameters

kb	

Returns

Some tips on using `uinput_user_dev in`

Definition at line 55 of file input_linux.c.

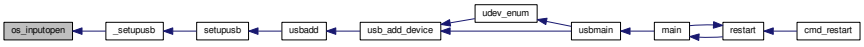
References usbdevice::fwversion, INDEX_OF, keyboard, usbdevice::name, usbdevice::product, usbdevice::uinput_kb, usbdevice::uinput_mouse, uinputopen(), and usbdevice::vendor.

Referenced by _setupusb().

```
55                                {
56     // Create the new input device
57     int index = INDEX_OF(kb, keyboard);
58     struct uinput_user_dev indev;
59     memset(&indev, 0, sizeof(indev));
60     snprintf(indev.name, UINPUT_MAX_NAME_SIZE, "ckb%d: %s", index, kb->name);
61     indev.id.bustype = BUS_USB;
62     indev.id.vendor = kb->vendor;
63     indev.id.product = kb->product;
64     indev.id.version = kb->fwversion;
65     // Open keyboard
66     int fd = uinputopen(&indev, 0);
67     kb->uinput_kb = fd;
68     if(fd <= 0)
69         return 0;
70     // Open mouse
71     fd = uinputopen(&indev, 1);
72     kb->uinput_mouse = fd;
73     return fd <= 0;
74 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.21.4.10 void os_keypress (usbdevice * *kb*, int *scancode*, int *down*)

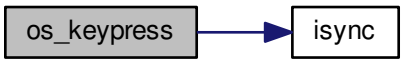
Definition at line 118 of file input_linux.c.

References BTN_WHEELDOWN, BTN_WHEELUP, ckb_warn, isync(), SCAN_MOUSE, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

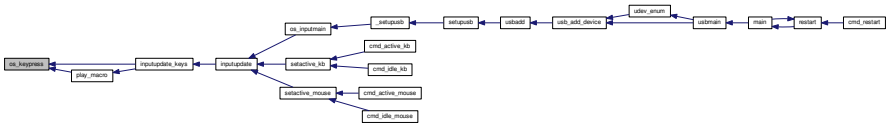
Referenced by inputupdate_keys(), and play_macro().

```
118                                                 {
119     struct input_event event;
120     memset(&event, 0, sizeof(event));
121     int is_mouse = 0;
122     if(scancode == BTN_WHEELUP || scancode == BTN_WHEELDOWN){
123         // The mouse wheel is a relative axis
124         if(!down)
125             return;
126         event.type = EV_REL;
127         event.code = REL_WHEEL;
128         event.value = (scancode == BTN_WHEELUP ? 1 : -1);
129         is_mouse = 1;
130     } else {
131         // Mouse buttons and key events are both EV_KEY. The scancodes are already correct, just remove the
    ckb bit
132         event.type = EV_KEY;
133         event.code = scancode & ~SCAN_MOUSE;
134         event.value = down;
135         is_mouse = !!(scancode & SCAN_MOUSE);
136     }
137     if(write((is_mouse ? kb->uinput_mouse : kb->uinput_kb) - 1, &event, sizeof(event))
    <= 0)
138         ckb_warn("uinput write failed: %s\n", strerror(errno));
139     else
140         isync(kb);
141 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.21.4.11 void os_mousemove (usbdevice * *kb*, int *x*, int *y*)

Definition at line 143 of file input_linux.c.

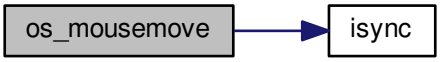
References ckb_warn, isync(), usbdevice::uinput_mouse, x, and y.

Referenced by inputupdate(), and play_macro().

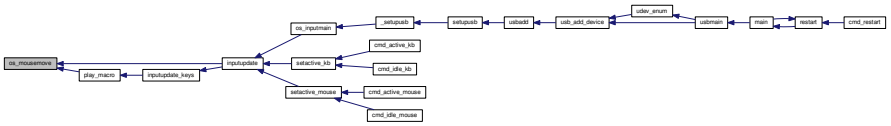
```
143                                                 {
144     struct input_event event;
145     memset(&event, 0, sizeof(event));
146     event.type = EV_REL;
147     if(x != 0){
148         event.code = REL_X;
149         event.value = x;
150         if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
151             ckb_warn("uinput write failed: %s\n", strerror(errno));
152         else
153             isync(kb);
154     }
155     if(y != 0){
```

```
156         event.code = REL_Y;
157         event.value = y;
158         if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
159             ckb_warn("uinput write failed: %s\n", strerror(errno));
160         else
161             isync(kb);
162     }
163 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.21.4.12 int os_setupindicators (usbdevice * *kb*)

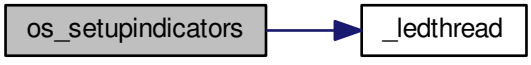
Definition at line 189 of file input_linux.c.

References _ledthread(), usbdevice::hw_ileds, usbdevice::hw_ileds_old, and usbdevice::ileds.

Referenced by _setupusb().

```
189                                     {
190     // Initialize LEDs to all off
191     kb->hw_ileds = kb->hw_ileds_old = kb->ileds = 0;
192     // Create and detach thread to read LED events
193     pthread_t thread;
194     int err = pthread_create(&thread, 0, _ledthread, kb);
195     if(err != 0)
196         return err;
197     pthread_detach(thread);
198     return 0;
199 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

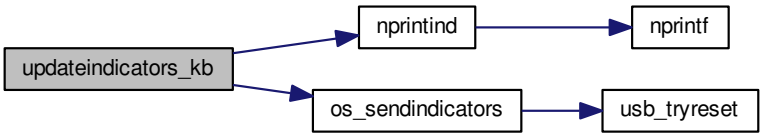
13.21.4.13 void updateindicators_kb (usbdevice * *kb*, int *force*)

Definition at line 261 of file input.c.

References usbdevice::active, usbprofile::currentmode, DELAY_SHORT, usbdevice::hw_ileds, usbdevice::hw_ileds_old, I_CAPS, I_NUM, I_SCROLL, usbdevice::ileds, usbmode::inotify, usbmode::ioff, usbmode::ion, nprintind(), os_sendindicators(), OUTFIFO_MAX, and usbdevice::profile.

```
261                                                 {
262     // Read current hardware indicator state (set externally)
263     uchar old = kb->ileds, hw_old = kb->hw_ileds_old;
264     uchar new = kb->hw_ileds, hw_new = new;
265     // Update them if needed
266     if(kb->active){
267         usbmode* mode = kb->profile->currentmode;
268         new = (new & ~mode->ioff) | mode->ion;
269     }
270     kb->ileds = new;
271     kb->hw_ileds_old = hw_new;
272     if(old != new || force){
273         DELAY_SHORT(kb);
274         os_sendindicators(kb);
275     }
276     // Print notifications if desired
277     if(!kb->active)
278         return;
279     usbmode* mode = kb->profile->currentmode;
280     uchar indicators[] = { I_NUM, I_CAPS, I_SCROLL };
281     for(unsigned i = 0; i < sizeof(indicators) / sizeof(uchar); i++){
282         uchar mask = indicators[i];
283         if((hw_old & mask) == (hw_new & mask))
284             continue;
285         for(int notify = 0; notify < OUTFIFO_MAX; notify++){
286             if(mode->inotify[notify] & mask)
287                 nprintind(kb, notify, mask, hw_new & mask);
288         }
289     }
290 }
```

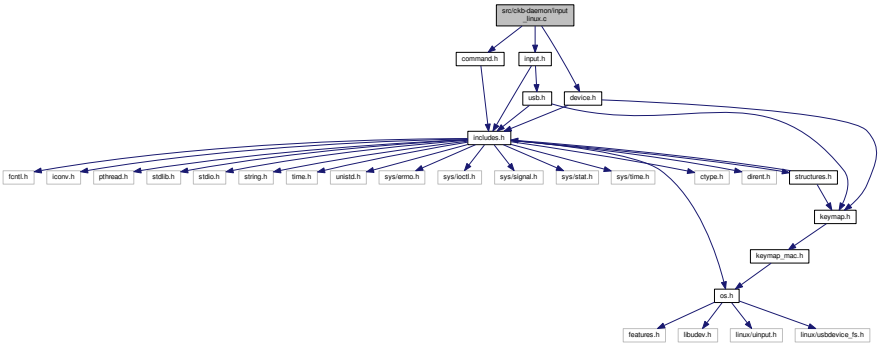
Here is the call graph for this function:



13.22 src/ckb-daemon/input_linux.c File Reference

```
#include "command.h"
#include "device.h"
#include "input.h"
```

Include dependency graph for input_linux.c:



Functions

- int uinputopen (struct uinput_user_dev *indev, int mouse)
- int os_inputopen (usbdevice *kb)

  os_inputopen
- void os_inputclose (usbdevice *kb)
- static void isync (usbdevice *kb)
- void os_keypress (usbdevice *kb, int scancode, int down)
- void os_mousemove (usbdevice *kb, int x, int y)
- void * _ledthread (void *ctx)
- int os_setupindicators (usbdevice *kb)

13.22.1 Function Documentation

13.22.1.1 void* _ledthread (void * *ctx*)

Definition at line 165 of file input_linux.c.

References dmutex, usbdevice::hw_ileds, usbdevice::uinput_kb, and usbdevice::vtable.

Referenced by os_setupindicators().

```
165                            {
166     usbdevice* kb = ctx;
167     uchar ileds = 0;
168     // Read LED events from the uinput device
169     struct input_event event;
170     while (read(kb->uinput_kb - 1, &event, sizeof(event)) > 0) {
171         if (event.type == EV_LED && event.code < 8){
172             char which = 1 << event.code;
173             if(event.value)
174                 ileds |= which;
175             else
176                 ileds &= ~which;
177         }
178         // Update them if needed
179         pthread_mutex_lock(dmutex(kb));
180         if(kb->hw_ileds != ileds){
181             kb->hw_ileds = ileds;
182             kb->vtable->updateindicators(kb, 0);
183         }
184         pthread_mutex_unlock(dmutex(kb));
185     }
186     return 0;
187 }
```

Here is the caller graph for this function:

13.22.1.2 static void isync (usbdevice * kb) [static]

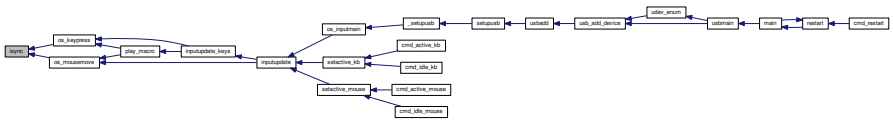
Definition at line 107 of file input_linux.c.

References ckb_warn, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

Referenced by os_keypress(), and os_mousemove().

```
107                                 {
108     struct input_event event;
109     memset(&event, 0, sizeof(event));
110     event.type = EV_SYN;
111     event.code = SYN_REPORT;
112     if(write(kb->uinput_kb - 1, &event, sizeof(event)) <= 0)
113         ckb_warn("uinput write failed: %s\n", strerror(errno));
114     if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
115         ckb_warn("uinput write failed: %s\n", strerror(errno));
116 }
```

Here is the caller graph for this function:



13.22.1.3 void os_inputclose (usbdevice * kb)

Definition at line 76 of file input_linux.c.

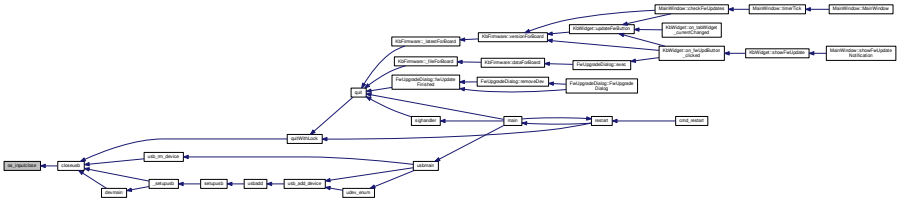
References ckb_warn, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

Referenced by closeusb().

```
76                                  {
77     if(kb->uinput_kb <= 0 || kb->uinput_mouse <= 0)
78         return;
79     // Set all keys released
80     struct input_event event;
81     memset(&event, 0, sizeof(event));
82     event.type = EV_KEY;
83     for(int key = 0; key < KEY_CNT; key++){
84         event.code = key;
85         if(write(kb->uinput_kb - 1, &event, sizeof(event)) <= 0)
86             ckb_warn("uinput write failed: %s\n", strerror(errno));
87         if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
88             ckb_warn("uinput write failed: %s\n", strerror(errno));
89     }
90     event.type = EV_SYN;
91     event.code = SYN_REPORT;
92     if(write(kb->uinput_kb - 1, &event, sizeof(event)) <= 0)
93         ckb_warn("uinput write failed: %s\n", strerror(errno));
94     if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
95         ckb_warn("uinput write failed: %s\n", strerror(errno));
96     // Close the keyboard
97     ioctl(kb->uinput_kb - 1, UI_DEV_DESTROY);
98     close(kb->uinput_kb - 1);
99     kb->uinput_kb = 0;
100     // Close the mouse
```

```
101     ioctl(kb->uinput_mouse - 1, UI_DEV_DESTROY);
102     close(kb->uinput_mouse - 1);
103     kb->uinput_mouse = 0;
104 }
```

Here is the caller graph for this function:



13.22.1.4 int os_inputopen (usbdevice * *kb*)

Parameters

kb	

Returns

Some tips on using `uinput_user_dev in`

Definition at line 55 of file input_linux.c.

References usbdevice::fwversion, INDEX_OF, keyboard, usbdevice::name, usbdevice::product, usbdevice::uinput-_kb, usbdevice::uinput_mouse, uinputopen(), and usbdevice::vendor.

Referenced by _setupusb().

```
55                                  {
56     // Create the new input device
57     int index = INDEX_OF(kb, keyboard);
58     struct uinput_user_dev indev;
59     memset(&indev, 0, sizeof(indev));
60     snprintf(indev.name, UINPUT_MAX_NAME_SIZE, "ckb%d: %s", index, kb->name);
61     indev.id.bustype = BUS_USB;
62     indev.id.vendor = kb->vendor;
63     indev.id.product = kb->product;
64     indev.id.version = kb->fwversion;
65     // Open keyboard
66     int fd = uinputopen(&indev, 0);
67     kb->uinput_kb = fd;
68     if(fd <= 0)
69         return 0;
70     // Open mouse
71     fd = uinputopen(&indev, 1);
72     kb->uinput_mouse = fd;
73     return fd <= 0;
74 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

13.22.1.5 void os_keypress (usbdevice * *kb*, int *scancode*, int *down*)

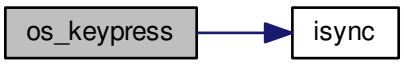
Definition at line 118 of file input_linux.c.

References BTN_WHEELDOWN, BTN_WHEELUP, ckb_warn, isync(), SCAN_MOUSE, usbdevice::uinput_kb, and usbdevice::uinput_mouse.

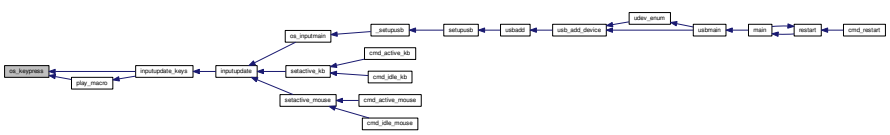
Referenced by inputupdate_keys(), and play_macro().

```
118                                                     {
119     struct input_event event;
120     memset(&event, 0, sizeof(event));
121     int is_mouse = 0;
122     if(scancode == BTN_WHEELUP || scancode == BTN_WHEELDOWN){
123         // The mouse wheel is a relative axis
124         if(!down)
125             return;
126         event.type = EV_REL;
127         event.code = REL_WHEEL;
128         event.value = (scancode == BTN_WHEELUP ? 1 : -1);
129         is_mouse = 1;
130     } else {
131         // Mouse buttons and key events are both EV_KEY. The scancodes are already correct, just remove the
    ckb bit
132         event.type = EV_KEY;
133         event.code = scancode & ~SCAN_MOUSE;
134         event.value = down;
135         is_mouse = !!(scancode & SCAN_MOUSE);
136     }
137     if(write((is_mouse ? kb->uinput_mouse : kb->uinput_kb) - 1, &event, sizeof(event))
    <= 0)
138         ckb_warn("uinput write failed: %s\n", strerror(errno));
139     else
140         isync(kb);
141 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.22.1.6 void os_mousemove (usbdevice * kb, int x, int y)

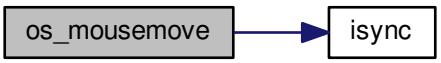
Definition at line 143 of file input_linux.c.

References ckb_warn, isync(), usbdevice::uinput_mouse, x, and y.

Referenced by inputupdate(), and play_macro().

```
143                                                 {
144     struct input_event event;
145     memset(&event, 0, sizeof(event));
146     event.type = EV_REL;
147     if(x != 0){
148         event.code = REL_X;
149         event.value = x;
150         if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
151             ckb_warn("uinput write failed: %s\n", strerror(errno));
152         else
153             isync(kb);
154     }
155     if(y != 0){
156         event.code = REL_Y;
157         event.value = y;
158         if(write(kb->uinput_mouse - 1, &event, sizeof(event)) <= 0)
159             ckb_warn("uinput write failed: %s\n", strerror(errno));
160         else
161             isync(kb);
162     }
163 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.22.1.7 int os_setupindicators (usbdevice * kb)

Definition at line 189 of file input_linux.c.

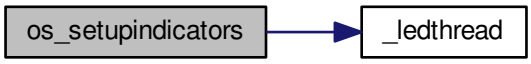
References _ledthread(), usbdevice::hw_ileds, usbdevice::hw_ileds_old, and usbdevice::ileds.

Referenced by _setupusb().

```
189                                         {
190     // Initialize LEDs to all off
191     kb->hw_ileds = kb->hw_ileds_old = kb->ileds = 0;
192     // Create and detach thread to read LED events
193     pthread_t thread;
194     int err = pthread_create(&thread, 0, _ledthread, kb);
195     if(err != 0)
```

```
196         return err;
197     pthread_detach(thread);
198     return 0;
199 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.22.1.8 int uinputopen (struct uinput_user_dev * *indev*, int *mouse*)

Definition at line 9 of file input_linux.c.

References ckb_err, and ckb_warn.

Referenced by os_inputopen().

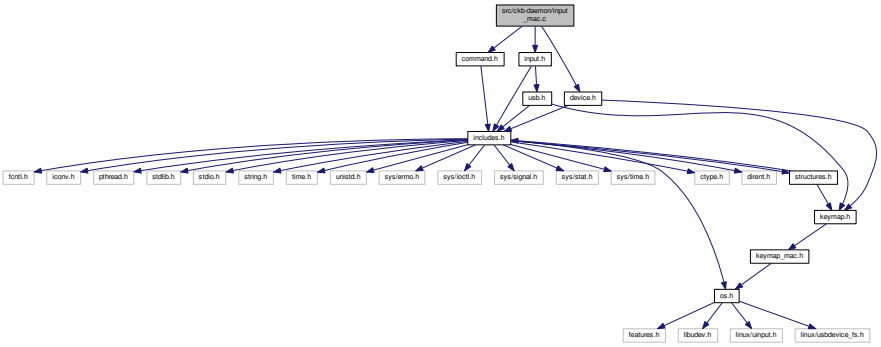
```
9                                                       {
10     int fd = open("/dev/uinput", O_RDWR);
11     if(fd < 0){
12         // If that didn't work, try /dev/input/uinput instead
13         fd = open("/dev/input/uinput", O_RDWR);
14         if(fd < 0){
15             ckb_err("Failed to open uinput: %s\n", strerror(errno));
16             return 0;
17         }
18     }
19     // Enable all keys and mouse buttons
20     ioctl(fd, UI_SET_EVBIT, EV_KEY);
21     for(int i = 0; i < KEY_CNT; i++)
22         ioctl(fd, UI_SET_KEYBIT, i);
23     if(mouse){
24         // Enable mouse axes
25         ioctl(fd, UI_SET_EVBIT, EV_REL);
26         for(int i = 0; i < REL_CNT; i++)
27             ioctl(fd, UI_SET_RELBIT, i);
28     } else {
29         // Enable LEDs
30         ioctl(fd, UI_SET_EVBIT, EV_LED);
31         for(int i = 0; i < LED_CNT; i++)
32             ioctl(fd, UI_SET_LEDBIT, i);
33         // Eanble autorepeat
34         ioctl(fd, UI_SET_EVBIT, EV_REP);
35     }
36     // Enable sychronization
37     ioctl(fd, UI_SET_EVBIT, EV_SYN);
38     // Create the device
39     if(write(fd, indev, sizeof(*indev)) <= 0)
40         ckb_warn("uinput write failed: %s\n", strerror(errno));
41     if(ioctl(fd, UI_DEV_CREATE)){
42         ckb_err("Failed to create uinput device: %s\n", strerror(errno));
43         close(fd);
44         return 0;
45     }
46     return fd + 1;
47 }
```

Here is the caller graph for this function:

13.23 src/ckb-daemon/input_mac.c File Reference

```
#include "command.h"
#include "device.h"
#include "input.h"
```

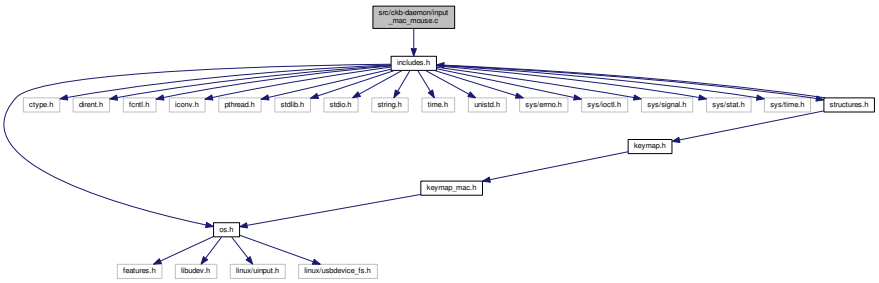
Include dependency graph for input_mac.c:



13.24 src/ckb-daemon/input_mac_mouse.c File Reference

```
#include "includes.h"
```

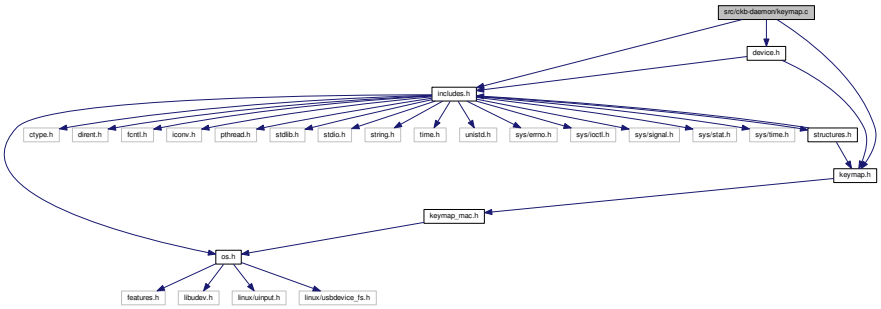
Include dependency graph for input_mac_mouse.c:



13.25 src/ckb-daemon/keymap.c File Reference

```
#include "device.h"
#include "includes.h"
#include "keymap.h"
```

Include dependency graph for keymap.c:



Macros

- #define BUTTON_HID_COUNT 5

Functions

- void hid_kb_translate (unsigned char ∗kbinput, int endpoint, int length, const unsigned char ∗urbinput)
- void hid_mouse_translate (unsigned char ∗kbinput, short ∗xaxis, short ∗yaxis, int endpoint, int length, const unsigned char ∗urbinput, ushort fwversion)
- void corsair_kbcopy (unsigned char ∗kbinput, int endpoint, const unsigned char ∗urbinput)
- void corsair_mousecopy (unsigned char ∗kbinput, int endpoint, const unsigned char ∗urbinput)

Variables

- const key keymap [(((152+22+12)+25)+12)]

13.25.1 Macro Definition Documentation

13.25.1.1 #define BUTTON_HID_COUNT 5

Definition at line 389 of file keymap.c.

Referenced by corsair_mousecopy(), and hid_mouse_translate().

13.25.2 Function Documentation

13.25.2.1 void corsair_kbcopy (unsigned char ∗ *kbinput,* int *endpoint,* const unsigned char ∗ *urbinput*)

Definition at line 420 of file keymap.c.

References N_KEYBYTES_HW.

Referenced by os_inputmain().

```
420                                                                                 {
421     if(endpoint == 2 || endpoint == -2){
422         if(urbinput[0] != 3)
423             return;
424         urbinput++;
425     }
426     memcpy(kbinput, urbinput, N_KEYBYTES_HW);
427 }
```

Here is the caller graph for this function:

13.25.2.2 void corsair_mousecopy (unsigned char * *kbinput*, int *endpoint*, const unsigned char * *urbinput*)

Definition at line 429 of file keymap.c.

References BUTTON_HID_COUNT, CLEAR_KEYBIT, MOUSE_BUTTON_FIRST, N_BUTTONS_HW, and SET_KEYBIT.

Referenced by os_inputmain().

```
429                                                                                                 {
430     if(endpoint == 2 || endpoint == -2){
431         if(urbinput[0] != 3)
432             return;
433         urbinput++;
434     }
435     for(int bit = BUTTON_HID_COUNT; bit < N_BUTTONS_HW; bit++){
436         int byte = bit / 8;
437         uchar test = 1 << (bit % 8);
438         if(urbinput[byte] & test)
439             SET_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
440         else
441             CLEAR_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
442     }
443 }
```

Here is the caller graph for this function:

13.25.2.3 void hid_kb_translate (unsigned char * *kbinput*, int *endpoint*, int *length*, const unsigned char * *urbinput*)

Definition at line 246 of file keymap.c.

References ckb_warn, CLEAR_KEYBIT, and SET_KEYBIT.

Referenced by os_inputmain().

```
246                                                                                                 {
247     if(length < 1)
248         return;
249     // LUT for HID -> Corsair scancodes (-1 for no scan code, -2 for currently unsupported)
250     // Modified from Linux drivers/hid/usbhid/usbkbd.c, key codes replaced with keymap array indices and
      K95 keys added
251     // Make sure the indices match the keyindex as passed to nprintkey() in notify.c
252     static const short hid_codes[256] = {
253         -1,  -1,  -1,  -1,  37,  54,  52,  39,  27,  40,  41,  42,  32,  43,  44,  45,
254         56,  55,  33,  34,  25,  28,  38,  29,  31,  53,  26,  51,  30,  50,  13,  14,
255         15,  16,  17,  18,  19,  20,  21,  22,  82,   0,  86,  24,  64,  23,  84,  35,
256         79,  80,  81,  46,  47,  12,  57,  58,  59,  36,   1,   2,   3,   4,   5,   6,
257          7,   8,   9,  10,  11,  72,  73,  74,  75,  76,  77,  78,  87,  88,  89,  95,
258         93,  94,  92, 102, 103, 104, 105, 106, 107, 115, 116, 117, 112, 113, 114, 108,
259        109, 110, 118, 119,  49,  69,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,
260         -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  98,  -2,  -2,  -2,  -2,  -2,  -2,  97,
261        130, 131,  -1,  -1,  -1,  -2,  -1,  83,  66,  85, 145, 144,  -2,  -1,  -1,  -1,
262         -2,  -2,  -2,  -2,  -2,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
263         -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
264         -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
265         -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -3,  -1,  -1,  -1,  // <- -3 = non-RGB
```

```
    program key
266         120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 136, 137, 138, 139, 140, 141,
267          60,  48,  62,  61,  91,  90,  67,  68, 142, 143,  99, 101,  -2, 130, 131,  97,
268          -2, 133, 134, 135,  -2,  96,  -2, 132,  -2,  -2,  71,  71,  71,  71,  -1,  -1,
269     };
270     switch(endpoint){
271     case 1:
272     case -1:
273         // EP 1: 6KRO input (RGB and non-RGB)
274         // Clear previous input
275         for(int i = 0; i < 256; i++){
276             if(hid_codes[i] >= 0)
277                 CLEAR_KEYBIT(kbinput, hid_codes[i]);
278         }
279         // Set new input
280         for(int i = 0; i < 8; i++){
281             if((urbinput[0] >> i) & 1)
282                 SET_KEYBIT(kbinput, hid_codes[i + 224]);
283         }
284         for(int i = 2; i < length; i++){
285             if(urbinput[i] > 3){
286                 int scan = hid_codes[urbinput[i]];
287                 if(scan >= 0)
288                     SET_KEYBIT(kbinput, scan);
289                 else
290                     ckb_warn("Got unknown key press %d on EP 1\n", urbinput[i]);
291             }
292         }
293         break;
294     case -2:
295         // EP 2 RGB: NKRO input
296         if(urbinput[0] == 1){
297             // Type 1: standard key
298             if(length != 21)
299                 return;
300             for(int bit = 0; bit < 8; bit++){
301                 if((urbinput[1] >> bit) & 1)
302                     SET_KEYBIT(kbinput, hid_codes[bit + 224]);
303                 else
304                     CLEAR_KEYBIT(kbinput, hid_codes[bit + 224]);
305             }
306             for(int byte = 0; byte < 19; byte++){
307                 char input = urbinput[byte + 2];
308                 for(int bit = 0; bit < 8; bit++){
309                     int keybit = byte * 8 + bit;
310                     int scan = hid_codes[keybit];
311                     if((input >> bit) & 1){
312                         if(scan >= 0)
313                             SET_KEYBIT(kbinput, hid_codes[keybit]);
314                         else
315                             ckb_warn("Got unknown key press %d on EP 2\n", keybit);
316                     } else if(scan >= 0)
317                         CLEAR_KEYBIT(kbinput, hid_codes[keybit]);
318                 }
319             }
320             break;
321         } else if (urbinput[0] == 2)
322             ; // Type 2: media key (implicitly falls through)
323         else
324             break;  // No other known types
325         /* FALLTHRU */
326     case 2:
327         // EP 2 Non-RGB: media keys
328         CLEAR_KEYBIT(kbinput, 97);          // mute
329         CLEAR_KEYBIT(kbinput, 98);          // stop
330         CLEAR_KEYBIT(kbinput, 99);          // prev
331         CLEAR_KEYBIT(kbinput, 100);         // play
332         CLEAR_KEYBIT(kbinput, 101);         // next
333         CLEAR_KEYBIT(kbinput, 130);         // volup
334         CLEAR_KEYBIT(kbinput, 131);         // voldn
335         for(int i = 0; i < length; i++){
336             switch(urbinput[i]){
337             case 181:
338                 SET_KEYBIT(kbinput, 101);   // next
339                 break;
340             case 182:
341                 SET_KEYBIT(kbinput, 99);    // prev
342                 break;
343             case 183:
344                 SET_KEYBIT(kbinput, 98);    // stop
345                 break;
346             case 205:
347                 SET_KEYBIT(kbinput, 100);   // play
348                 break;
349             case 226:
350                 SET_KEYBIT(kbinput, 97);    // mute
351                 break;
```

```
352             case 233:
353                 SET_KEYBIT(kbinput, 130);   // volup
354                 break;
355             case 234:
356                 SET_KEYBIT(kbinput, 131);   // voldn
357                 break;
358             }
359         }
360         break;
361     case 3:
362         // EP 3 non-RGB: NKRO input
363         if(length != 15)
364             return;
365         for(int bit = 0; bit < 8; bit++){
366             if((urbinput[0] >> bit) & 1)
367                 SET_KEYBIT(kbinput, hid_codes[bit + 224]);
368             else
369                 CLEAR_KEYBIT(kbinput, hid_codes[bit + 224]);
370         }
371         for(int byte = 0; byte < 14; byte++){
372             char input = urbinput[byte + 1];
373             for(int bit = 0; bit < 8; bit++){
374                 int keybit = byte * 8 + bit;
375                 int scan = hid_codes[keybit];
376                 if((input >> bit) & 1){
377                     if(scan >= 0)
378                         SET_KEYBIT(kbinput, hid_codes[keybit]);
379                     else
380                         ckb_warn("Got unknown key press %d on EP 3\n", keybit);
381                 } else if(scan >= 0)
382                     CLEAR_KEYBIT(kbinput, hid_codes[keybit]);
383             }
384         }
385         break;
386     }
387 }
```

Here is the caller graph for this function:

13.25.2.4 void hid_mouse_translate (unsigned char * *kbinput*, short * *xaxis*, short * *yaxis*, int *endpoint*, int *length*, const unsigned char * *urbinput*, ushort *fwversion*)

Definition at line 391 of file keymap.c.

References BUTTON_HID_COUNT, CLEAR_KEYBIT, MOUSE_BUTTON_FIRST, MOUSE_EXTRA_FIRST, and SET_KEYBIT.

Referenced by os_inputmain().

```
391                                                 {
392     //The HID Input Endpoint on FWv3 is 64 bytes, so we can't check for length.
393     if((endpoint != 2 && endpoint != -2) || (fwversion < 0x300 && length < 10))
394         return;
395     // EP 2: mouse input
396     if(urbinput[0] != 1)
397         return;
398     // Byte 1 = mouse buttons (bitfield)
399     for(int bit = 0; bit < BUTTON_HID_COUNT; bit++){
400         if(urbinput[1] & (1 << bit))
401             SET_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
402         else
403             CLEAR_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
404     }
405     // Bytes 5 - 8: movement
406     *xaxis += *(short*)(urbinput + 5);
407     *yaxis += *(short*)(urbinput + 7);
408     // Byte 9: wheel
409     char wheel = urbinput[9];
410     if(wheel > 0)
411         SET_KEYBIT(kbinput, MOUSE_EXTRA_FIRST);         // wheelup
```

```
412     else
413         CLEAR_KEYBIT(kbinput, MOUSE_EXTRA_FIRST);
414     if(wheel < 0)
415         SET_KEYBIT(kbinput, MOUSE_EXTRA_FIRST + 1);     // wheeldn
416     else
417         CLEAR_KEYBIT(kbinput, MOUSE_EXTRA_FIRST + 1);
418 }
```

Here is the caller graph for this function:

13.25.3 Variable Documentation

13.25.3.1 const key keymap[(((152+22+12)+25)+12)]

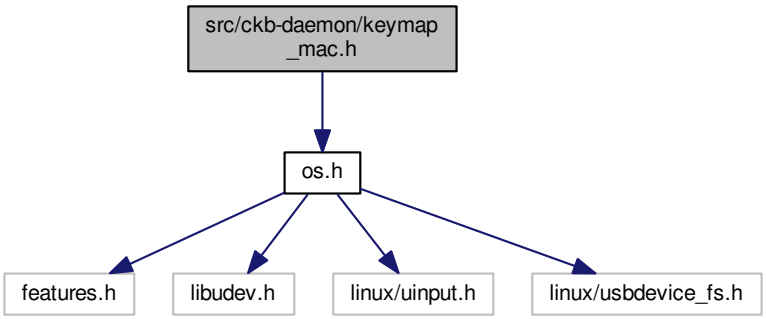
Definition at line 5 of file keymap.c.

Referenced by _cmd_get(), _cmd_macro(), cmd_bind(), cmd_rebind(), cmd_rgb(), initbind(), inputupdate_keys(), nprintkey(), printrgb(), readcmd(), and setactive_kb().

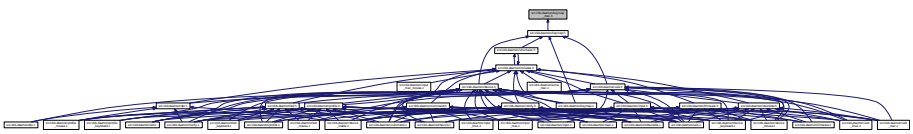
13.26 src/ckb-daemon/keymap_mac.h File Reference

```
#include "os.h"
```

Include dependency graph for keymap_mac.h:



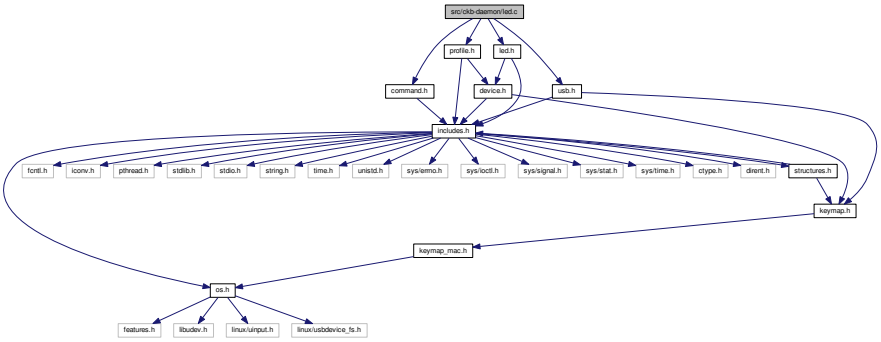
This graph shows which files directly or indirectly include this file:



13.27 src/ckb-daemon/led.c File Reference

```
#include "command.h"
#include "led.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for led.c:



Functions

- void cmd_rgb (usbdevice ∗kb, usbmode ∗mode, int dummy, int keyindex, const char ∗code)
- static uchar iselect (const char ∗led)
- void cmd_ioff (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗led)
- void cmd_ion (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗led)
- void cmd_iauto (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗led)
- void cmd_inotify (usbdevice ∗kb, usbmode ∗mode, int nnumber, int dummy, const char ∗led)
- static int has_key (const char ∗name, const usbdevice ∗kb)
- char ∗ printrgb (const lighting ∗light, const usbdevice ∗kb)

13.27.1 Function Documentation

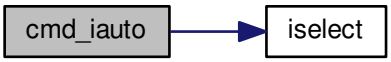
13.27.1.1 void cmd_iauto (usbdevice ∗ *kb,* usbmode ∗ *mode,* int *dummy1,* int *dummy2,* const char ∗ *led*)

Definition at line 63 of file led.c.

References usbmode::ioff, usbmode::ion, iselect(), and usbdevice::vtable.

```
63                                                                                  {
64     (void)dummy1;
65     (void)dummy2;
66
67     uchar bits = iselect(led);
68     // Remove the bits from both ioff and ion
69     mode->ioff &= ~bits;
70     mode->ion &= ~bits;
71     kb->vtable->updateindicators(kb, 0);
72 }
```

Here is the call graph for this function:



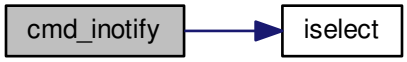
13.27.1.2 void cmd_inotify (usbdevice * *kb*, usbmode * *mode*, int *nnumber*, int *dummy*, const char * *led*)

Definition at line 74 of file led.c.

References usbmode::inotify, and iselect().

```
74                                                                                  {
75     (void)kb;
76     (void)dummy;
77
78     uchar bits = iselect(led);
79     if(strstr(led, ":off"))
80         // Turn notifications for these bits off
81         mode->inotify[nnumber] &= ~bits;
82     else
83         // Turn notifications for these bits on
84         mode->inotify[nnumber] |= bits;
85 }
```

Here is the call graph for this function:



13.27.1.3 void cmd_ioff (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *led*)

Definition at line 41 of file led.c.

References usbmode::ioff, usbmode::ion, iselect(), and usbdevice::vtable.

```
41                                                                                  {
42     (void)dummy1;
43     (void)dummy2;
44
45     uchar bits = iselect(led);
46     // Add the bits to ioff, remove them from ion
47     mode->ioff |= bits;
48     mode->ion &= ~bits;
49     kb->vtable->updateindicators(kb, 0);
50 }
```

Here is the call graph for this function:

13.27.1.4 void cmd_ion (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *led*)

Definition at line 52 of file led.c.

References usbmode::ioff, usbmode::ion, iselect(), and usbdevice::vtable.

```
52                                                                                  {
53     (void)dummy1;
54     (void)dummy2;
55
56     uchar bits = iselect(led);
57     // Remove the bits from ioff, add them to ion
58     mode->ioff &= ~bits;
59     mode->ion |= bits;
60     kb->vtable->updateindicators(kb, 0);
61 }
```

Here is the call graph for this function:

13.27.1.5 void cmd_rgb (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *code*)

Definition at line 6 of file led.c.

References lighting::b, lighting::g, keymap, key::led, usbmode::light, lighting::r, and lighting::sidelight.

```
6                                                                                   {
7     (void)kb;
8     (void)dummy;
9
10     int index = keymap[keyindex].led;
11     if(index < 0) {
12         if (index == -2){      // Process strafe sidelights
13             uchar sideshine;
14             if (sscanf(code, "%2hhx",&sideshine)) // monochromatic
15                 mode->light.sidelight = sideshine;
16         }
17         return;
18     }
19     uchar r, g, b;
20     if(sscanf(code, "%2hhx%2hhx%2hhx", &r, &g, &b) == 3){
21         mode->light.r[index] = r;
```

```
22          mode->light.g[index] = g;
23          mode->light.b[index] = b;
24      }
25  }
```

13.27.1.6 static int has_key (const char * *name*, const usbdevice * *kb*) [static]

Definition at line 88 of file led.c.

References IS_K63, IS_K65, IS_K95, IS_MOUSE, IS_SABRE, IS_SCIMITAR, usbdevice::product, and usbdevice::vendor.

Referenced by printrgb().

```
88                                                      {
89      if(!name)
90          return 0;
91      if(IS_MOUSE(kb->vendor, kb->product)){
92          // Mice only have the RGB zones
93          if((IS_SABRE(kb) || IS_SCIMITAR(kb)) && !strcmp(name, "wheel"))
94              return 1;
95          if(IS_SCIMITAR(kb) && !strcmp(name, "thumb"))
96              return 1;
97          if(strstr(name, "dpi") == name || !strcmp(name, "front") || !strcmp(name, "back"))
98              return 1;
99          return 0;
100     } else {
101         // But keyboards don't have them at all
102         if(strstr(name, "dpi") == name || !strcmp(name, "front") || !strcmp(name, "back") || !strcmp(name,
    "wheel") || !strcmp(name, "thumb"))
103             return 0;
104         // Only K95 has G keys and M keys (G1 - G18, MR, M1 - M3)
105         if(!IS_K95(kb) && ((name[0] == 'g' && name[1] >= '1' && name[1] <= '9') || (name[0] == 'm' &&
    (name[1] == 'r' || name[1] == '1' || name[1] == '2' || name[1] == '3'))))
106             return 0;
107         // K65 and K63 have lights on VolUp/VolDn
108         if((!IS_K65(kb) && !IS_K63(kb)) && (!strcmp(name, "volup") || !strcmp(name, "voldn")))
109             return 0;
110         // K65 lacks numpad and media buttons
111         if(IS_K65(kb) && (strstr(name, "num") == name || !strcmp(name, "stop") || !strcmp(name, "prev
    ") || !strcmp(name, "play") || !strcmp(name, "next")))
112             return 0;
113         // K63 lacks numpad
114         if(IS_K63(kb) && strstr(name, "num") == name)
115             return 0;
116     }
117     return 1;
118 }
```

Here is the caller graph for this function:

13.27.1.7 static uchar iselect (const char * *led*) [static]

Definition at line 28 of file led.c.

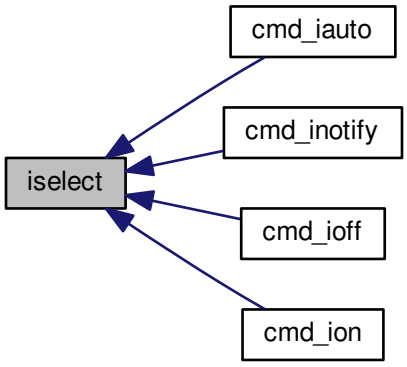
References I_CAPS, I_NUM, and I_SCROLL.

Referenced by cmd_iauto(), cmd_inotify(), cmd_ioff(), and cmd_ion().

```
28                                      {
```

```
29     int result = 0;
30     if(!strncmp(led, "num", 3) || strstr(led, ",num"))
31         result |= I_NUM;
32     if(!strncmp(led, "caps", 4) || strstr(led, ",caps"))
33         result |= I_CAPS;
34     if(!strncmp(led, "scroll", 6) || strstr(led, ",scroll"))
35         result |= I_SCROLL;
36     if(!strncmp(led, "all", 3) || strstr(led, ",all"))
37         result |= I_NUM | I_CAPS | I_SCROLL;
38     return result;
39 }
```

Here is the caller graph for this function:



13.27.1.8 char* printrgb (const lighting * *light*, const usbdevice * *kb*)

Definition at line 120 of file led.c.

References lighting::b, lighting::g, has_key(), keymap, key::led, N_KEYS_EXTENDED, key::name, and lighting::r.

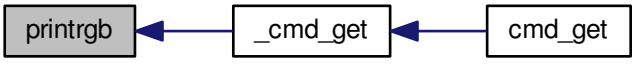
Referenced by _cmd_get().

```
120                                                         {
121     uchar r[N_KEYS_EXTENDED], g[N_KEYS_EXTENDED], b[
    N_KEYS_EXTENDED];
122     const uchar* mr = light->r;
123     const uchar* mg = light->g;
124     const uchar* mb = light->b;
125     for(int i = 0; i < N_KEYS_EXTENDED; i++){
126         // Translate the key index to an RGB index using the key map
127         int k = keymap[i].led;
128         if(k < 0)
129             continue;
130         r[i] = mr[k];
131         g[i] = mg[k];
132         b[i] = mb[k];
133     }
134     // Make a buffer to track key names and to filter out duplicates
135     char names[N_KEYS_EXTENDED][11];
136     for(int i = 0; i < N_KEYS_EXTENDED; i++){
137         const char* name = keymap[i].name;
138         if(keymap[i].led < 0 || !has_key(name, kb))
139             names[i][0] = 0;
140         else
141             strncpy(names[i], name, 11);
142     }
143     // Check to make sure these aren't all the same color
144     int same = 1;
145     for(int i = 1; i < N_KEYS_EXTENDED; i++){
146         if(!names[i][0])
```

```
147             continue;
148         if(r[i] != r[0] || g[i] != g[0] || b[i] != b[0]){
149             same = 0;
150             break;
151         }
152     }
153     // If they are, just output that color
154     if(same){
155         char* buffer = malloc(7);
156         snprintf(buffer, 7, "%02x%02x%02x", r[0], g[0], b[0]);
157         return buffer;
158     }
159     const int BUFFER_LEN = 4096;    // Should be more than enough to fit all keys
160     char* buffer = malloc(BUFFER_LEN);
161     int length = 0;
162     for(int i = 0; i < N_KEYS_EXTENDED; i++){
163         if(!names[i][0])
164             continue;
165         // Print the key name
166         int newlen = 0;
167         snprintf(buffer + length, BUFFER_LEN - length, length == 0 ? "%s%n" : " %s%n", names[i], &newlen);
168         length += newlen;
169         // Look ahead to see if any other keys have this color. If so, print them here as well.
170         uchar kr = r[i], kg = g[i], kb = b[i];
171         for(int j = i + 1; j < N_KEYS_EXTENDED; j++){
172             if(!names[j][0])
173                 continue;
174             if(r[j] != kr || g[j] != kg || b[j] != kb)
175                 continue;
176             snprintf(buffer + length, BUFFER_LEN - length, ",%s%n", names[j], &newlen);
177             length += newlen;
178             // Erase the key's name so it won't get printed later
179             names[j][0] = 0;
180         }
181         // Print the color
182         snprintf(buffer + length, BUFFER_LEN - length, ":%02x%02x%02x%n", kr, kg, kb, &newlen);
183         length += newlen;
184     }
185     return buffer;
186 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

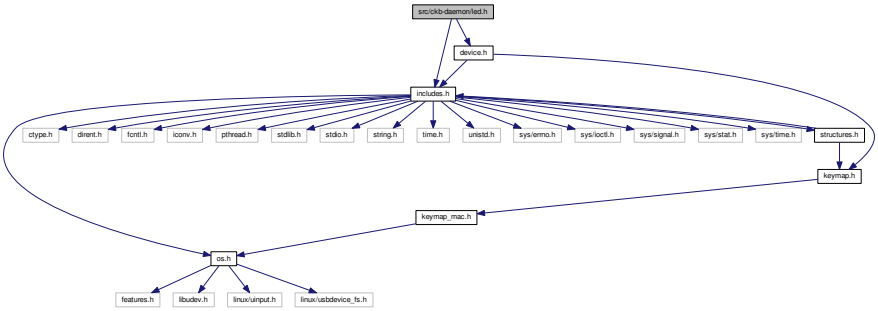


13.28 src/ckb-daemon/led.h File Reference

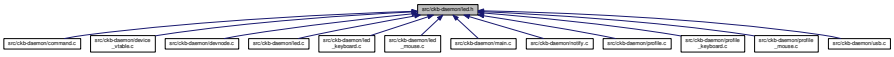
```
#include "includes.h"
```

```
#include "device.h"
```

Include dependency graph for led.h:



This graph shows which files directly or indirectly include this file:



Functions

- int updatergb_kb (usbdevice ∗kb, int force)
- int updatergb_mouse (usbdevice ∗kb, int force)
- int savergb_kb (usbdevice ∗kb, lighting ∗light, int mode)
- int savergb_mouse (usbdevice ∗kb, lighting ∗light, int mode)
- int loadrgb_kb (usbdevice ∗kb, lighting ∗light, int mode)
- int loadrgb_mouse (usbdevice ∗kb, lighting ∗light, int mode)
- char ∗ printrgb (const lighting ∗light, const usbdevice ∗kb)
- void cmd_rgb (usbdevice ∗kb, usbmode ∗mode, int dummy, int keyindex, const char ∗code)
- void cmd_ioff (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗led)
- void cmd_ion (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗led)
- void cmd_iauto (usbdevice ∗kb, usbmode ∗mode, int dummy1, int dummy2, const char ∗led)
- void cmd_inotify (usbdevice ∗kb, usbmode ∗mode, int nnumber, int dummy, const char ∗led)

13.28.1 Function Documentation

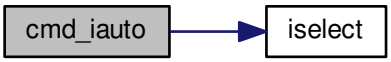
13.28.1.1 void cmd_iauto (usbdevice ∗ *kb,* usbmode ∗ *mode,* int *dummy1,* int *dummy2,* const char ∗ *led*)

Definition at line 63 of file led.c.

References usbmode::ioff, usbmode::ion, iselect(), and usbdevice::vtable.

```
63                                                                                  {
64     (void)dummy1;
65     (void)dummy2;
66
67     uchar bits = iselect(led);
68     // Remove the bits from both ioff and ion
69     mode->ioff &= ~bits;
70     mode->ion &= ~bits;
71     kb->vtable->updateindicators(kb, 0);
72 }
```

Here is the call graph for this function:



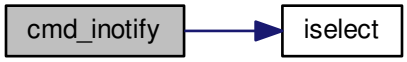
13.28.1.2 void cmd_inotify (usbdevice * *kb*, usbmode * *mode*, int *nnumber*, int *dummy*, const char * *led*)

Definition at line 74 of file led.c.

References usbmode::inotify, and iselect().

```
74                                                                                  {
75     (void)kb;
76     (void)dummy;
77
78     uchar bits = iselect(led);
79     if(strstr(led, ":off"))
80         // Turn notifications for these bits off
81         mode->inotify[nnumber] &= ~bits;
82     else
83         // Turn notifications for these bits on
84         mode->inotify[nnumber] |= bits;
85 }
```

Here is the call graph for this function:



13.28.1.3 void cmd_ioff (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *led*)

Definition at line 41 of file led.c.

References usbmode::ioff, usbmode::ion, iselect(), and usbdevice::vtable.

```
41                                                                                  {
42     (void)dummy1;
43     (void)dummy2;
44
45     uchar bits = iselect(led);
46     // Add the bits to ioff, remove them from ion
47     mode->ioff |= bits;
48     mode->ion &= ~bits;
49     kb->vtable->updateindicators(kb, 0);
50 }
```

Here is the call graph for this function:

13.28.1.4 void cmd_ion (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *led*)

Definition at line 52 of file led.c.

References usbmode::ioff, usbmode::ion, iselect(), and usbdevice::vtable.

```
52                                                                                  {
53      (void)dummy1;
54      (void)dummy2;
55
56      uchar bits = iselect(led);
57      // Remove the bits from ioff, add them to ion
58      mode->ioff &= ~bits;
59      mode->ion |= bits;
60      kb->vtable->updateindicators(kb, 0);
61  }
```

Here is the call graph for this function:

13.28.1.5 void cmd_rgb (usbdevice * *kb*, usbmode * *mode*, int *dummy*, int *keyindex*, const char * *code*)

Definition at line 6 of file led.c.

References lighting::b, lighting::g, keymap, key::led, usbmode::light, lighting::r, and lighting::sidelight.

```
6                                                                                   {
7       (void)kb;
8       (void)dummy;
9
10      int index = keymap[keyindex].led;
11      if(index < 0) {
12          if (index == -2){      // Process strafe sidelights
13              uchar sideshine;
14              if (sscanf(code, "%2hhx",&sideshine)) // monochromatic
15                  mode->light.sidelight = sideshine;
16          }
17          return;
18      }
19      uchar r, g, b;
20      if(sscanf(code, "%2hhx%2hhx%2hhx", &r, &g, &b) == 3){
21          mode->light.r[index] = r;
```

```
22          mode->light.g[index] = g;
23          mode->light.b[index] = b;
24      }
25  }
```

13.28.1.6 int loadrgb_kb (usbdevice * *kb*, lighting * *light*, int *mode*)

Since Firmware Version 2.05 for K95RGB the answers for getting the stored color-maps from the hardware has changed a bit. So comparing for the correct answer cannot validate against the cmd, and has to be done against a third map. Up to now we know, that K70RGB Pro and K70 Lux RGB have firmware version 2.04 and having the problem also. So we have to determine in the most inner loop the firmware version and type of KB to select the correct compare-table.

Read colors

```
                                                    < That is the old comparison method: you get back what you sent.
```

Normally a firmware version >= 2.05 runs with the new compare array. Up to now there is a 2.04 running in K70 RGB Lux with the same behavior. It seems that K70RGB has the same problem

Definition at line 190 of file led_keyboard.c.

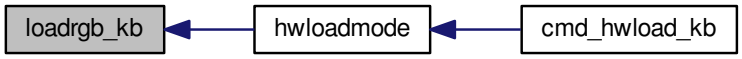
References lighting::b, ckb_err, usbdevice::fwversion, lighting::g, IS_V2_OVERRIDE, MSG_SIZE, N_KEYS_HW, P_K70_LUX, P_K70_LUX_NRGB, usbdevice::product, lighting::r, usbrecv, and usbsend.

Referenced by hwloadmode().

```
190                                                     {
191     if(kb->fwversion >= 0x0120 || IS_V2_OVERRIDE(kb)){
192         uchar data_pkt[12][MSG_SIZE] = {
193             { 0x0e, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x01 },
194             { 0xff, 0x01, 60, 0 },
195             { 0xff, 0x02, 60, 0 },
196             { 0xff, 0x03, 24, 0 },
197             { 0x0e, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x02 },
198             { 0xff, 0x01, 60, 0 },
199             { 0xff, 0x02, 60, 0 },
200             { 0xff, 0x03, 24, 0 },
201             { 0x0e, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x03 },
202             { 0xff, 0x01, 60, 0 },
203             { 0xff, 0x02, 60, 0 },
204             { 0xff, 0x03, 24, 0 },
205         };
206         uchar in_pkt[4][MSG_SIZE] = {
207             { 0x0e, 0x14, 0x03, 0x01 },
208             { 0xff, 0x01, 60, 0 },
209             { 0xff, 0x02, 60, 0 },
210             { 0xff, 0x03, 24, 0 },
211         };
212
218
219         uchar cmp_pkt[4][4] = {
220             { 0x0e, 0x14, 0x03, 0x01 },
221             { 0x0e, 0xff, 0x01, 60 },
222             { 0x0e, 0xff, 0x02, 60 },
223             { 0x0e, 0xff, 0x03, 24 },
224         };
226         uchar* colors[3] = { light->r, light->g, light->b };
227         for(int clr = 0; clr < 3; clr++){
228             for(int i = 0; i < 4; i++){
229                 if(!usbrecv(kb, data_pkt[i + clr * 4], in_pkt[i]))
230                     return -1;
231
232                 uchar* comparePacket = data_pkt[i + clr * 4];
233                 if ((kb->fwversion >= 0x205)
237                         || ((kb->fwversion >= 0x204)
238                             && ((kb->product == P_K70_LUX_NRGB) || (kb->
    product == P_K70_LUX)))) {
239                     comparePacket = cmp_pkt[i];
240                 }
241
242                 if (memcmp(in_pkt[i], comparePacket, 4)) {
243                     ckb_err("Bad input header\n");
244                     ckb_err("color = %d, i = %d, mode = %d\nOutput (Request): %2.2x %2.2x %2.2x
     %2.2x\nInput(Reply): %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x\n", clr, i, mode,
245                         comparePacket[0], comparePacket[1], comparePacket[2], comparePacket[3],
246                         in_pkt[i][0], in_pkt[i][1], in_pkt[i][2], in_pkt[i][3], in_pkt[i][4], in_pkt[i][5],
```

```
      in_pkt[i][6], in_pkt[i][7]);
247                   in_pkt[2][0] = 0x99;
248                   in_pkt[2][1] = 0x99;
249                   in_pkt[2][2] = 0x99;
250                   in_pkt[2][3] = 0x99;
251                   usbrecv(kb, in_pkt[2], in_pkt[2]); // just to find it in the wireshark log
252                   return -1;
253               }
254           }
255           // Copy colors to lighting. in_pkt[0] is irrelevant.
256           memcpy(colors[clr], in_pkt[1] + 4, 60);
257           memcpy(colors[clr] + 60, in_pkt[2] + 4, 60);
258           memcpy(colors[clr] + 120, in_pkt[3] + 4, 24);
259       }
260    } else {
261       uchar data_pkt[5][MSG_SIZE] = {
262           { 0x0e, 0x14, 0x02, 0x01, 0x01, mode + 1, 0 },
263           { 0xff, 0x01, 60, 0 },
264           { 0xff, 0x02, 60, 0 },
265           { 0xff, 0x03, 60, 0 },
266           { 0xff, 0x04, 36, 0 },
267       };
268       uchar in_pkt[4][MSG_SIZE] = {
269           { 0xff, 0x01, 60, 0 },
270           { 0xff, 0x02, 60, 0 },
271           { 0xff, 0x03, 60, 0 },
272           { 0xff, 0x04, 36, 0 },
273       };
274       // Write initial packet
275       if(!usbsend(kb, data_pkt[0], 1))
276           return -1;
277       // Read colors
278       for(int i = 1; i < 5; i++){
279           if(!usbrecv(kb, data_pkt[i],in_pkt[i - 1]))
280               return -1;
281           if(memcmp(in_pkt[i - 1], data_pkt[i], 4)){
282               ckb_err("Bad input header\n");
283               return -1;
284           }
285       }
286       // Copy the data back to the mode
287       uint8_t mr[N_KEYS_HW / 2], mg[N_KEYS_HW / 2], mb[
     N_KEYS_HW / 2];
288       memcpy(mr,      in_pkt[0] +  4, 60);
289       memcpy(mr + 60, in_pkt[1] +  4, 12);
290       memcpy(mg,      in_pkt[1] + 16, 48);
291       memcpy(mg + 48, in_pkt[2] +  4, 24);
292       memcpy(mb,      in_pkt[2] + 28, 36);
293       memcpy(mb + 36, in_pkt[3] +  4, 36);
294       // Unpack LED data to 8bpc format
295       for(int i = 0; i < N_KEYS_HW; i++){
296           int     i_2 = i / 2;
297           uint8_t r, g, b;
298
299           // 3-bit intensities stored in alternate nybbles.
300           if (i & 1) {
301               r = 7 - (mr[i_2] >> 4);
302               g = 7 - (mg[i_2] >> 4);
303               b = 7 - (mb[i_2] >> 4);
304           } else {
305               r = 7 - (mr[i_2] & 0x0F);
306               g = 7 - (mg[i_2] & 0x0F);
307               b = 7 - (mb[i_2] & 0x0F);
308           }
309           // Scale 3-bit values up to 8 bits.
310           light->r[i] = r << 5 | r << 2 | r >> 1;
311           light->g[i] = g << 5 | g << 2 | g >> 1;
312           light->b[i] = b << 5 | b << 2 | b >> 1;
313       }
314    }
315    return 0;
316 }
```

Here is the caller graph for this function:



13.28.1.7 int loadrgb_mouse (usbdevice * *kb*, lighting * *light*, int *mode*)

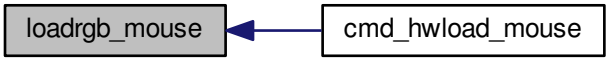
Definition at line 87 of file led_mouse.c.

References lighting::b, ckb_err, lighting::g, IS_SABRE, IS_SCIMITAR, LED_DPI, LED_MOUSE, MSG_SIZE, lighting::r, and usbrecv.

Referenced by cmd_hwload_mouse().

```
87                                                          {
88      (void)mode;
89
90      uchar data_pkt[MSG_SIZE] = { 0x0e, 0x13, 0x10, 1, 0 };
91      uchar in_pkt[MSG_SIZE] = { 0 };
92      // Load each RGB zone
93      int zonecount = IS_SCIMITAR(kb) ? 4 : IS_SABRE(kb) ? 3 : 2;
94      for(int i = 0; i < zonecount; i++){
95          if(!usbrecv(kb, data_pkt, in_pkt))
96              return -1;
97          if(memcmp(in_pkt, data_pkt, 4)){
98              ckb_err("Bad input header\n");
99              return -2;
100         }
101         // Copy data
102         int led = LED_MOUSE + i;
103         if(led >= LED_DPI)
104             led++;          // Skip DPI light
105         light->r[led] = in_pkt[4];
106         light->g[led] = in_pkt[5];
107         light->b[led] = in_pkt[6];
108         // Set packet for next zone
109         data_pkt[2]++;
110     }
111     return 0;
112 }
```

Here is the caller graph for this function:



13.28.1.8 char* printrgb (const lighting * *light*, const usbdevice * *kb*)

Definition at line 120 of file led.c.

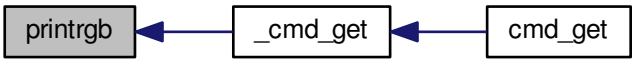
References lighting::b, lighting::g, has_key(), keymap, key::led, N_KEYS_EXTENDED, key::name, and lighting::r.

Referenced by _cmd_get().

```
120                                                     {
121     uchar r[N_KEYS_EXTENDED], g[N_KEYS_EXTENDED], b[
    N_KEYS_EXTENDED];
122     const uchar* mr = light->r;
123     const uchar* mg = light->g;
124     const uchar* mb = light->b;
125     for(int i = 0; i < N_KEYS_EXTENDED; i++){
126         // Translate the key index to an RGB index using the key map
127         int k = keymap[i].led;
128         if(k < 0)
129             continue;
130         r[i] = mr[k];
131         g[i] = mg[k];
132         b[i] = mb[k];
133     }
134     // Make a buffer to track key names and to filter out duplicates
135     char names[N_KEYS_EXTENDED][11];
136     for(int i = 0; i < N_KEYS_EXTENDED; i++){
137         const char* name = keymap[i].name;
138         if(keymap[i].led < 0 || !has_key(name, kb))
139             names[i][0] = 0;
140         else
141             strncpy(names[i], name, 11);
142     }
143     // Check to make sure these aren't all the same color
144     int same = 1;
145     for(int i = 1; i < N_KEYS_EXTENDED; i++){
146         if(!names[i][0])
147             continue;
148         if(r[i] != r[0] || g[i] != g[0] || b[i] != b[0]){
149             same = 0;
150             break;
151         }
152     }
153     // If they are, just output that color
154     if(same){
155         char* buffer = malloc(7);
156         snprintf(buffer, 7, "%02x%02x%02x", r[0], g[0], b[0]);
157         return buffer;
158     }
159     const int BUFFER_LEN = 4096;    // Should be more than enough to fit all keys
160     char* buffer = malloc(BUFFER_LEN);
161     int length = 0;
162     for(int i = 0; i < N_KEYS_EXTENDED; i++){
163         if(!names[i][0])
164             continue;
165         // Print the key name
166         int newlen = 0;
167         snprintf(buffer + length, BUFFER_LEN - length, length == 0 ? "%s%n" : " %s%n", names[i], &newlen);
168         length += newlen;
169         // Look ahead to see if any other keys have this color. If so, print them here as well.
170         uchar kr = r[i], kg = g[i], kb = b[i];
171         for(int j = i + 1; j < N_KEYS_EXTENDED; j++){
172             if(!names[j][0])
173                 continue;
174             if(r[j] != kr || g[j] != kg || b[j] != kb)
175                 continue;
176             snprintf(buffer + length, BUFFER_LEN - length, ",%s%n", names[j], &newlen);
177             length += newlen;
178             // Erase the key's name so it won't get printed later
179             names[j][0] = 0;
180         }
181         // Print the color
182         snprintf(buffer + length, BUFFER_LEN - length, ":%02x%02x%02x%n", kr, kg, kb, &newlen);
183         length += newlen;
184     }
185     return buffer;
186 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.28.1.9 int savergb_kb (usbdevice * *kb,* lighting * *light,* int *mode*)

Definition at line 148 of file led_keyboard.c.

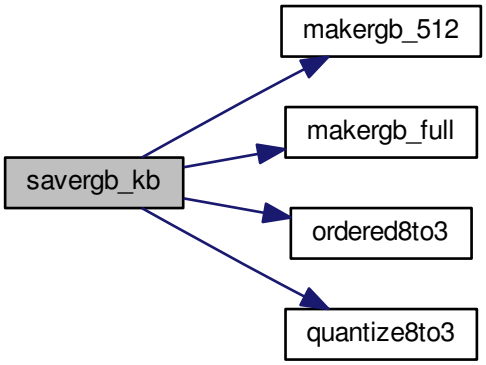
References usbdevice::dither, usbdevice::fwversion, IS_STRAFE, IS_V2_OVERRIDE, makergb_512(), makergb_-full(), MSG_SIZE, ordered8to3(), quantize8to3(), and usbsend.

Referenced by cmd_hwsave_kb().

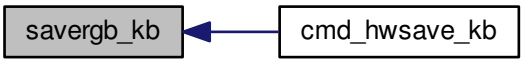
```
148                                                     {
149     if(kb->fwversion >= 0x0120 || IS_V2_OVERRIDE(kb)){
150         uchar data_pkt[12][MSG_SIZE] = {
151             // Red
152             { 0x7f, 0x01, 60, 0 },
153             { 0x7f, 0x02, 60, 0 },
154             { 0x7f, 0x03, 24, 0 },
155             { 0x07, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x01 },
156             // Green
157             { 0x7f, 0x01, 60, 0 },
158             { 0x7f, 0x02, 60, 0 },
159             { 0x7f, 0x03, 24, 0 },
160             { 0x07, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x02 },
161             // Blue
162             { 0x7f, 0x01, 60, 0 },
163             { 0x7f, 0x02, 60, 0 },
164             { 0x7f, 0x03, 24, 0 },
165             { 0x07, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x03 }
166         };
167         makergb_full(light, data_pkt);
168         if(!usbsend(kb, data_pkt[0], 12))
169             return -1;
170         if (IS_STRAFE(kb)){ // end save
171             uchar save_end_pkt[MSG_SIZE] = { 0x07, 0x14, 0x04, 0x01, 0x01 };
172             if(!usbsend(kb, save_end_pkt, 1))
173                 return -1;
174         }
175     } else {
176         uchar data_pkt[5][MSG_SIZE] = {
177             { 0x7f, 0x01, 60, 0 },
178             { 0x7f, 0x02, 60, 0 },
179             { 0x7f, 0x03, 60, 0 },
180             { 0x7f, 0x04, 36, 0 },
181             { 0x07, 0x14, 0x02, 0x00, 0x01, mode + 1 }
182         };
```

```
183         makergb_512(light, data_pkt, kb->dither ? ordered8to3 :
    quantize8to3);
184         if(!usbsend(kb, data_pkt[0], 5))
185             return -1;
186     }
187     return 0;
188 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.28.1.10 int savergb_mouse (usbdevice * *kb*, lighting * *light*, int *mode*)

Definition at line 66 of file led_mouse.c.

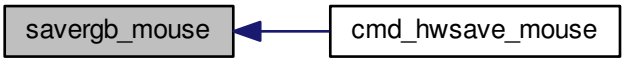
References lighting::b, lighting::g, IS_SABRE, IS_SCIMITAR, LED_DPI, LED_MOUSE, MSG_SIZE, lighting::r, and usbsend.

Referenced by cmd_hwsave_mouse().

```
66                                                      {
67     (void)mode;
68
69     uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x10, 1, 0 };
70     // Save each RGB zone, minus the DPI light which is sent in the DPI packets
71     int zonecount = IS_SCIMITAR(kb) ? 4 : IS_SABRE(kb) ? 3 : 2;
72     for(int i = 0; i < zonecount; i++){
73         int led = LED_MOUSE + i;
74         if(led >= LED_DPI)
75             led++;          // Skip DPI light
76         data_pkt[4] = light->r[led];
```

```
77          data_pkt[5] = light->g[led];
78          data_pkt[6] = light->b[led];
79          if(!usbsend(kb, data_pkt, 1))
80              return -1;
81          // Set packet for next zone
82          data_pkt[2]++;
83      }
84      return 0;
85  }
```

Here is the caller graph for this function:



13.28.1.11 int updatergb_kb (usbdevice * *kb*, int *force*)

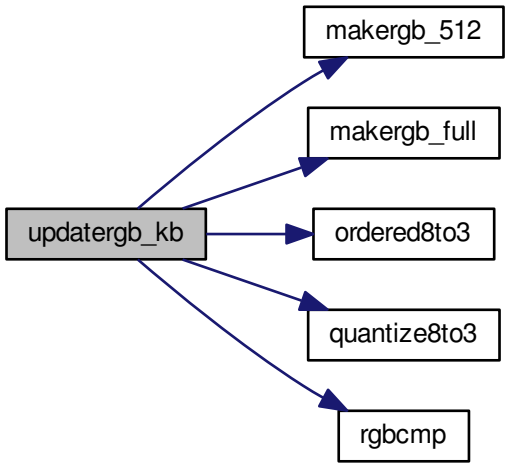
Definition at line 79 of file led_keyboard.c.

References usbdevice::active, usbprofile::currentmode, usbdevice::dither, lighting::forceupdate, IS_FULLRANGE, usbprofile::lastlight, usbmode::light, makergb_512(), makergb_full(), MSG_SIZE, ordered8to3(), P_K95_PLATINUM, usbdevice::product, usbdevice::profile, quantize8to3(), rgbcmp(), lighting::sidelight, and usbsend.

```
79                                          {
80      if(!kb->active)
81          return 0;
82      lighting* lastlight = &kb->profile->lastlight;
83      lighting* newlight = &kb->profile->currentmode->
        light;
84      // Don't do anything if the lighting hasn't changed
85      if(!force && !lastlight->forceupdate && !newlight->forceupdate
86              && !rgbcmp(lastlight, newlight) && lastlight->sidelight == newlight->
        sidelight)   // strafe sidelights
87          return 0;
88      lastlight->forceupdate = newlight->forceupdate = 0;
89
90      if(IS_FULLRANGE(kb)){
91          // Update strafe sidelights if necessary
92          if(lastlight->sidelight != newlight->sidelight) {
93              uchar data_pkt[2][MSG_SIZE] = {
94                   { 0x07, 0x05, 0x08, 0x00, 0x00 },
95                   { 0x07, 0x05, 0x02, 0, 0x03 }
96               };
97               if (newlight->sidelight)
98                   data_pkt[0][4]=1;    // turn on
99               if(!usbsend(kb, data_pkt[0], 2))
100                   return -1;
101          }
102          // 16.8M color lighting works fine on strafe and is the only way it actually works
103          uchar data_pkt[12][MSG_SIZE] = {
104              // Red
105              { 0x7f, 0x01, 0x3c, 0 },
106              { 0x7f, 0x02, 0x3c, 0 },
107              { 0x7f, 0x03, 0x18, 0 },
108              { 0x07, 0x28, 0x01, 0x03, 0x01, 0},
109              // Green
110              { 0x7f, 0x01, 0x3c, 0 },
111              { 0x7f, 0x02, 0x3c, 0 },
112              { 0x7f, 0x03, 0x18, 0 },
113              { 0x07, 0x28, 0x02, 0x03, 0x01, 0},
114              // Blue
115              { 0x7f, 0x01, 0x3c, 0 },
116              { 0x7f, 0x02, 0x3c, 0 },
117              { 0x7f, 0x03, 0x18, 0 },
118              { 0x07, 0x28, 0x03, 0x03, 0x02, 0}
119          };
120          // The K95 Platinum needs 0x30 for the lightbar to work, due to the length of the packet.
```

```
121         // A way to dynamically calculate the length would be preferred, based on the device.
122         if(kb->product == P_K95_PLATINUM){
123             data_pkt[2][2] = 0x30;
124             data_pkt[6][2] = 0x30;
125             data_pkt[10][2] = 0x30;
126         }
127         makergb_full(newlight, data_pkt);
128         if(!usbsend(kb, data_pkt[0], 12))
129             return -1;
130     } else {
131         // On older keyboards it looks flickery and causes lighting glitches, so we don't use it.
132         uchar data_pkt[5][MSG_SIZE] = {
133             { 0x7f, 0x01, 60, 0 },
134             { 0x7f, 0x02, 60, 0 },
135             { 0x7f, 0x03, 60, 0 },
136             { 0x7f, 0x04, 36, 0 },
137             { 0x07, 0x27, 0x00, 0x00, 0xD8 }
138         };
139         makergb_512(newlight, data_pkt, kb->dither ?
      ordered8to3 : quantize8to3);
140         if(!usbsend(kb, data_pkt[0], 5))
141             return -1;
142     }
143
144     memcpy(lastlight, newlight, sizeof(lighting));
145     return 0;
146 }
```

Here is the call graph for this function:



13.28.1.12 int updatergb_mouse (usbdevice * *kb*, int *force*)

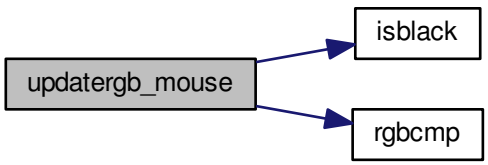
Definition at line 20 of file led_mouse.c.

References usbdevice::active, lighting::b, usbprofile::currentmode, lighting::forceupdate, lighting::g, IS_GLAIVE, isblack(), usbprofile::lastlight, LED_MOUSE, usbmode::light, MSG_SIZE, N_MOUSE_ZONES, usbdevice::profile, lighting::r, rgbcmp(), and usbsend.

```
20                                          {
21     if(!kb->active)
22         return 0;
23     lighting* lastlight = &kb->profile->lastlight;
24     lighting* newlight = &kb->profile->currentmode->
      light;
25     // Don't do anything if the lighting hasn't changed
```

```
26      if(!force && !lastlight->forceupdate && !newlight->forceupdate
27              && !rgbcmp(lastlight, newlight))
28          return 0;
29      lastlight->forceupdate = newlight->forceupdate = 0;
30
31      // Prevent writing to DPI LEDs or non-existent LED zones for the Glaive.
32      int num_zones = IS_GLAIVE(kb) ? 3 : N_MOUSE_ZONES;
33      // Send the RGB values for each zone to the mouse
34      uchar data_pkt[2][MSG_SIZE] = {
35          { 0x07, 0x22, num_zones, 0x01, 0 }, // RGB colors
36          { 0x07, 0x05, 0x02, 0 }                 // Lighting on/off
37      };
38      uchar* rgb_data = &data_pkt[0][4];
39      for(int i = 0; i < N_MOUSE_ZONES; i++){
40          if (IS_GLAIVE(kb) && i != 0 && i != 1 && i != 5)
41          continue;
42          *rgb_data++ = i + 1;
43          *rgb_data++ = newlight->r[LED_MOUSE + i];
44          *rgb_data++ = newlight->g[LED_MOUSE + i];
45          *rgb_data++ = newlight->b[LED_MOUSE + i];
46      }
47      // Send RGB data
48      if(!usbsend(kb, data_pkt[0], 1))
49          return -1;
50      int was_black = isblack(kb, lastlight), is_black = isblack(kb, newlight);
51      if(is_black){
52          // If the lighting is black, send the deactivation packet (M65 only)
53          if(!usbsend(kb, data_pkt[1], 1))
54              return -1;
55      } else if(was_black || force){
56          // If the lighting WAS black, or if we're on forced update, send the activation packet
57          data_pkt[1][4] = 1;
58          if(!usbsend(kb, data_pkt[1], 1))
59              return -1;
60      }
61
62      memcpy(lastlight, newlight, sizeof(lighting));
63      return 0;
64  }
```

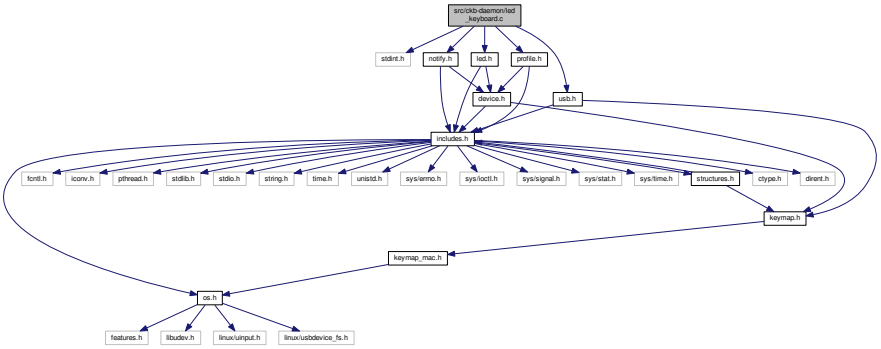
Here is the call graph for this function:



13.29 src/ckb-daemon/led_keyboard.c File Reference

```
#include <stdint.h>
#include "led.h"
#include "notify.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for led_keyboard.c:



Macros

- #define BR1(x) ((((x) & 0xaa) >> 1) | (((x) & 0x55) << 1))
- #define BR2(x) (((BR1(x) & 0xcc) >> 2) | ((BR1(x) & 0x33) << 2))
- #define BR4(x) (((BR2(x) & 0xf0) >> 4) | ((BR2(x) & 0x0f) << 4))
- #define O0(i) BR4(i),
- #define O1(i) O0(i) O0((i) + 1)
- #define O2(i) O1(i) O1((i) + 2)
- #define O3(i) O2(i) O2((i) + 4)
- #define O4(i) O3(i) O3((i) + 8)
- #define O5(i) O4(i) O4((i) + 16)
- #define O6(i) O5(i) O5((i) + 32)
- #define O7(i) O6(i) O6((i) + 64)
- #define O8(i) O7(i) O7((i) + 127)

Functions

- static uchar ordered8to3 (int index, uchar value)
- static uchar quantize8to3 (int index, uchar value)
- static void makergb_512 (const lighting *light, uchar data_pkt[5][64], uchar(*ditherfn)(int, uchar))
- static void makergb_full (const lighting *light, uchar data_pkt[12][64])
- static int rgbcmp (const lighting *lhs, const lighting *rhs)
- int updatergb_kb (usbdevice *kb, int force)
- int savergb_kb (usbdevice *kb, lighting *light, int mode)
- int loadrgb_kb (usbdevice *kb, lighting *light, int mode)

Variables

- static uchar bit_reverse_table [256] = { (((((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 2) + 1) & 0xaa) >> 1) |

((((((0) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 4) & 0xaa) >> 1) | (((((0) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 4) & 0xaa) >> 1) | (((((0) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 4) & 0xaa) >> 1) | (((((0) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 4) & 0xaa) >> 1) | (((((0) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 8) & 0xaa) >> 1) | (((((0) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 8) & 0xaa) >> 1) | (((((0) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 8) & 0xaa) >> 1) | (((((0) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 8) & 0xaa) >> 1) | (((((0) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 2) & 0xaa) >> 1) | (((((0) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 16) & 0xaa) >> 1) | (((((0) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 16) & 0xaa) >> 1) | (((((0) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 16) & 0xaa) >> 1) | (((((0) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 16) & 0xaa) >> 1) | (((((0) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0

) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 16) + 4) & 0xaa) >> 1) | (((((0) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 4) & 0xaa) >> 1) | (((((0) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 4) & 0xaa) >> 1) | (((((0) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 4) & 0xaa) >> 1) | (((((0) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 16) + 8) & 0xaa) >> 1) | (((((0) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 8) & 0xaa) >> 1) | (((((0) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 8) & 0xaa) >> 1) | (((((0) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 8) & 0xaa) >> 1) | (((((0) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 16) + 8) + 4) & 0xaa) >> 1) | (((((0) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 8) + 4) & 0xaa) >> 1) | (((((0) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 8) + 4) & 0xaa) >> 1) | (((((0) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 8) + 4) & 0xaa) >> 1) | (((((0) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) +

16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 32) & 0xaa) >> 1) | ((((0) + 32) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 32) & 0xaa) >> 1) | ((((0) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) + 32) & 0xaa) >> 1) | ((((0) + 32) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 32) & 0xaa) >> 1) | ((((0) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 32) + 1) & 0xaa) >> 1) | (((((0) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 32) + 1) & 0xaa) >> 1) | (((((0) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 32) + 1) & 0xaa) >> 1) | (((((0) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 32) + 1) & 0xaa) >> 1) | (((((0) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 32) + 2) & 0xaa) >> 1) | (((((0) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 32) + 2) & 0xaa) >> 1) | (((((0) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 32) + 2) & 0xaa) >> 1) | (((((0) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 32) + 2) & 0xaa) >> 1) | (((((0) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 4) & 0xaa) >> 1) | (((((0) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 4) & 0xaa) >> 1) | (((((0) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 32) + 4) & 0xaa) >> 1) | (((((0) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 4) & 0xaa) >> 1) | (((((0) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 8) & 0xaa) >> 1) | (((((0) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 8) & 0xaa) >> 1) | (((((0) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 32) + 8) & 0xaa) >> 1) | (((((0) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 8) & 0xaa) >> 1) | (((((0) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 1) & 0x55)

<< 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 32) + 16) & 0xaa) >> 1) | (((((0) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 32) + 16) & 0xaa) >> 1) | (((((0) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 32) + 16) & 0xaa) >> 1) | (((((0) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 32) + 16) & 0xaa) >> 1) | (((((0) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 32) + 16)

+ 8) + 1) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 64) & 0xaa) >> 1) | ((((0) + 64) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) & 0xaa) >> 1) | ((((0) + 64) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) + 64) & 0xaa) >> 1) | ((((0) + 64) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) & 0xaa) >> 1) | ((((0) + 64) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 64) + 1) & 0xaa) >> 1) | ((((0) + 64) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) + 1) & 0xaa) >> 1) | ((((0) + 64) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) + 64) + 1) & 0xaa) >> 1) | ((((0) + 64) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) + 1) & 0xaa) >> 1) | ((((0) + 64) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 64) + 2) & 0xaa) >> 1) | ((((0) + 64) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) + 2) & 0xaa) >> 1) | ((((0) + 64) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) + 64) + 2) & 0xaa) >> 1) | ((((0) + 64) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) + 2) & 0xaa) >> 1) | ((((0) + 64) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 64) + 2) + 1) & 0xaa) >> 1) | ((((0) + 64) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 64) + 2) + 1) & 0xaa) >> 1) | ((((0) + 64) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 64) + 2) + 1) & 0xaa) >> 1) | (((((0) + 64) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 64) + 2) + 1) & 0xaa) >> 1) | (((((0) + 64) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 64) + 4) & 0xaa) >> 1) | ((((0) + 64) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 64) + 4) & 0xaa) >> 1) | ((((0) + 64) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 64) + 4) & 0xaa) >> 1) | ((((0) + 64) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 64) + 4) & 0xaa) >> 1) | ((((0) + 64) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 64) + 4) + 1) & 0xaa) >> 1) | (((((0) + 64) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 64) + 4) + 1) & 0xaa) >> 1) | (((((0) + 64) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 64) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 64) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 4) + 2) & 0x55) << 1)) & 0x33) <<

2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 64) + 8) & 0xaa) >> 1) | (((((0) + 64) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 8) & 0xaa) >> 1) | (((((0) + 64) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 64) + 8) & 0xaa) >> 1) | (((((0) + 64) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 8) & 0xaa) >> 1) | (((((0) + 64) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 64) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 64) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 64) + 16) & 0xaa) >> 1) | (((((0) + 64) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 16) & 0xaa) >> 1) | (((((0) + 64) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 64) + 16) & 0xaa) >> 1) | (((((0) + 64) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 16) & 0xaa) >> 1) | (((((0) + 64) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 2) + 1) & 0x55) << 1))

& 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 16) + 8) + 4)

+ 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 64) + 32) & 0xaa) >> 1) | (((((0) + 64) + 32) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 32) & 0xaa) >> 1) | (((((0) + 64) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 32) & 0xaa) >> 1) | (((((0) + 64) + 32) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 64) + 32) & 0xaa) >> 1) | (((((0) + 64) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 32) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 1) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 32) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 2) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) +

8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((0) + 64) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32

) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 127) & 0xaa) >> 1) | ((((0) + 127) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 127) & 0xaa) >> 1) | ((((0) + 127) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) + 127) & 0xaa) >> 1) | ((((0) + 127) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 127) & 0xaa) >> 1) | ((((0) + 127) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 127) + 1) & 0xaa) >> 1) | ((((0) + 127) + 1) & 0x55) <<

1)) & 0xcc) >> 2) | (((((((((0) + 127) + 1) & 0xaa) >> 1) | (((((0) + 127) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 1) & 0xaa) >> 1) | (((((0) + 127) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 1) & 0xaa) >> 1) | (((((0) + 127) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 2) & 0xaa) >> 1) | (((((0) + 127) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 2) & 0xaa) >> 1) | (((((0) + 127) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 2) & 0xaa) >> 1) | (((((0) + 127) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 2) & 0xaa) >> 1) | (((((0) + 127) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 2) + 1) & 0xaa) >> 1) | (((((0) + 127) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 4) & 0xaa) >> 1) | (((((0) + 127) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 4) & 0xaa) >> 1) | (((((0) + 127) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 4) & 0xaa) >> 1) | (((((0) + 127) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 4) & 0xaa) >> 1) | (((((0) + 127) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 8) & 0xaa) >> 1) | (((((0) + 127) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 8) & 0xaa) >> 1) | (((((0) + 127) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 8) & 0xaa) >> 1) | (((((0) + 127) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 8) & 0xaa) >> 1) | (((((0) + 127) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 8) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 8) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 8) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0)

>> 4) | (((((((((((((0) + 127) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) >> 4) | ((((((((((((((0) + 127) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 16) & 0xaa) >> 1) | (((((0) + 127) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 16) & 0xaa) >> 1) | (((((0) + 127) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 16) & 0xaa) >> 1) | (((((0) + 127) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 16) & 0xaa) >> 1) | (((((0) + 127) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 16) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 16) + 8) + 2) & 0xaa) >>

1) | ((((((((0) + 127) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((((0) + 127) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((((0) + 127) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 127) + 32) & 0xaa) >> 1) | (((((0) + 127) + 32) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 127) + 32) & 0xaa) >> 1) | (((((0) + 127) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) + 127) + 32) & 0xaa) >> 1) | (((((0) + 127) + 32) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 127) + 32) & 0xaa) >> 1) | (((((0) + 127) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 32) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 32) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 32) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 32) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 32) + 4) + 2)

& 0xaa) >> 1) | (((((((0) + 127) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((
((((0) + 127) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 4) + 2) & 0x55) << 1)) &
0xcc) >> 2) | (((((((((((0) + 127) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 4) + 2) &
0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 4) + 2) + 1) & 0xaa) >>
1) | ((((((((0) + 127) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32)
+ 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2))
& 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) +
4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 4) + 2) + 1) & 0xaa) >> 1) |
(((((((((0) + 127) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) +
127) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0)
+ 127) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0)
>> 4) | (((((((((((((0) + 127) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 8) & 0x55) << 1)) &
0xcc) >> 2) | ((((((((((0) + 127) + 32) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 8) & 0x55) << 1))
& 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127
) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 8) + 1) & 0xaa) >> 1) | (((
((((0) + 127) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) +
32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((
((0) + 127) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 8) + 1) & 0x55) << 1)) & 0x33)
<< 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32)
+ 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0
) + 127) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 32) + 8
) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0)
+ 127) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2))
& 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32)
+ 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 8) + 2) + 1) & 0xaa) >> 1)
| ((((((((0) + 127) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0
) + 127) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 2) + 1) & 0x55) << 1)) &
0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8)
+ 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 32) + 8) + 4) & 0xaa)
>> 1) | (((((((0) + 127) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) +
8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >>
4) | ((((((((((((((0) + 127) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 8) + 4) & 0x55)
<< 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) +
8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 8) + 4) + 1) &
0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) +
127) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 4) + 1) & 0x55) << 1)) &
0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0)
+ 127) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 8) + 4) + 1)
& 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)),
((((((((((((((((0) + 127) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 4) + 2)
& 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0)
+ 127) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32
) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >>
2) | ((((((((((((0) + 127) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 8) + 4) + 2
) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 32) + 8) + 4) + 2) + 1) &
0xaa) >> 1) | (((((((((0) + 127) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((
((0) + 127) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 8) + 4) + 2) + 1
) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 32) + 8) + 4) + 2) + 1) &
0xaa) >> 1) | (((((((((0) + 127) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((
(((0) + 127) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 8) + 4) + 2) + 1
) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 32) + 16) & 0xaa) >> 1) | ((((((
0) + 127) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 32) + 16) & 0xaa) >> 1) |
((((((0) + 127) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 32)
+ 16) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127
) + 32) + 16) & 0xaa) >> 1) | ((((((0) + 127) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) <<
4)), (((((((((((((((0) + 127) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 1) & 0x55)
<< 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) +
16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 32) + 16) + 1) & 0xaa)

>> 1) | ((((((((0) + 127) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) >> 4) | ((((((((((((((0) + 127) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) >> 4) | ((((((((((((((0) + 127) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 32) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 4) & 0x55) << 1))

& 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) & 0xaa) >> 1) | (((((((0) + 127) + 64) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) & 0xaa) >> 1) | (((((((0) + 127) + 64) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) & 0xaa) >> 1) | (((((((0) + 127) + 64) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) & 0xaa) >> 1) | (((((((0) + 127) + 64) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 1) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 64) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 64) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 2) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 64) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 64) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) +

127) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 64) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 64) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 8) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 8) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 8) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 127) + 64) + 16) & 0xaa) >> 1) | (((((0) + 127) + 64) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 64) + 16) & 0xaa) >> 1) | (((((0) + 127) + 64) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 127) + 64) + 16) & 0xaa) >> 1) | (((((0) + 127) + 64) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 127) + 64) + 16) & 0xaa) >> 1) | (((((0) + 127) + 64) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 16) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 16) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 16) + 2) & 0xaa) >> 1) | (((((((0

+ 127) + 64) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 16) + 8) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 1)

& 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 64) + 32) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 32) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 32) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 127) + 64) + 32) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 32) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 127) + 64) + 32) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 32) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 32) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 32) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 4) & 0xaa) >> 1) | ((((((0) + 127) + 64) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 32) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 4) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 4) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 4) + 2) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0xaa) >> 1) | (((

((((((0) + 127) + 64) + 32) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 64) + 32) + 8) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 32) + 8) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 8) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 127) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 16) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0) + 127) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((0) + 127) + 64) + 32) + 16) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((0)

+ 127) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 2) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 4) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 8) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | ((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((

((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0xaa) >> 1) | (((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((((((0) + 127) + 64) + 32) + 16) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), }

13.29.1 Macro Definition Documentation

13.29.1.1 #define BR1(x) ((((x) & 0xaa) >> 1) | (((x) & 0x55) << 1))

Definition at line 9 of file led_keyboard.c.

13.29.1.2 #define BR2(x) (((BR1(x) & 0xcc) >> 2) | ((BR1(x) & 0x33) << 2))

Definition at line 10 of file led_keyboard.c.

13.29.1.3 #define BR4(x) (((BR2(x) & 0xf0) >> 4) | ((BR2(x) & 0x0f) << 4))

Definition at line 11 of file led_keyboard.c.

13.29.1.4 #define O0(*i*) BR4(i),

Definition at line 12 of file led_keyboard.c.

13.29.1.5 #define O1(*i*) O0(i) O0((i) + 1)

Definition at line 13 of file led_keyboard.c.

13.29.1.6 #define O2(*i*) O1(i) O1((i) + 2)

Definition at line 14 of file led_keyboard.c.

13.29.1.7 #define O3(*i*) O2(i) O2((i) + 4)

Definition at line 15 of file led_keyboard.c.

13.29.1.8 #define O4(*i*) O3(i) O3((i) + 8)

Definition at line 16 of file led_keyboard.c.

13.29.1.9 #define O5(*i*) O4(i) O4((i) + 16)

Definition at line 17 of file led_keyboard.c.

13.29.1.10 #define O6(*i*) O5(i) O5((i) + 32)

Definition at line 18 of file led_keyboard.c.

13.29.1.11 #define O7(*i*) O6(i) O6((i) + 64)

Definition at line 19 of file led_keyboard.c.

13.29.1.12 #define O8(*i*) O7(i) O7((i) + 127)

Definition at line 20 of file led_keyboard.c.

13.29.2 Function Documentation

13.29.2.1 int loadrgb_kb (usbdevice * *kb,* lighting * *light,* int *mode*)

Since Firmware Version 2.05 for K95RGB the answers for getting the stored color-maps from the hardware has changed a bit. So comparing for the correct answer cannot validate against the cmd, and has to be done against a third map. Up to now we know, that K70RGB Pro and K70 Lux RGB have firmware version 2.04 and having the problem also. So we have to determine in the most inner loop the firmware version and type of KB to select the correct compare-table.

Read colors

```
                                                    < That is the old comparison method: you get back what you sent.
```

Normally a firmware version >= 2.05 runs with the new compare array. Up to now there is a 2.04 running in K70 RGB Lux with the same behavior. It seems that K70RGB has the same problem

Definition at line 190 of file led_keyboard.c.

References lighting::b, ckb_err, usbdevice::fwversion, lighting::g, IS_V2_OVERRIDE, MSG_SIZE, N_KEYS_HW, P_K70_LUX, P_K70_LUX_NRGB, usbdevice::product, lighting::r, usbrecv, and usbsend.

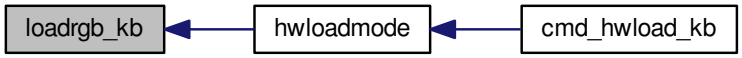
Referenced by hwloadmode().

```
190                                                     {
191     if(kb->fwversion >= 0x0120 || IS_V2_OVERRIDE(kb)){
192         uchar data_pkt[12][MSG_SIZE] = {
193             { 0x0e, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x01 },
```

```
194             { 0xff, 0x01, 60, 0 },
195             { 0xff, 0x02, 60, 0 },
196             { 0xff, 0x03, 24, 0 },
197             { 0x0e, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x02 },
198             { 0xff, 0x01, 60, 0 },
199             { 0xff, 0x02, 60, 0 },
200             { 0xff, 0x03, 24, 0 },
201             { 0x0e, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x03 },
202             { 0xff, 0x01, 60, 0 },
203             { 0xff, 0x02, 60, 0 },
204             { 0xff, 0x03, 24, 0 },
205         };
206         uchar in_pkt[4][MSG_SIZE] = {
207             { 0x0e, 0x14, 0x03, 0x01 },
208             { 0xff, 0x01, 60, 0 },
209             { 0xff, 0x02, 60, 0 },
210             { 0xff, 0x03, 24, 0 },
211         };
212
213
214         uchar cmp_pkt[4][4] = {
215             { 0x0e, 0x14, 0x03, 0x01 },
216             { 0x0e, 0xff, 0x01, 60 },
217             { 0x0e, 0xff, 0x02, 60 },
218             { 0x0e, 0xff, 0x03, 24 },
219         };
220         uchar* colors[3] = { light->r, light->g, light->b };
221         for(int clr = 0; clr < 3; clr++){
222             for(int i = 0; i < 4; i++){
223                 if(!usbrecv(kb, data_pkt[i + clr * 4], in_pkt[i]))
224                     return -1;
225
226                 uchar* comparePacket = data_pkt[i + clr * 4];
227                 if ((kb->fwversion >= 0x205)
228                         || ((kb->fwversion >= 0x204)
229                             && ((kb->product == P_K70_LUX_NRGB) || (kb->
    product == P_K70_LUX)))) {
230                     comparePacket = cmp_pkt[i];
231                 }
232
233                 if (memcmp(in_pkt[i], comparePacket, 4)) {
234                     ckb_err("Bad input header\n");
235                     ckb_err("color = %d, i = %d, mode = %d\nOutput (Request): %2.2x %2.2x %2.2x
     %2.2x\nInput(Reply): %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x %2.2x\n", clr, i, mode,
236                         comparePacket[0], comparePacket[1], comparePacket[2], comparePacket[3],
237                         in_pkt[i][0], in_pkt[i][1], in_pkt[i][2], in_pkt[i][3], in_pkt[i][4], in_pkt[i][5],
     in_pkt[i][6], in_pkt[i][7]);
238                     in_pkt[2][0] = 0x99;
239                     in_pkt[2][1] = 0x99;
240                     in_pkt[2][2] = 0x99;
241                     in_pkt[2][3] = 0x99;
242                     usbrecv(kb, in_pkt[2], in_pkt[2]); // just to find it in the wireshark log
243                     return -1;
244                 }
245             }
246             // Copy colors to lighting. in_pkt[0] is irrelevant.
247             memcpy(colors[clr], in_pkt[1] + 4, 60);
248             memcpy(colors[clr] + 60, in_pkt[2] + 4, 60);
249             memcpy(colors[clr] + 120, in_pkt[3] + 4, 24);
250         }
251     } else {
252         uchar data_pkt[5][MSG_SIZE] = {
253             { 0x0e, 0x14, 0x02, 0x01, 0x01, mode + 1, 0 },
254             { 0xff, 0x01, 60, 0 },
255             { 0xff, 0x02, 60, 0 },
256             { 0xff, 0x03, 60, 0 },
257             { 0xff, 0x04, 36, 0 },
258         };
259         uchar in_pkt[4][MSG_SIZE] = {
260             { 0xff, 0x01, 60, 0 },
261             { 0xff, 0x02, 60, 0 },
262             { 0xff, 0x03, 60, 0 },
263             { 0xff, 0x04, 36, 0 },
264         };
265         // Write initial packet
266         if(!usbsend(kb, data_pkt[0], 1))
267             return -1;
268         // Read colors
269         for(int i = 1; i < 5; i++){
270             if(!usbrecv(kb, data_pkt[i],in_pkt[i - 1]))
271                 return -1;
272             if(memcmp(in_pkt[i - 1], data_pkt[i], 4)){
273                 ckb_err("Bad input header\n");
274                 return -1;
275             }
276         }
277         // Copy the data back to the mode
```

```
287         uint8_t mr[N_KEYS_HW / 2], mg[N_KEYS_HW / 2], mb[
    N_KEYS_HW / 2];
288         memcpy(mr,      in_pkt[0] +  4, 60);
289         memcpy(mr + 60, in_pkt[1] +  4, 12);
290         memcpy(mg,      in_pkt[1] + 16, 48);
291         memcpy(mg + 48, in_pkt[2] +  4, 24);
292         memcpy(mb,      in_pkt[2] + 28, 36);
293         memcpy(mb + 36, in_pkt[3] +  4, 36);
294         // Unpack LED data to 8bpc format
295         for(int i = 0; i < N_KEYS_HW; i++){
296             int     i_2 = i / 2;
297             uint8_t r, g, b;
298 
299             // 3-bit intensities stored in alternate nybbles.
300             if (i & 1) {
301                 r = 7 - (mr[i_2] >> 4);
302                 g = 7 - (mg[i_2] >> 4);
303                 b = 7 - (mb[i_2] >> 4);
304             } else {
305                 r = 7 - (mr[i_2] & 0x0F);
306                 g = 7 - (mg[i_2] & 0x0F);
307                 b = 7 - (mb[i_2] & 0x0F);
308             }
309             // Scale 3-bit values up to 8 bits.
310             light->r[i] = r << 5 | r << 2 | r >> 1;
311             light->g[i] = g << 5 | g << 2 | g >> 1;
312             light->b[i] = b << 5 | b << 2 | b >> 1;
313         }
314     }
315     return 0;
316 }
```

Here is the caller graph for this function:



13.29.2.2 static void makergb_512 (const lighting * *light*, uchar *data_pkt[5][64]*, uchar(*)(int, uchar) *ditherfn*) [static]

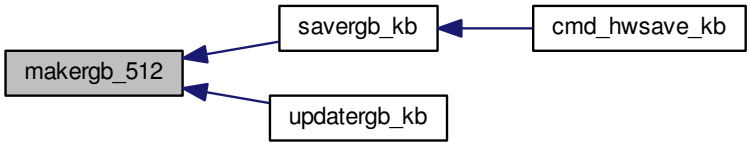
Definition at line 38 of file led_keyboard.c.

References lighting::b, lighting::g, N_KEYS_HW, and lighting::r.

Referenced by savergb_kb(), and updatergb_kb().

```
39                                                          {
40     uchar r[N_KEYS_HW / 2], g[N_KEYS_HW / 2], b[N_KEYS_HW / 2];
41     // Compress RGB values to a 512-color palette
42     for(int i = 0; i < N_KEYS_HW; i += 2){
43         char r1 = ditherfn(i, light->r[i]), r2 = ditherfn(i + 1, light->r[i + 1]);
44         char g1 = ditherfn(i, light->g[i]), g2 = ditherfn(i + 1, light->g[i + 1]);
45         char b1 = ditherfn(i, light->b[i]), b2 = ditherfn(i + 1, light->b[i + 1]);
46         r[i / 2] = (7 - r2) << 4 | (7 - r1);
47         g[i / 2] = (7 - g2) << 4 | (7 - g1);
48         b[i / 2] = (7 - b2) << 4 | (7 - b1);
49     }
50     memcpy(data_pkt[0] + 4, r, 60);
51     memcpy(data_pkt[1] + 4, r + 60, 12);
52     memcpy(data_pkt[1] + 16, g, 48);
53     memcpy(data_pkt[2] + 4, g + 48, 24);
54     memcpy(data_pkt[2] + 28, b, 36);
55     memcpy(data_pkt[3] + 4, b + 36, 36);
56 }
```

Here is the caller graph for this function:



13.29.2.3 static void makergb_full (const lighting * *light*, uchar *data_pkt[12][64]*) [static]

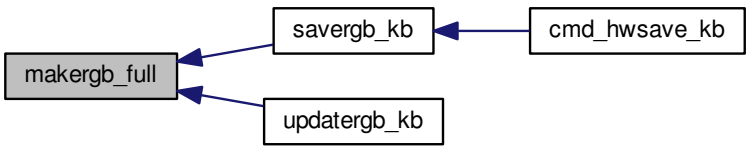
Definition at line 58 of file led_keyboard.c.

References lighting::b, lighting::g, and lighting::r.

Referenced by savergb_kb(), and updatergb_kb().

```
58                                                                                  {
59     const uchar* r = light->r, *g = light->g, *b = light->b;
60     // Red
61     memcpy(data_pkt[0] + 4, r, 60);
62     memcpy(data_pkt[1] + 4, r + 60, 60);
63     memcpy(data_pkt[2] + 4, r + 120, 60);
64     // Green (final R packet is blank)
65     memcpy(data_pkt[4] + 4, g, 60);
66     memcpy(data_pkt[5] + 4, g + 60, 60);
67     memcpy(data_pkt[6] + 4, g + 120, 60);
68     // Blue (final G packet is blank)
69     memcpy(data_pkt[8] + 4, b, 60);
70     memcpy(data_pkt[9] + 4, b + 60, 60);
71     memcpy(data_pkt[10] + 4, b + 120, 60);
72 }
```

Here is the caller graph for this function:



13.29.2.4 static uchar ordered8to3 (int *index*, uchar *value*) [static]

Definition at line 24 of file led_keyboard.c.

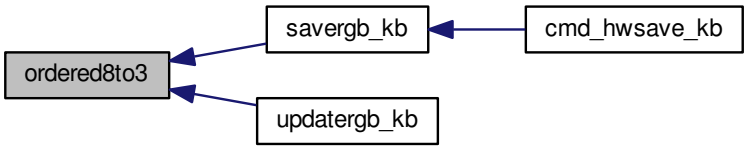
References bit_reverse_table.

Referenced by savergb_kb(), and updatergb_kb().

```
24                                                  {
```

```
25     int m = value * 7;
26     int b = m / 255;
27     if((m % 255) > bit_reverse_table[index & 0xff])
28         b++;
29     return b;
30 }
```

Here is the caller graph for this function:



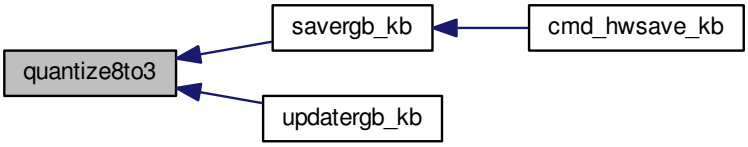
13.29.2.5 static uchar quantize8to3 (int *index*, uchar *value*) [static]

Definition at line 32 of file led_keyboard.c.

Referenced by savergb_kb(), and updatergb_kb().

```
32                                                {
33     (void)index;
34
35     return value >> 5;
36 }
```

Here is the caller graph for this function:



13.29.2.6 static int rgbcmp (const lighting * *lhs*, const lighting * *rhs*) [static]

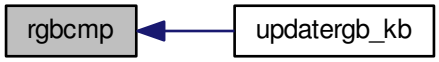
Definition at line 74 of file led_keyboard.c.

References lighting::b, lighting::g, N_KEYS_HW, and lighting::r.

Referenced by updatergb_kb().

```
74                                                      {
75     // Compare two light structures, ignore mouse zones
76     return memcmp(lhs->r, rhs->r, N_KEYS_HW) || memcmp(lhs->g, rhs->
       g, N_KEYS_HW) || memcmp(lhs->b, rhs->b, N_KEYS_HW);
77 }
```

Here is the caller graph for this function:



13.29.2.7 int savergb_kb (usbdevice * *kb*, lighting * *light*, int *mode*)

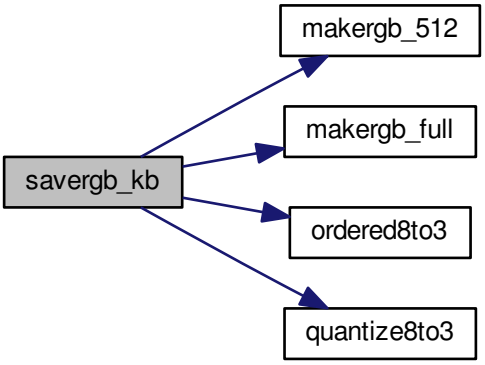
Definition at line 148 of file led_keyboard.c.

References usbdevice::dither, usbdevice::fwversion, IS_STRAFE, IS_V2_OVERRIDE, makergb_512(), makergb_full(), MSG_SIZE, ordered8to3(), quantize8to3(), and usbsend.

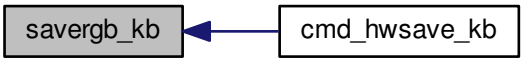
Referenced by cmd_hwsave_kb().

```
148                                                 {
149     if(kb->fwversion >= 0x0120 || IS_V2_OVERRIDE(kb)){
150         uchar data_pkt[12][MSG_SIZE] = {
151             // Red
152             { 0x7f, 0x01, 60, 0 },
153             { 0x7f, 0x02, 60, 0 },
154             { 0x7f, 0x03, 24, 0 },
155             { 0x07, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x01 },
156             // Green
157             { 0x7f, 0x01, 60, 0 },
158             { 0x7f, 0x02, 60, 0 },
159             { 0x7f, 0x03, 24, 0 },
160             { 0x07, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x02 },
161             // Blue
162             { 0x7f, 0x01, 60, 0 },
163             { 0x7f, 0x02, 60, 0 },
164             { 0x7f, 0x03, 24, 0 },
165             { 0x07, 0x14, 0x03, 0x01, 0x01, mode + 1, 0x03 }
166         };
167         makergb_full(light, data_pkt);
168         if(!usbsend(kb, data_pkt[0], 12))
169             return -1;
170         if (IS_STRAFE(kb)){ // end save
171             uchar save_end_pkt[MSG_SIZE] = { 0x07, 0x14, 0x04, 0x01, 0x01 };
172             if(!usbsend(kb, save_end_pkt, 1))
173                 return -1;
174         }
175     } else {
176         uchar data_pkt[5][MSG_SIZE] = {
177             { 0x7f, 0x01, 60, 0 },
178             { 0x7f, 0x02, 60, 0 },
179             { 0x7f, 0x03, 60, 0 },
180             { 0x7f, 0x04, 36, 0 },
181             { 0x07, 0x14, 0x02, 0x00, 0x01, mode + 1 }
182         };
183         makergb_512(light, data_pkt, kb->dither ? ordered8to3 :
    quantize8to3);
184         if(!usbsend(kb, data_pkt[0], 5))
185             return -1;
186     }
187     return 0;
188 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.29.2.8 int updatergb_kb (usbdevice * *kb*, int *force*)

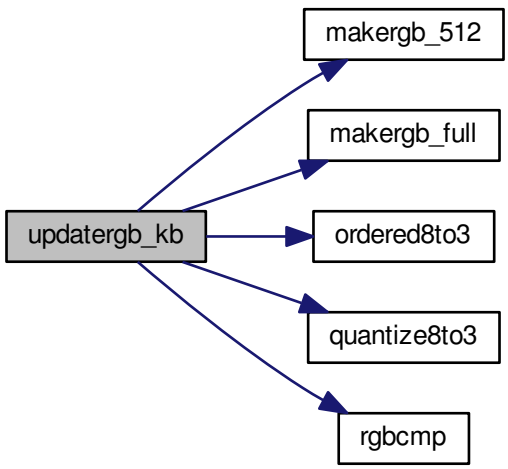
Definition at line 79 of file led_keyboard.c.

References usbdevice::active, usbprofile::currentmode, usbdevice::dither, lighting::forceupdate, IS_FULLRANGE, usbprofile::lastlight, usbmode::light, makergb_512(), makergb_full(), MSG_SIZE, ordered8to3(), P_K95_PLATINUM, usbdevice::product, usbdevice::profile, quantize8to3(), rgbcmp(), lighting::sidelight, and usbsend.

```
79                                          {
80     if(!kb->active)
81         return 0;
82     lighting* lastlight = &kb->profile->lastlight;
83     lighting* newlight = &kb->profile->currentmode->
       light;
84     // Don't do anything if the lighting hasn't changed
85     if(!force && !lastlight->forceupdate && !newlight->forceupdate
86             && !rgbcmp(lastlight, newlight) && lastlight->sidelight == newlight->
       sidelight)   // strafe sidelights
87         return 0;
88     lastlight->forceupdate = newlight->forceupdate = 0;
89
90     if(IS_FULLRANGE(kb)){
91         // Update strafe sidelights if necessary
92         if(lastlight->sidelight != newlight->sidelight) {
93             uchar data_pkt[2][MSG_SIZE] = {
94                 { 0x07, 0x05, 0x08, 0x00, 0x00 },
95                 { 0x07, 0x05, 0x02, 0, 0x03 }
96              };
97              if (newlight->sidelight)
```

```
98                  data_pkt[0][4]=1;    // turn on
99              if(!usbsend(kb, data_pkt[0], 2))
100                  return -1;
101         }
102         // 16.8M color lighting works fine on strafe and is the only way it actually works
103         uchar data_pkt[12][MSG_SIZE] = {
104             // Red
105             { 0x7f, 0x01, 0x3c, 0 },
106             { 0x7f, 0x02, 0x3c, 0 },
107             { 0x7f, 0x03, 0x18, 0 },
108             { 0x07, 0x28, 0x01, 0x03, 0x01, 0},
109             // Green
110             { 0x7f, 0x01, 0x3c, 0 },
111             { 0x7f, 0x02, 0x3c, 0 },
112             { 0x7f, 0x03, 0x18, 0 },
113             { 0x07, 0x28, 0x02, 0x03, 0x01, 0},
114             // Blue
115             { 0x7f, 0x01, 0x3c, 0 },
116             { 0x7f, 0x02, 0x3c, 0 },
117             { 0x7f, 0x03, 0x18, 0 },
118             { 0x07, 0x28, 0x03, 0x03, 0x02, 0}
119         };
120         // The K95 Platinum needs 0x30 for the lightbar to work, due to the length of the packet.
121         // A way to dynamically calculate the length would be preferred, based on the device.
122         if(kb->product == P_K95_PLATINUM){
123             data_pkt[2][2] = 0x30;
124             data_pkt[6][2] = 0x30;
125             data_pkt[10][2] = 0x30;
126         }
127         makergb_full(newlight, data_pkt);
128         if(!usbsend(kb, data_pkt[0], 12))
129             return -1;
130     } else {
131         // On older keyboards it looks flickery and causes lighting glitches, so we don't use it.
132         uchar data_pkt[5][MSG_SIZE] = {
133             { 0x7f, 0x01, 60, 0 },
134             { 0x7f, 0x02, 60, 0 },
135             { 0x7f, 0x03, 60, 0 },
136             { 0x7f, 0x04, 36, 0 },
137             { 0x07, 0x27, 0x00, 0x00, 0xD8 }
138         };
139         makergb_512(newlight, data_pkt, kb->dither ?
    ordered8to3 : quantize8to3);
140         if(!usbsend(kb, data_pkt[0], 5))
141             return -1;
142     }
143
144     memcpy(lastlight, newlight, sizeof(lighting));
145     return 0;
146 }
```

Here is the call graph for this function:



13.29.3 Variable Documentation

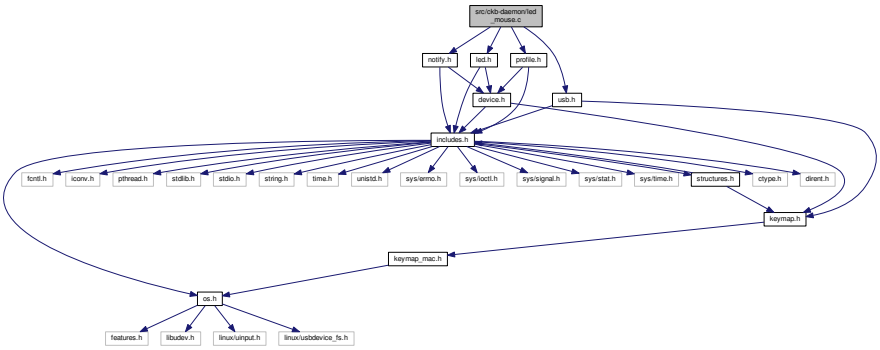
13.29.3.1 uchar bit_reverse_table[256] = { ((((((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0xcc) >> 2) | ((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0xcc) >> 2) | ((((((0) & 0xaa) >> 1) | (((0) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 1) & 0xaa) >> 1) | ((((0) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 2) & 0xaa) >> 1) | ((((0) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 2) + 1) & 0xaa) >> 1) | (((((0) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 2) + 1) & 0xaa) >> 1) | (((((0) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 2) + 1) & 0xaa) >> 1) | (((((0) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 2) + 1) & 0xaa) >> 1) | (((((0) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 4) & 0xaa) >> 1) | ((((0) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 4) & 0xaa) >> 1) | ((((0) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 4) & 0xaa) >> 1) | ((((0) + 4) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 4) & 0xaa) >> 1) | ((((0) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 4) + 1) & 0xaa) >> 1) | (((((0) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 1) & 0xaa) >> 1) | (((((0) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 4) + 1) & 0xaa) >> 1) | (((((0) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 1) & 0xaa) >> 1) | (((((0) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 4) + 2) & 0xaa) >> 1) | (((((0) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 2) & 0xaa) >> 1) | (((((0) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 4) + 2) & 0xaa) >> 1) | (((((0) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 4) + 2) & 0xaa) >> 1) | (((((0) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 4) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 8) & 0xaa) >> 1) | ((((0) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 8) & 0xaa) >> 1) | ((((0) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 8) & 0xaa) >> 1) | ((((0) + 8) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 8) & 0xaa) >> 1) | ((((0) + 8) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 8) + 1) & 0xaa) >> 1) | (((((0) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 1) & 0xaa) >> 1) | (((((0) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 8) + 1) & 0xaa) >> 1) | (((((0) + 8) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 1) & 0xaa) >> 1) | (((((0) + 8) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 8) + 2) & 0xaa) >> 1) | (((((0) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 2) & 0xaa) >> 1) | (((((0) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 8) + 2) & 0xaa) >> 1) | (((((0) + 8) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 2) & 0xaa) >> 1) | (((((0) + 8) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 8) + 4) & 0xaa) >> 1) | (((((0) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 4) & 0xaa) >> 1) | (((((0) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 8) + 4) & 0xaa) >> 1) | (((((0) + 8) + 4) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 8) + 4) & 0xaa) >> 1) | (((((0) + 8) + 4) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 1) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 8) + 4) + 2) & 0xaa) >> 1) | ((((((0) + 8) + 4) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((((0) + 8) + 4) + 2) + 1) & 0xaa) >> 1) | (((((((0) + 8) + 4) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((0) + 16) & 0xaa) >> 1) | ((((0) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 16) & 0xaa) >> 1) | ((((0) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((0) + 16) & 0xaa) >> 1) | ((((0) + 16) & 0x55) << 1)) & 0xcc) >> 2) | (((((((0) + 16) & 0xaa) >> 1) | ((((0) + 16) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 16) + 1) & 0xaa) >> 1) | (((((0) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 16) + 1) & 0xaa) >> 1) | (((((0) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 1) & 0xaa) >> 1) | (((((0) + 16) + 1) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 16) + 1) & 0xaa) >> 1) | (((((0) + 16) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), ((((((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | (((((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0xcc) >> 2) | ((((((((0) + 16) + 2) & 0xaa) >> 1) | (((((0) + 16) + 2) & 0x55) << 1)) & 0x33) << 2)) & 0x0f) << 4)), (((((((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2) + 1) & 0x55) << 1)) & 0xcc) >> 2) | (((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2) + 1) & 0x55) << 1)) & 0x33) << 2)) & 0xf0) >> 4) | ((((((((((((0) + 16) + 2) + 1) & 0xaa) >> 1) | ((((((0) + 16) + 2

Referenced by ordered8to3().

13.30 src/ckb-daemon/led_mouse.c File Reference

```
#include "led.h"
#include "notify.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for led_mouse.c:



Functions

- static int rgbcmp (const lighting *lhs, const lighting *rhs)
- static int isblack (const usbdevice *kb, const lighting *light)
- int updatergb_mouse (usbdevice *kb, int force)
- int savergb_mouse (usbdevice *kb, lighting *light, int mode)
- int loadrgb_mouse (usbdevice *kb, lighting *light, int mode)

13.30.1 Function Documentation

13.30.1.1 static int isblack (const usbdevice * *kb*, const lighting * *light*) [static]

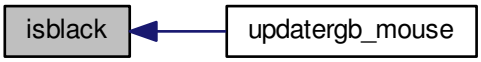
Definition at line 13 of file led_mouse.c.

References lighting::b, lighting::g, IS_M65, LED_MOUSE, N_MOUSE_ZONES, and lighting::r.

Referenced by updatergb_mouse().

```
13                                                          {
14     if(!IS_M65(kb))
15         return 0;
16     uchar black[N_MOUSE_ZONES] = { 0 };
17     return !memcmp(light->r + LED_MOUSE, black, sizeof(black)) && !memcmp(light->
       g + LED_MOUSE, black, sizeof(black)) && !memcmp(light->b + LED_MOUSE, black, sizeof(
       black));
18 }
```

Here is the caller graph for this function:



13.30.1.2 int loadrgb_mouse (usbdevice * *kb*, lighting * *light*, int *mode*)

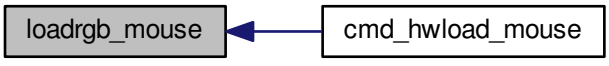
Definition at line 87 of file led_mouse.c.

References lighting::b, ckb_err, lighting::g, IS_SABRE, IS_SCIMITAR, LED_DPI, LED_MOUSE, MSG_SIZE, lighting::r, and usbrecv.

Referenced by cmd_hwload_mouse().

```
87                                                                  {
88      (void)mode;
89
90      uchar data_pkt[MSG_SIZE] = { 0x0e, 0x13, 0x10, 1, 0 };
91      uchar in_pkt[MSG_SIZE] = { 0 };
92      // Load each RGB zone
93      int zonecount = IS_SCIMITAR(kb) ? 4 : IS_SABRE(kb) ? 3 : 2;
94      for(int i = 0; i < zonecount; i++){
95          if(!usbrecv(kb, data_pkt, in_pkt))
96              return -1;
97          if(memcmp(in_pkt, data_pkt, 4)){
98              ckb_err("Bad input header\n");
99              return -2;
100         }
101         // Copy data
102         int led = LED_MOUSE + i;
103         if(led >= LED_DPI)
104             led++;          // Skip DPI light
105         light->r[led] = in_pkt[4];
106         light->g[led] = in_pkt[5];
107         light->b[led] = in_pkt[6];
108         // Set packet for next zone
109         data_pkt[2]++;
110     }
111     return 0;
112 }
```

Here is the caller graph for this function:



13.30.1.3 static int rgbcmp (const lighting * *lhs*, const lighting * *rhs*) [static]

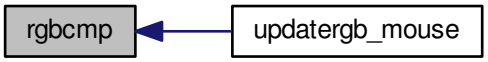
Definition at line 7 of file led_mouse.c.

References lighting::b, lighting::g, LED_MOUSE, N_MOUSE_ZONES, and lighting::r.

Referenced by updatergb_mouse().

```
7                                                                   {
8       return memcmp(lhs->r + LED_MOUSE, rhs->r + LED_MOUSE,
        N_MOUSE_ZONES) || memcmp(lhs->g + LED_MOUSE, rhs->g +
        LED_MOUSE, N_MOUSE_ZONES) || memcmp(lhs->b + LED_MOUSE, rhs->
        b + LED_MOUSE, N_MOUSE_ZONES);
9 }
```

Here is the caller graph for this function:



13.30.1.4 int savergb_mouse (usbdevice * *kb*, lighting * *light*, int *mode*)

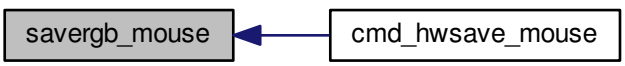
Definition at line 66 of file led_mouse.c.

References lighting::b, lighting::g, IS_SABRE, IS_SCIMITAR, LED_DPI, LED_MOUSE, MSG_SIZE, lighting::r, and usbsend.

Referenced by cmd_hwsave_mouse().

```
66                                                                  {
67      (void)mode;
68
69      uchar data_pkt[MSG_SIZE] = { 0x07, 0x13, 0x10, 1, 0 };
70      // Save each RGB zone, minus the DPI light which is sent in the DPI packets
71      int zonecount = IS_SCIMITAR(kb) ? 4 : IS_SABRE(kb) ? 3 : 2;
72      for(int i = 0; i < zonecount; i++){
73          int led = LED_MOUSE + i;
74          if(led >= LED_DPI)
75              led++;          // Skip DPI light
76          data_pkt[4] = light->r[led];
77          data_pkt[5] = light->g[led];
78          data_pkt[6] = light->b[led];
79          if(!usbsend(kb, data_pkt, 1))
80              return -1;
81          // Set packet for next zone
82          data_pkt[2]++;
83      }
84      return 0;
85 }
```

Here is the caller graph for this function:



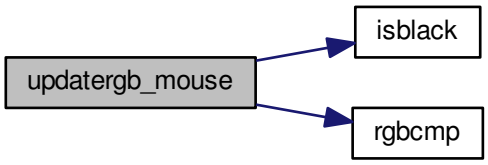
13.30.1.5 int updatergb_mouse (usbdevice * *kb*, int *force*)

Definition at line 20 of file led_mouse.c.

References usbdevice::active, lighting::b, usbprofile::currentmode, lighting::forceupdate, lighting::g, IS_GLAIVE, isblack(), usbprofile::lastlight, LED_MOUSE, usbmode::light, MSG_SIZE, N_MOUSE_ZONES, usbdevice::profile, lighting::r, rgbcmp(), and usbsend.

```
20                                              {
21     if(!kb->active)
22         return 0;
23     lighting* lastlight = &kb->profile->lastlight;
24     lighting* newlight = &kb->profile->currentmode->
       light;
25     // Don't do anything if the lighting hasn't changed
26     if(!force && !lastlight->forceupdate && !newlight->forceupdate
27             && !rgbcmp(lastlight, newlight))
28         return 0;
29     lastlight->forceupdate = newlight->forceupdate = 0;
30
31     // Prevent writing to DPI LEDs or non-existent LED zones for the Glaive.
32     int num_zones = IS_GLAIVE(kb) ? 3 : N_MOUSE_ZONES;
33     // Send the RGB values for each zone to the mouse
34     uchar data_pkt[2][MSG_SIZE] = {
35         { 0x07, 0x22, num_zones, 0x01, 0 }, // RGB colors
36         { 0x07, 0x05, 0x02, 0 }                 // Lighting on/off
37     };
38     uchar* rgb_data = &data_pkt[0][4];
39     for(int i = 0; i < N_MOUSE_ZONES; i++){
40         if (IS_GLAIVE(kb) && i != 0 && i != 1 && i != 5)
41         continue;
42         *rgb_data++ = i + 1;
43         *rgb_data++ = newlight->r[LED_MOUSE + i];
44         *rgb_data++ = newlight->g[LED_MOUSE + i];
45         *rgb_data++ = newlight->b[LED_MOUSE + i];
46     }
47     // Send RGB data
48     if(!usbsend(kb, data_pkt[0], 1))
49         return -1;
50     int was_black = isblack(kb, lastlight), is_black = isblack(kb, newlight);
51     if(is_black){
52         // If the lighting is black, send the deactivation packet (M65 only)
53         if(!usbsend(kb, data_pkt[1], 1))
54             return -1;
55     } else if(was_black || force){
56         // If the lighting WAS black, or if we're on forced update, send the activation packet
57         data_pkt[1][4] = 1;
58         if(!usbsend(kb, data_pkt[1], 1))
59             return -1;
60     }
61
62     memcpy(lastlight, newlight, sizeof(lighting));
63     return 0;
64 }
```

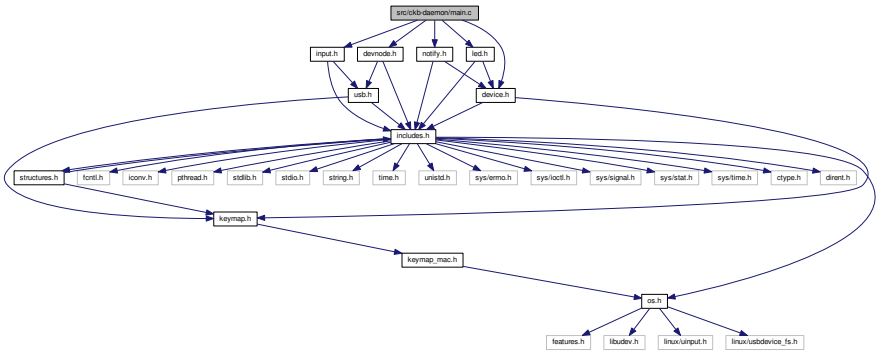
Here is the call graph for this function:



13.31 src/ckb-daemon/main.c File Reference

```
#include "device.h"
#include "devnode.h"
#include "input.h"
#include "led.h"
#include "notify.h"
```

Include dependency graph for main.c:



Functions

- static void quitWithLock (char mut)

  quitWithLock
- int restart ()
- void timespec_add (struct timespec *timespec, long nanoseconds)
- static void quit ()

  quit Stop working the daemon. function is called if the daemon received a sigterm In this case, locking the device-mutex is ok.
- void sighandler2 (int type)
- void sighandler (int type)
- void localecase (char *dst, size_t length, const char *src)
- int main (int argc, char **argv)

Variables

- static int main_ac
- static char ** main_av
- volatile int reset_stop

  brief .
- int features_mask

  brief .
- int hwload_mode

  hwload_mode = 1 means read hardware once. should be enough

13.31.1 Function Documentation

13.31.1.1 void localecase (char * *dst*, size_t *length*, const char * *src*)

Definition at line 71 of file main.c.

```
71                                                                  {
72     char* ldst = dst + length;
73     char s;
74     while((s = *src++)){
75         if(s == '_')
76             s = '-';
77         else
78             s = tolower(s);
79         *dst++ = s;
80         if(dst == ldst){
81             dst--;
82             break;
83         }
84     }
85     *dst = 0;
86 }
```

13.31.1.2 int main (int *argc,* char ** *argv*)

Definition at line 88 of file main.c.

References ckb_fatal_nofile, ckb_info(), ckb_info_nofile, ckb_warn_nofile, devpath, FEAT_BIND, FEAT_MOUSEACCEL, FEAT_NOTIFY, features_mask, gid, hwload_mode, keyboard, main_ac, main_av, mkdevpath(), quit(), restart(), sighandler(), and usbmain().

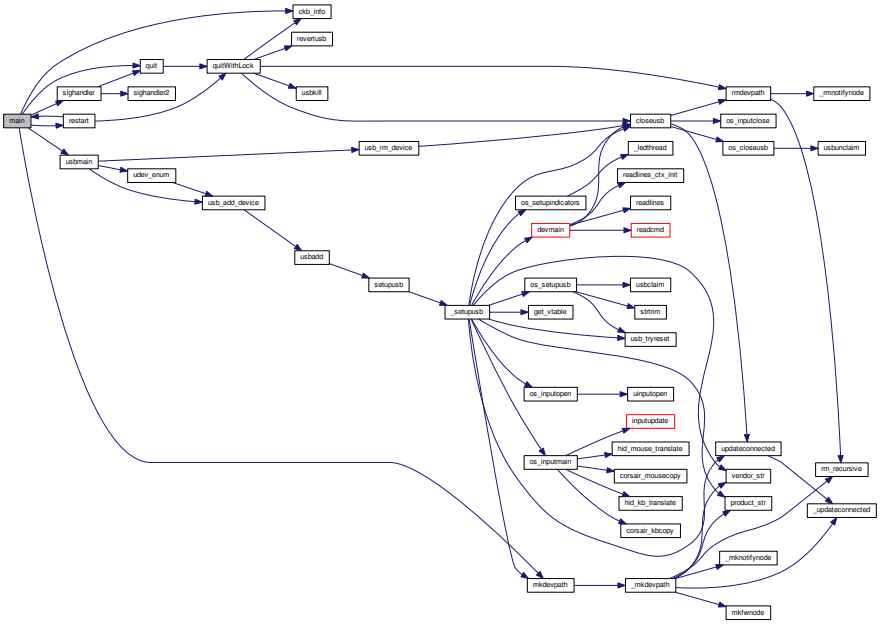
Referenced by restart().

```
88                                  {
89     // Set output pipes to buffer on newlines, if they weren't set that way already
90     setlinebuf(stdout);
91     setlinebuf(stderr);
92     main_ac = argc;
93     main_av = argv;
94
95     printf("    ckb: Corsair RGB driver %s\n", CKB_VERSION_STR);
96     // If --help occurs anywhere in the command-line, don't launch the program but instead print usage
97     for(int i = 1; i < argc; i++){
98         if(!strcmp(argv[i], "--help")){
99             printf(
100 #ifdef OS_MAC
101                         "Usage: ckb-daemon [--gid=<gid>] [--hwload=<always|try|never>] [--nonotify]
    [--nobind] [--nomouseaccel] [--nonroot]\n"
102 #else
103                         "Usage: ckb-daemon [--gid=<gid>] [--hwload=<always|try|never>] [--nonotify]
    [--nobind] [--nonroot]\n"
104 #endif
105                         "\n"
106                         "See https://github.com/ccMSC/ckb/blob/master/DAEMON.md for full instructions.\n"
107                         "\n"
108                         "Command-line parameters:\n"
109                         "    --gid=<gid>\n"
110                         "        Restrict access to %s* nodes to users in group <gid>.\n"
111                         "        (Ordinarily they are accessible to anyone)\n"
112                         "    --hwload=<always|try|never>\n"
113                         "        --hwload=always will force loading of stored hardware profiles on
    compatible devices. May result in long start up times.\n"
114                         "        --hwload=try will try to load the profiles, but give up if not immediately
    successful (default).\n"
115                         "        --hwload=never will ignore hardware profiles completely.\n"
116                         "    --nonotify\n"
117                         "        Disables key monitoring/notifications.\n"
118                         "        Note that this makes reactive lighting impossible.\n"
119                         "    --nobind\n"
120                         "        Disables all key rebinding, macros, and notifications. Implies --nonotify.
    \n"
121 #ifdef OS_MAC
122                         "    --nomouseaccel\n"
123                         "        Disables mouse acceleration, even if the system preferences enable it.\n"
124 #endif
125                         "    --nonroot\n"
126                         "        Allows running ckb-daemon as a non root user.\n"
127                         "        This will almost certainly not work. Use only if you know what you're
    doing.\n"
128                         "\n", devpath);
129             exit(0);
130         }
131     }
132
133     // Check PID, quit if already running
134     char pidpath[strlen(devpath) + 6];
135     snprintf(pidpath, sizeof(pidpath), "%s0/pid", devpath);
```

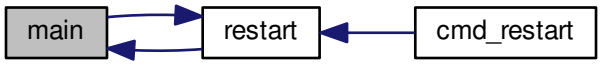
```
136     FILE* pidfile = fopen(pidpath, "r");
137     if(pidfile){
138         pid_t pid;
139         fscanf(pidfile, "%d", &pid);
140         fclose(pidfile);
141         if(pid > 0){
142             // kill -s 0 checks if the PID is active but doesn't send a signal
143             if(!kill(pid, 0)){
144                 ckb_fatal_nofile("ckb-daemon is already running (PID %d). Try `killall
    ckb-daemon`.\n", pid);
145                 ckb_fatal_nofile("(If you're certain the process is dead, delete %s and try
    again)\n", pidpath);
146                 return 0;
147             }
148         }
149     }
150 
151     // Read parameters
152     int forceroot = 1;
153     for(int i = 1; i < argc; i++){
154         char* argument = argv[i];
155         unsigned newgid;
156         char hwload[7];
157         if(sscanf(argument, "--gid=%u", &newgid) == 1){
158             // Set dev node GID
159             gid = newgid;
160             ckb_info_nofile("Setting /dev node gid: %u\n", newgid);
161         } else if(!strcmp(argument, "--nobind")){
162             // Disable key notifications and rebinding
163             features_mask &= ~FEAT_BIND & ~FEAT_NOTIFY;
164             ckb_info_nofile("Key binding and key notifications are disabled\n");
165         } else if(!strcmp(argument, "--nonotify")){
166             // Disable key notifications
167             features_mask &= ~FEAT_NOTIFY;
168             ckb_info_nofile("Key notifications are disabled\n");
169         } else if(sscanf(argument, "--hwload=%6s", hwload) == 1){
170             if(!strcmp(hwload, "always") || !strcmp(hwload, "yes") || !strcmp(hwload, "y") || !strcmp(
    hwload, "a")){
171                 hwload_mode = 2;
172                 ckb_info_nofile("Setting hardware load: always\n");
173             } else if(!strcmp(hwload, "tryonce") || !strcmp(hwload, "try") || !strcmp(hwload, "once") || !
    strcmp(hwload, "t") || !strcmp(hwload, "o")){
174                 hwload_mode = 1;
175                 ckb_info_nofile("Setting hardware load: tryonce\n");
176             } else if(!strcmp(hwload, "never") || !strcmp(hwload, "none") || !strcmp(hwload, "no") || !
    strcmp(hwload, "n")){
177                 hwload_mode = 0;
178                 ckb_info_nofile("Setting hardware load: never\n");
179             }
180         } else if(!strcmp(argument, "--nonroot")){
181             // Allow running as a non-root user
182             forceroot = 0;
183         }
184 #ifdef OS_MAC
185         else if(!strcmp(argument, "--nomouseaccel")){
186             // On OSX, provide an option to disable mouse acceleration
187             features_mask &= ~FEAT_MOUSEACCEL;
188             ckb_info_nofile("Mouse acceleration disabled\n");
189         }
190 #endif
191     }
192 
193     // Check UID
194     if(getuid() != 0){
195         if(forceroot){
196             ckb_fatal_nofile("ckb-daemon must be run as root. Try `sudo %s`\n", argv[0]);
197             exit(0);
198         } else
199             ckb_warn_nofile("Warning: not running as root, allowing anyway per command-line
    parameter...\n");
200     }
201 
202     // Make root keyboard
203     umask(0);
204     memset(keyboard, 0, sizeof(keyboard));
205     if(!mkdevpath(keyboard))
206         ckb_info("Root controller ready at %s0\n", devpath);
207 
208     // Set signals
209     sigset_t signals;
210     sigfillset(&signals);
211     sigdelset(&signals, SIGTERM);
212     sigdelset(&signals, SIGINT);
213     sigdelset(&signals, SIGQUIT);
214     sigdelset(&signals, SIGUSR1);
215     // Set up signal handlers for quitting the service.
216     sigprocmask(SIG_SETMASK, &signals, 0);
```

```
217     signal(SIGTERM, sighandler);
218     signal(SIGINT, sighandler);
219     signal(SIGQUIT, sighandler);
220     signal(SIGUSR1, (void (*)())restart);
221
222     // Start the USB system
223     int result = usbmain();
224     quit();
225     return result;
226 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.31.1.3 static void quit ( ) [static]

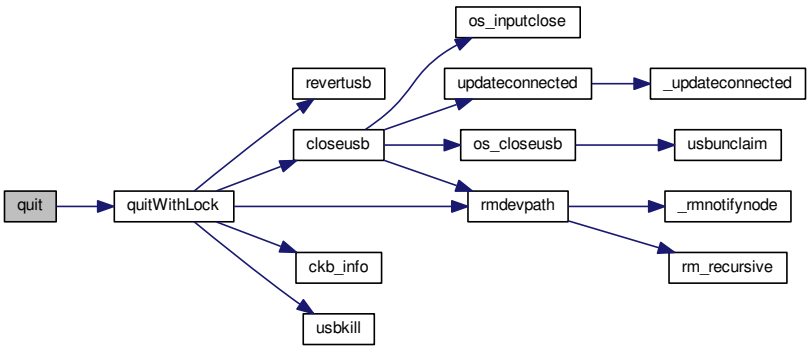
Definition at line 30 of file main.c.

References quitWithLock().

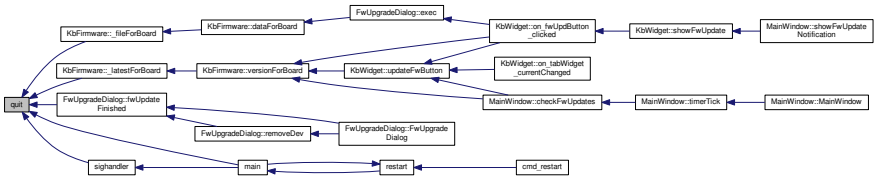
Referenced by KbFirmware::_fileForBoard(), KbFirmware::_latestForBoard(), FwUpgradeDialog::fwUpdateFinished(), main(), and sighandler().

```
30                  {
31     quitWithLock(1);
32 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.31.1.4 void quitWithLock (char *mut*) [static]

Parameters

mut	try to close files maybe without locking the mutex if mut == true then lock

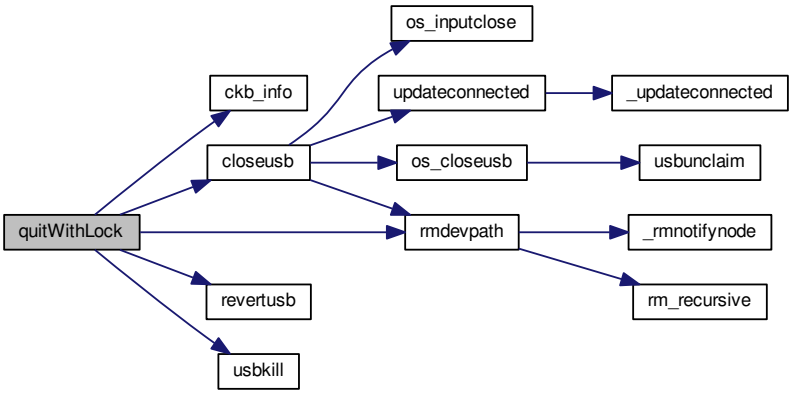
Definition at line 40 of file main.c.

References ckb_info(), closeusb(), DEV_MAX, devmutex, IS_CONNECTED, keyboard, reset_stop, revertusb(), rmdevpath(), and usbkill().

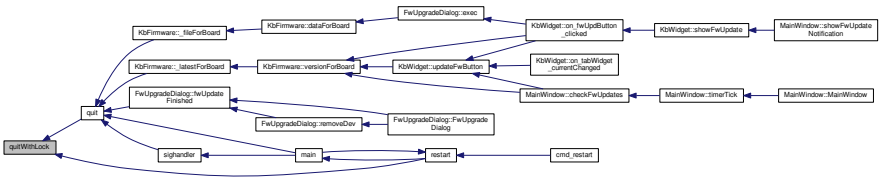
Referenced by quit(), and restart().

```
40                              {
41     // Abort any USB resets in progress
42     reset_stop = 1;
43     for(int i = 1; i < DEV_MAX; i++){
44         // Before closing, set all keyboards back to HID input mode so that the stock driver can still talk
       to them
45         if (mut) pthread_mutex_lock(devmutex + i);
46         if(IS_CONNECTED(keyboard + i)){
47             revertusb(keyboard + i);
48             closeusb(keyboard + i);
49         }
50         pthread_mutex_unlock(devmutex + i);
51     }
52     ckb_info("Closing root controller\n");
53     rmdevpath(keyboard);
54     usbkill();
55 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.31.1.5 int restart ( )

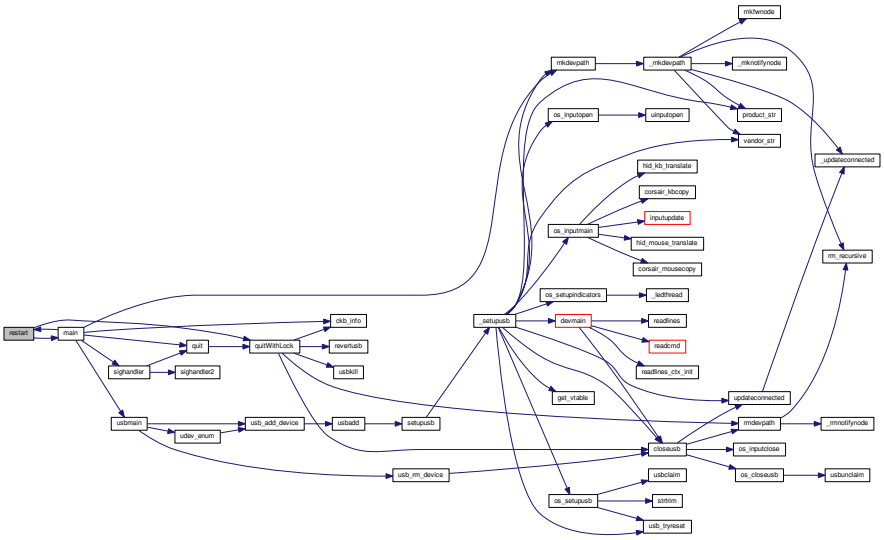
Definition at line 228 of file main.c.

References ckb_err, main(), main_ac, main_av, and quitWithLock().

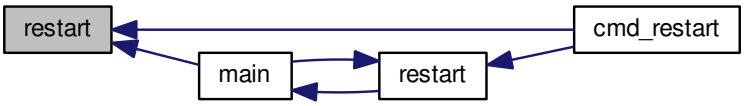
Referenced by cmd_restart(), and main().

```
228             {
229     ckb_err("restart called, running quit without mutex-lock.\n");
230     quitWithLock(0);
231     return main(main_ac, main_av);
232 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.31.1.6 void sighandler (int *type*)

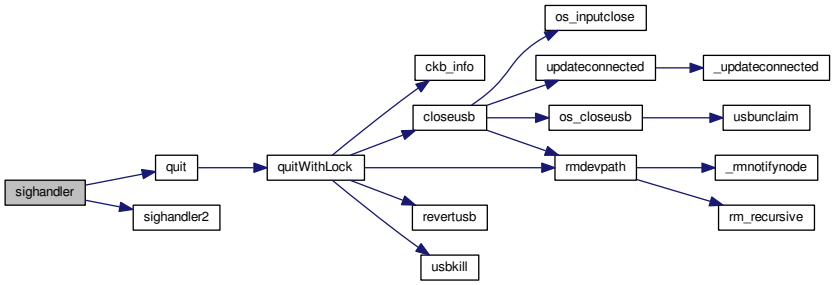
Definition at line 62 of file main.c.

References quit(), and sighandler2().

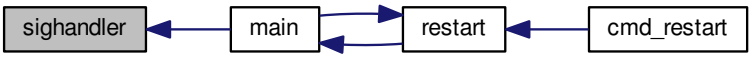
Referenced by main().

```
62                          {
63     signal(SIGTERM, sighandler2);
64     signal(SIGINT, sighandler2);
65     signal(SIGQUIT, sighandler2);
66     printf("\n[I] Caught signal %d\n", type);
67     quit();
68     exit(0);
69 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



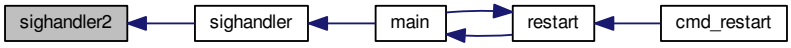
13.31.1.7 void sighandler2 (int *type*)

Definition at line 57 of file main.c.

Referenced by sighandler().

```
57                          {
58     // Don't use ckb_warn, we want an extra \n at the beginning
59     printf("\n[W] Ignoring signal %d (already shutting down)\n", type);
60 }
```

Here is the caller graph for this function:



13.31.1.8 void timespec_add (struct timespec * *timespec*, long *nanoseconds*)

Definition at line 19 of file main.c.

```
19                                                          {
20     nanoseconds += timespec->tv_nsec;
21     timespec->tv_sec += nanoseconds / 1000000000;
22     timespec->tv_nsec = nanoseconds % 1000000000;
23 }
```

13.31.2 Variable Documentation

13.31.2.1 int features_mask

features_mask Mask of features to exclude from all devices

That bit mask ist set to enable all (-1). When interpreting the input parameters, some of these bits can be cleared.

At the moment binding, notifying and mouse-acceleration can be disabled via command line.

Have a look at *main()* in main.c for details.

Definition at line 35 of file usb.c.

Referenced by _setupusb(), and main().

13.31.2.2 int hwload_mode

Definition at line 7 of file device.c.

Referenced by main().

13.31.2.3 int main_ac [static]

Definition at line 7 of file main.c.

Referenced by main(), and restart().

13.31.2.4 char** main_av [static]

Definition at line 8 of file main.c.

Referenced by main(), and restart().

13.31.2.5 volatile int reset_stop

reset_stop is boolean: Reset stopper for when the program shuts down.

Is set only by *quit()* to true (1) to inform several usb_* functions to end their loops and tries.

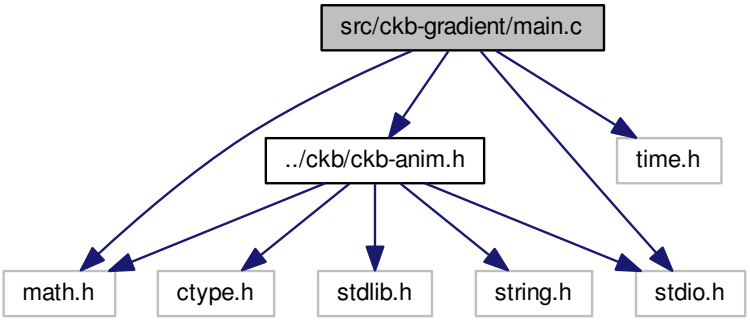
Definition at line 25 of file usb.c.

Referenced by _usbrecv(), _usbsend(), quitWithLock(), and usb_tryreset().

13.32 src/ckb-gradient/main.c File Reference

```
#include "../ckb/ckb-anim.h"
#include <math.h>
#include <time.h>
#include <stdio.h>
```

Include dependency graph for main.c:



Data Structures

- struct keyAnim

Macros

- #define NONE -1.f
- #define HOLD -2.f

Functions

- void ckb_info ()
- void ckb_init (ckb_runctx *context)
- void ckb_parameter (ckb_runctx *context, const char *name, const char *value)
- void ckb_keypress (ckb_runctx *context, ckb_key *key, int x, int y, int state)
- void ckb_start (ckb_runctx *context, int state)
- void ckb_time (ckb_runctx *context, double delta)
- int ckb_frame (ckb_runctx *context)

Variables

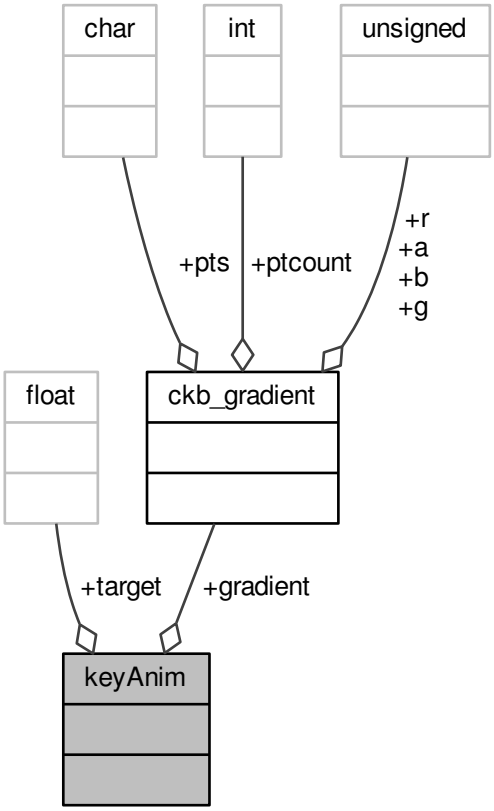
- ckb_gradient animcolor = { 0 }
- int kphold = 0
- int kprelease = 0
- int randomize = 0
- struct keyAnim * anim

13.32.1 Data Structure Documentation

13.32.1.1 struct keyAnim

Definition at line 47 of file main.c.

Collaboration diagram for keyAnim:



Data Fields

ckb_gradient	gradient	
float	target	

13.32.2 Macro Definition Documentation

13.32.2.1 #define HOLD -2.f

Definition at line 45 of file main.c.

Referenced by ckb_frame(), and ckb_keypress().

13.32.2.2 #define NONE -1.f

Definition at line 44 of file main.c.

Referenced by ckb_init(), ckb_keypress(), and ckb_start().

13.32.3 Function Documentation

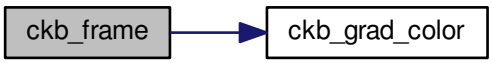
13.32.3.1 int ckb_frame (ckb_runctx * *context*)

Definition at line 137 of file main.c.

References ckb_key::a, animcolor, ckb_key::b, ckb_grad_color(), ckb_key::g, keyAnim::gradient, HOLD, ckb_runctx::keycount, ckb_runctx::keys, phase, ckb_key::r, randomize, and keyAnim::target.

```
137                                 {
138     // Draw key colors
139     unsigned count = context->keycount;
140     for(unsigned i = 0; i < count; i++){
141         float phase = anim[i].target;
142 
143         if(phase == HOLD)
144             phase = 0.f;
145         else if(phase < 0.f)
146             phase = 1.f;
147         ckb_key* key = context->keys + i;
148         float a, r, g, b;
149         ckb_gradient thisGradient;
150         if (randomize) {
151             thisGradient = anim[i].gradient;
152         } else {
153             thisGradient = animcolor;
154         }
155         ckb_grad_color(&a, &r, &g, &b, &thisGradient, phase * 100.);
156         key->a = a;
157         key->r = r;
158         key->g = g;
159         key->b = b;
160     }
161     return 0;
162 }
```

Here is the call graph for this function:



13.32.3.2 void ckb_info ( )

Definition at line 5 of file main.c.

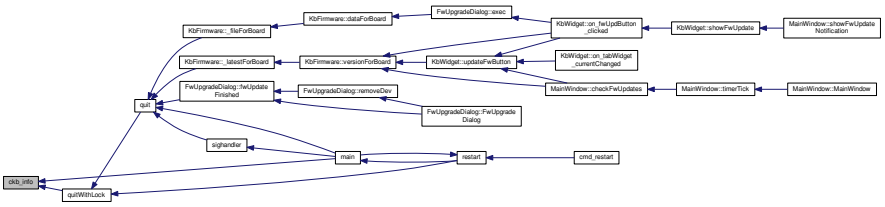
References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_NAME, CKB_KPMODE, CKB_LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, CKB_PARAM_BOOL, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_TIME_DURATION, CKB_TIMEMODE, CKB_VERSION, and TRUE.

Referenced by main(), and quitWithLock().

```
5                {
6     // Plugin info
7     CKB_NAME("Gradient");
8     CKB_VERSION("0.9");
9     CKB_COPYRIGHT("2014-2016", "MSC");
10     CKB_LICENSE("GPLv2");
11     CKB_GUID("{54DD2975-E192-457D-BCFC-D912A24E33B4}");
12     CKB_DESCRIPTION("A transition from one color to another.");
13 
14     // Effect parameters
15     CKB_PARAM_AGRADIENT("color", "Color:", "", "ffffffff");
16     CKB_PARAM_BOOL("kphold", "Freeze until key is released", TRUE);
17     CKB_PARAM_BOOL("randomize", "Randomly select from gradient", 0);
```

```
18
19      // Timing/input parameters
20      CKB_KPMODE(CKB_KP_NAME);
21      CKB_TIMEMODE(CKB_TIME_DURATION);
22      CKB_LIVEPARAMS(TRUE);
23
24      // Presets
25      CKB_PRESET_START("Fade in");
26      CKB_PRESET_PARAM("color", "ff000000");
27      CKB_PRESET_PARAM("duration", "1.0");
28      CKB_PRESET_PARAM("stop", "0");
29      CKB_PRESET_PARAM("kpstop", "0");
30      CKB_PRESET_END;
31
32      CKB_PRESET_START("Rainbow");
33      CKB_PRESET_PARAM("color", "0:ffff0000 17:ffffff00 33:ff00ff00 50:ff00ffff 67:ff0000ff
        83:ffff00ff 100:ffff0000");
34      CKB_PRESET_PARAM("duration", "2.0");
35      CKB_PRESET_END;
36
37      CKB_PRESET_START("Type lighting");
38      CKB_PRESET_PARAM("duration", "1.0");
39      CKB_PRESET_PARAM("trigger", "0");
40      CKB_PRESET_PARAM("kptrigger", "1");
41      CKB_PRESET_END;
42  }
```

Here is the caller graph for this function:



13.32.3.3 void ckb_init (ckb_runctx * *context*)

Definition at line 57 of file main.c.

References ckb_runctx::keycount, and NONE.

```
57                                  {
58      // Initialize all keys to 100% (animation over)
59
60      srand((unsigned)time(NULL));
61      unsigned count = context->keycount;
62      anim =  malloc(count * sizeof *anim);
63      for(unsigned i = 0; i < count; i++) {
64          anim[i].target = NONE;
65      }
66  }
```

13.32.3.4 void ckb_keypress (ckb_runctx * *context*, ckb_key * *key*, int *x*, int *y*, int *state*)

Definition at line 75 of file main.c.

References ckb_gradient::a, ckb_gradient::b, ckb_grad_color(), ckb_gradient::g, keyAnim::gradient, HOLD, ckb_runctx::keys, kphold, kprelease, NONE, ckb_gradient::ptcount, ckb_gradient::pts, ckb_gradient::r, randomize, and keyAnim::target.

```
75                                                                          {
76      // Start or stop animation on key
77      int i = key - context->keys;
78      if(state){
79          if (randomize) {
```

```
80              float a, r, g, b;
81              float colorChoice = (float)rand()/(float)(RAND_MAX);
82              ckb_grad_color(&a, &r, &g, &b, &animcolor, colorChoice * 100.);
83              ckb_gradient newGradient = { 0 };
84              newGradient.ptcount = 2;
85              newGradient.pts[0] = 0;
86              newGradient.pts[1] = 100;
87              newGradient.a[0] = 255;
88              newGradient.b[0] = b;
89              newGradient.g[0] = g;
90              newGradient.r[0] = r;
91              newGradient.a[1] = 0;
92              newGradient.b[1] = b;
93              newGradient.g[1] = g;
94              newGradient.r[1] = r;
95
96              anim[i].gradient = newGradient;
97          }
98
99          if(kphold) {
100             anim[i].target = HOLD;
101         } else {
102             anim[i].target = 0.f;
103         }
104     } else {
105         if(kprelease) {
106             anim[i].target = NONE;
107         }
108         else if(kphold) {
109             anim[i].target = 0.f;
110         }
111     }
112 }
```

Here is the call graph for this function:

13.32.3.5 void ckb_parameter (ckb_runctx * *context*, const char * *name*, const char * *value*)

Definition at line 68 of file main.c.

References CKB_PARSE_AGRADIENT, CKB_PARSE_BOOL, kphold, kprelease, and randomize.

```
68                                                                          {
69     CKB_PARSE_AGRADIENT("color", &animcolor){}
70     CKB_PARSE_BOOL("kphold", &kphold){}
71     CKB_PARSE_BOOL("kprelease", &kprelease){}
72     CKB_PARSE_BOOL("randomize", &randomize){}
73 }
```

13.32.3.6 void ckb_start (ckb_runctx * *context*, int *state*)

Definition at line 114 of file main.c.

References ckb_runctx::keycount, NONE, and keyAnim::target.

```
114                                             {
115     // Start/stop all keys
116     unsigned count = context->keycount;
117     if(state) {
```

```
118         memset(anim, 0, count * sizeof *anim);
119     } else {
120         for(unsigned i = 0; i < count; i++) {
121             anim[i].target = NONE;
122         }
123     }
124 }
```

13.32.3.7 void ckb_time (ckb_runctx * *context*, double *delta*)

Definition at line 126 of file main.c.

References ckb_runctx::keycount, phase, and keyAnim::target.

```
126                                                 {
127     // Advance animation on each key
128     unsigned count = context->keycount;
129     for(unsigned i = 0; i < count; i++){
130         float phase = anim[i].target;
131         if(phase > 1.f || phase < 0.f)
132             continue;
133         anim[i].target = phase + delta;
134     }
135 }
```

13.32.4 Variable Documentation

13.32.4.1 struct keyAnim* anim

Definition at line 55 of file main.c.

Referenced by KbLight::addAnim(), anim_add(), anim_remove(), KbLight::animKeypress(), RebindWidget::applyChanges(), ckb_frame(), ckb_time(), KbLight::close(), KbLight::duplicateAnim(), KbLight::findAnim(), KbLight::frameUpdate(), KeyAction::keyEvent(), KbLight::load(), KbLight::map(), KbLight::needsSave(), KbMode::newId(), KbAnimWidget::on_animList_itemChanged(), KbLight::open(), KbLight::previewAnim(), KbAnimWidget::refreshList(), KbAnimWidget::reorderAnims(), KbLight::restartAnimation(), KbLight::save(), and RebindWidget::setSelection().

13.32.4.2 ckb_gradient animcolor = { 0 }

Definition at line 52 of file main.c.

Referenced by ckb_frame().

13.32.4.3 int kphold = 0

Definition at line 53 of file main.c.

Referenced by ckb_keypress(), and ckb_parameter().

13.32.4.4 int kprelease = 0

Definition at line 53 of file main.c.

Referenced by ckb_keypress(), ckb_parameter(), ckb_start(), and AnimScript::load().

13.32.4.5 int randomize = 0

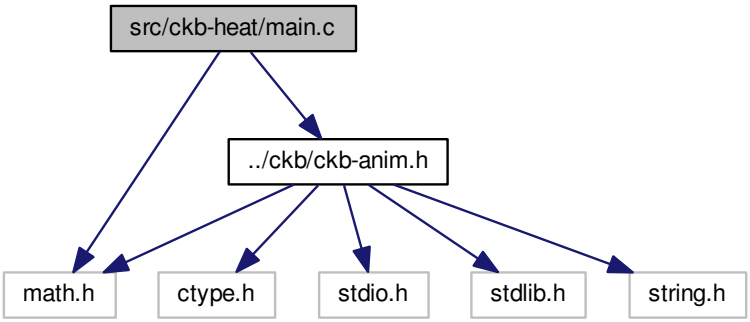
Definition at line 53 of file main.c.

Referenced by ckb_frame(), ckb_keypress(), and ckb_parameter().

13.33 src/ckb-heat/main.c File Reference

```
#include "../ckb/ckb-anim.h"
#include <math.h>
```

Include dependency graph for main.c:



Data Structures

- struct keyanim

Functions

- void ckb_info ()
- void ckb_init (ckb_runctx *context)
- void ckb_parameter (ckb_runctx *context, const char *name, const char *value)
- void anim_add (int index)
- void anim_remove (int index)
- void ckb_keypress (ckb_runctx *context, ckb_key *key, int x, int y, int state)
- void ckb_start (ckb_runctx *context, int state)
- void ckb_time (ckb_runctx *context, double delta)
- int ckb_frame (ckb_runctx *context)

Variables

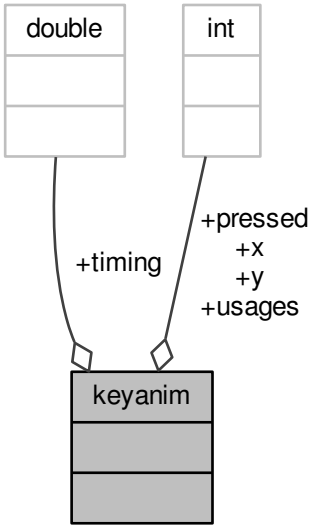
- ckb_gradient animcolor = { 0 }
- int randomBright = 0
- long ffade = 30
- double pressestofull = 10.f
- keyanim * anims = NULL

13.33.1 Data Structure Documentation

13.33.1.1 struct keyanim

Definition at line 56 of file main.c.

Collaboration diagram for keyanim:



Data Fields

int	pressed	
double	timing	
unsigned int	usages	
int	x	
int	y	

13.33.2 Function Documentation

13.33.2.1 void anim_add (int *index*)

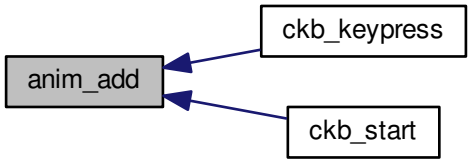
Definition at line 81 of file main.c.

References ffade, keyanim::pressed, keyanim::timing, and keyanim::usages.

Referenced by ckb_keypress(), and ckb_start().

```
81                          {
82     anims[index].usages += ffade;
83     anims[index].pressed = 1;
84     anims[index].timing = 0.f;
85 }
```

Here is the caller graph for this function:



13.33.2.2 void anim_remove (int *index*)

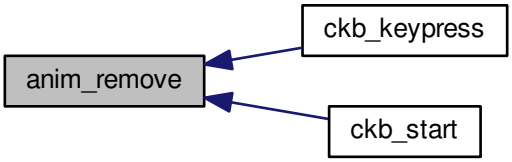
Definition at line 87 of file main.c.

References keyanim::pressed.

Referenced by ckb_keypress(), and ckb_start().

```
87                              {
88     anims[index].pressed = 0;
89 }
```

Here is the caller graph for this function:



13.33.2.3 int ckb_frame (ckb_runctx * *context*)

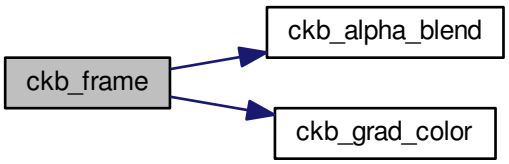
Definition at line 117 of file main.c.

References ckb_alpha_blend(), ckb_grad_color(), CKB_KEYCLEAR, ffade, ckb_runctx::keycount, ckb_runctx::keys, pressestofull, and randomBright.

```
117                                 {
118     CKB_KEYCLEAR(context);
119     // Draw spots
120     unsigned count = context->keycount;
121     ckb_key* keys = context->keys;
122     unsigned int full = pressestofull*ffade;
123     for(int i =0; i < count; i++){
124         if(!anims[i].usages)
125             continue;
126         float a, r, g, b;
```

```
127         if(randomBright)
128             ckb_grad_color(&a, &r, &g, &b, &animcolor, ((float)(rand() % 1024))/10.f
    );
129         else
130             ckb_grad_color(&a, &r, &g, &b, &animcolor, (float)fmin(
    anims[i].usages,full)*100.f/full);
131         ckb_alpha_blend(keys+i, a, r, g, b);
132     }
133     return 0;
134 }
```

Here is the call graph for this function:



13.33.2.4 void ckb_info ( )

Definition at line 20 of file main.c.

References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_POSITION, CKB_KPMODE, CKB_-LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, CKB_PARAM_BOOL, CKB_PARAM_-DOUBLE, CKB_PARAM_LONG, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_R-EPEAT, CKB_TIME_ABSOLUTE, CKB_TIMEMODE, CKB_VERSION, FALSE, and TRUE.

```
20              {
21     // Plugin info
22     CKB_NAME("Heat Map");
23     CKB_VERSION("0.2");
24     CKB_COPYRIGHT("2017", "RULER501");
25     CKB_LICENSE("GPLv2");
26     CKB_GUID("{097D69F0-70B2-48B8-AFE2-25A1CDB0D92C}");
27     CKB_DESCRIPTION("A spot effect on pressed keys that shows usage");
28
29     // Effect parameters
30     CKB_PARAM_AGRADIENT("color", "Fade color:", "", "ffffffff");
31     CKB_PARAM_BOOL("random", "Random Brightness", 0);
32     CKB_PARAM_LONG("ffade", "Frames to fade", "frames", 30, 10, 1000);
33     CKB_PARAM_DOUBLE("pressestofull", "Presses to full", "keypresses", 10.f, 1.f, 100.f);
34
35     // Timing/input parameters
36     CKB_KPMODE(CKB_KP_POSITION);
37     CKB_TIMEMODE(CKB_TIME_ABSOLUTE);
38     CKB_REPEAT(FALSE);
39     CKB_LIVEPARAMS(TRUE);
40
41     // Presets
42     CKB_PRESET_START("Single Spot");
43     CKB_PRESET_PARAM("random", "0");
44     CKB_PRESET_PARAM("ffade", "30");
45     CKB_PRESET_PARAM("pressestofull", "10");
46     CKB_PRESET_PARAM("trigger", "0");
47     CKB_PRESET_PARAM("kptrigger", "1");
48     CKB_PRESET_END;
49 }
```

13.33.2.5 void ckb_init (ckb_runctx * *context*)

Definition at line 66 of file main.c.

References ckb_runctx::keycount, ckb_runctx::keys, keyanim::x, ckb_key::x, keyanim::y, and ckb_key::y.

```
66                                  {
67      anims = calloc(context->keycount, sizeof(keyanim));
68      for(int i=0; i < context->keycount; i++){
69          anims[i].x = context->keys[i].x;
70          anims[i].y = context->keys[i].y;
71      }
72  }
```

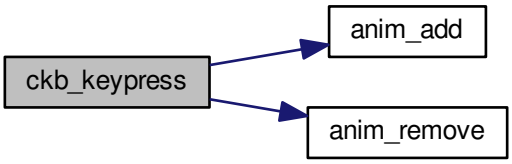
13.33.2.6 void ckb_keypress (ckb_runctx * *context*, ckb_key * *key*, int *x*, int *y*, int *state*)

Definition at line 91 of file main.c.

References anim_add(), anim_remove(), and ckb_runctx::keys.

```
91                                                                                  {
92      // Add or remove a spot on this key
93      if(state)
94          anim_add(key - context->keys);
95      else
96          anim_remove(key - context->keys);
97  }
```

Here is the call graph for this function:



13.33.2.7 void ckb_parameter (ckb_runctx * *context*, const char * *name*, const char * *value*)

Definition at line 74 of file main.c.

References CKB_PARSE_AGRADIENT, CKB_PARSE_BOOL, CKB_PARSE_DOUBLE, CKB_PARSE_LONG, ffade, pressestofull, and randomBright.

```
74                                                                                  {
75      CKB_PARSE_AGRADIENT("color", &animcolor){}
76      CKB_PARSE_BOOL("random", &randomBright){}
77      CKB_PARSE_LONG("ffade", &ffade){}
78      CKB_PARSE_DOUBLE("pressestofull", &pressestofull){}
79  }
```

13.33.2.8 void ckb_start (ckb_runctx * *context*, int *state*)

Definition at line 99 of file main.c.

```
99                                          {
100     return;
101 }
```

13.33.2.9 void ckb_time (ckb_runctx * *context*, double *delta*)

Definition at line 103 of file main.c.

References ckb_runctx::keycount, keyanim::timing, and keyanim::usages.

```
103                                                 {
104     unsigned int count = context->keycount;
105     for(unsigned int i=0; i < count; i++){
106         if(anims[i].usages && !anims[i].pressed){
107             anims[i].timing -= delta;
108             while(anims[i].timing < 0){
109                 anims[i].timing += 1.f/30.f;
110                 if(anims[i].usages > 0)
111                     anims[i].usages--;
112             }
113         }
114     }
115 }
```

13.33.3 Variable Documentation

13.33.3.1 ckb_gradient animcolor = { 0 }

Definition at line 51 of file main.c.

13.33.3.2 keyanim* anims = NULL

Definition at line 64 of file main.c.

13.33.3.3 long ffade = 30

Definition at line 53 of file main.c.

Referenced by anim_add(), ckb_frame(), and ckb_parameter().

13.33.3.4 double pressestofull = 10.f

Definition at line 54 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.33.3.5 int randomBright = 0

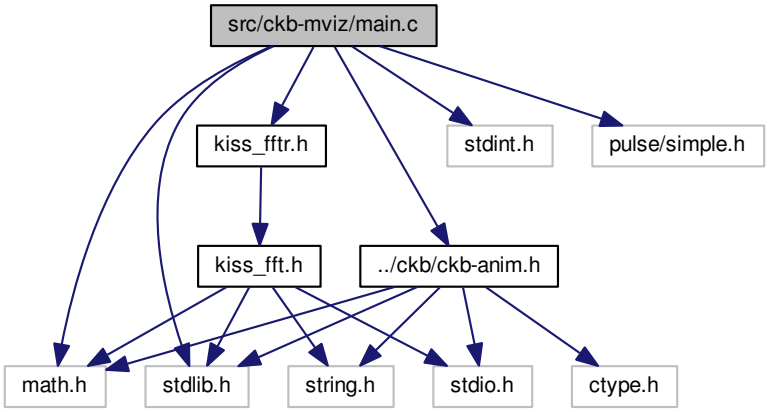
Definition at line 52 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.34 src/ckb-mviz/main.c File Reference

```
#include <math.h>
#include <stdint.h>
#include <stdlib.h>
#include <pulse/simple.h>
#include "../ckb/ckb-anim.h"
#include "kiss_fftr.h"
```

Include dependency graph for main.c:



Functions

- void ckb_info ()
- void ckb_init (ckb_runctx *context)
- void ckb_parameter (ckb_runctx *context, const char *name, const char *value)
- void anim_add (ckb_key *press, float x, float y)
- void anim_remove (float x, float y)
- void ckb_keypress (ckb_runctx *context, ckb_key *key, int x, int y, int state)
- void ckb_start (ckb_runctx *context, int state)
- void ckb_time (ckb_runctx *context, double delta)
- int max (int a, int b)
- int min (int a, int b)
- void getFreqDec ()
- int ckb_frame (ckb_runctx *context)

Variables

- double powers [2048] = { 0.f }
- kiss_fft_cpx * inbuf
- kiss_fft_cpx * outbuf
- ckb_gradient animcolor = { 0 }
- pa_simple * pas = NULL
- int power = 0
- int gcounter = 0

13.34.1 Function Documentation

13.34.1.1 void anim_add (ckb_key * *press*, float *x*, float *y*)

Definition at line 77 of file main.c.

```
77                                                  {
78     return;
79 }
```

13.34.1.2 void anim_remove (float *x*, float *y*)

Definition at line 81 of file main.c.

```
81                                      {
82     return;
83 }
```

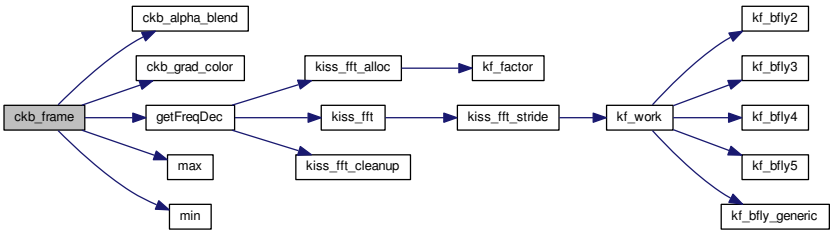
13.34.1.3 int ckb_frame (ckb_runctx * *context*)

Definition at line 126 of file main.c.

References ckb_alpha_blend(), ckb_grad_color(), CKB_KEYCLEAR, getFreqDec(), ckb_runctx::height, ckb_runctx::keycount, ckb_runctx::keys, max(), min(), power, powers, and ckb_runctx::width.

```
126                                     {
127     CKB_KEYCLEAR(context);
128     ckb_key* keys = context->keys;
129     ckb_key* maxkey = keys+context->keycount-1;
130     getFreqDec();
131     unsigned int frames = context->width*context->height - 1;
132     int height = context->height;
133     for(ckb_key* key = keys; key < maxkey; key++){
134         int posl = height*key->x + key->y - 1;
135         posl = max(posl, 0);
136         int posr = height*key->x + key->y + 1;
137         posr = max(posr, 0);
138         int lowi = floorf(pow(2,posl*11.f/frames));
139         int highi = ceilf(pow(2,posr*11.f/frames));
140         highi= min(highi, (int)sizeof(powers)/sizeof(double)-1);
141         lowi = max(lowi, 0);
142         double total = 0;
143         unsigned int height = context->height;
144         for(unsigned int i = lowi; i <= highi; i++)
145             total += powers[i];
146         total /= highi - lowi + 1;
147         float a, r, g, b;
148         ckb_grad_color(&a, &r, &g, &b, &animcolor, total/(
      power ? 150994944.f : 12288.f));
149         ckb_alpha_blend(key, a, r, g, b);
150     }
151     return 0;
152 }
```

Here is the call graph for this function:



13.34.1.4 void ckb_info ( )

Definition at line 27 of file main.c.

References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_NONE, CKB_KPMODE, CKB_LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, CKB_PARAM_BOOL, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_REPEAT, CKB_TIME_ABSOLUTE, CKB_TIMEMODE, CKB_VERSION, FALSE, and TRUE.

```
27              {
28     // Plugin info
29     CKB_NAME("Music Visualization");
30     CKB_VERSION("0.2");
31     CKB_COPYRIGHT("2017", "RULER501");
32     CKB_LICENSE("GPLv2");
33     CKB_GUID("{097D69F0-70B2-48B8-AFE2-25CA1DB0D92C}");
34     CKB_DESCRIPTION("A collection of music visualization effects");
35
36     // Effect parameters
37     CKB_PARAM_AGRADIENT("color", "Fade color:", "", "ffffffff");
38     CKB_PARAM_BOOL("power", "Use Power instead of Magnitude?", 0);
39
40     // Timing/input parameters
41     CKB_KPMODE(CKB_KP_NONE);
42     CKB_TIMEMODE(CKB_TIME_ABSOLUTE);
43     CKB_REPEAT(FALSE);
44     CKB_LIVEPARAMS(TRUE);
45
46     // Presets
47     CKB_PRESET_START("Default");
48     CKB_PRESET_PARAM("power", "0");
49     CKB_PRESET_PARAM("trigger", "0");
50     CKB_PRESET_PARAM("kptrigger", "1");
51     CKB_PRESET_END;
52 }
```

13.34.1.5 void ckb_init (ckb_runctx * *context*)

Definition at line 61 of file main.c.

References pas.

```
61                                  {
62     static const pa_sample_spec ss ={
63         .format = PA_SAMPLE_S16LE,
64         .rate = 44100,
65         .channels = 1
66     };
67     pas = pa_simple_new(NULL, "CKB Music Viz", PA_STREAM_RECORD, NULL, "CKB Music Viz", &ss, NULL, NULL,
       NULL);
68     inbuf = malloc(2048*sizeof(kiss_fft_cpx));
69     outbuf = malloc(2048*sizeof(kiss_fft_cpx));
70 }
```

13.34.1.6 void ckb_keypress (ckb_runctx * *context,* ckb_key * *key,* int *x,* int *y,* int *state*)

Definition at line 85 of file main.c.

```
85                                                                                  {
86     return;
87 }
```

13.34.1.7 void ckb_parameter (ckb_runctx * *context,* const char * *name,* const char * *value*)

Definition at line 72 of file main.c.

References CKB_PARSE_AGRADIENT, CKB_PARSE_BOOL, and power.

```
72                                                                                  {
73     CKB_PARSE_AGRADIENT("color", &animcolor){}
74     CKB_PARSE_BOOL("power", &power);
75 }
```

13.34.1.8 void ckb_start (ckb_runctx * *context,* int *state*)

Definition at line 89 of file main.c.

```
89                                                  {
90      return;
91  }
```

13.34.1.9 void ckb_time (ckb_runctx * *context*, double *delta*)

Definition at line 93 of file main.c.

```
93                                                  {
94      return;
95  }
```

13.34.1.10 void getFreqDec ( )

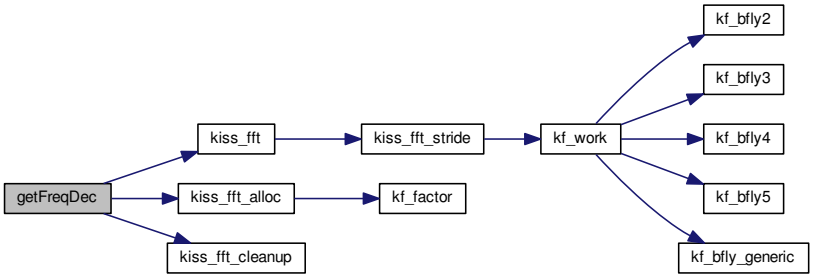
Definition at line 107 of file main.c.

References kiss_fft_cpx::i, kiss_fft(), kiss_fft_alloc(), kiss_fft_cleanup(), kiss_fft_free, pas, power, powers, and kiss_fft_cpx::r.

Referenced by ckb_frame().

```
107                 {
108     int16_t data[2048];
109     pa_simple_read(pas, data, sizeof(data), NULL);
110     for(int j=0; j<2048; j++){
111         inbuf[j].r = data[j];
112         inbuf[j].i = 0;
113     }
114     kiss_fft_cfg config = kiss_fft_alloc(2048, 0, NULL, NULL);
115     kiss_fft(config, inbuf, outbuf);
116 
117     for(unsigned int j=0; j < 2048; j++)
118         if(power)
119             powers[j] = outbuf[j].r*outbuf[j].r + outbuf[j].
    i*outbuf[j].i;
120         else
121             powers[j] = sqrt(outbuf[j].r*outbuf[j].r + outbuf[j].i*
    outbuf[j].i);
122     kiss_fft_free(config);
123     kiss_fft_cleanup();
124 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

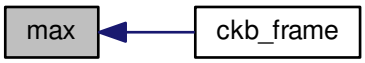
13.34.1.11 int max (int *a*, int *b*)

Definition at line 97 of file main.c.

Referenced by ckb_frame().

```
97                      {
98     return a > b ? a : b;
99 }
```

Here is the caller graph for this function:



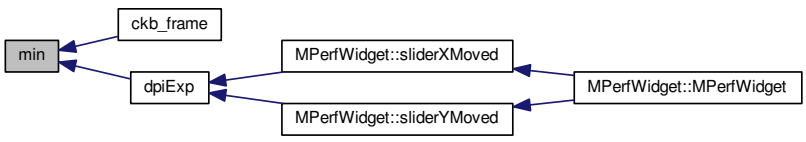
13.34.1.12 int min (int *a*, int *b*)

Definition at line 101 of file main.c.

Referenced by ckb_frame(), and dpiExp().

```
101                     {
102     return a < b ? a : b;
103 }
```

Here is the caller graph for this function:



13.34.2 Variable Documentation

13.34.2.1 ckb_gradient animcolor = { 0 }

Definition at line 57 of file main.c.

13.34.2.2 int gcounter = 0

Definition at line 105 of file main.c.

13.34.2.3 kiss_fft_cpx* inbuf

Definition at line 55 of file main.c.

13.34.2.4 kiss_fft_cpx* outbuf

Definition at line 56 of file main.c.

13.34.2.5 pa_simple* pas = NULL

Definition at line 58 of file main.c.

Referenced by ckb_init(), and getFreqDec().

13.34.2.6 int power = 0

Definition at line 59 of file main.c.

Referenced by ckb_frame(), ckb_parameter(), and getFreqDec().

13.34.2.7 double powers[2048] = { 0.f }

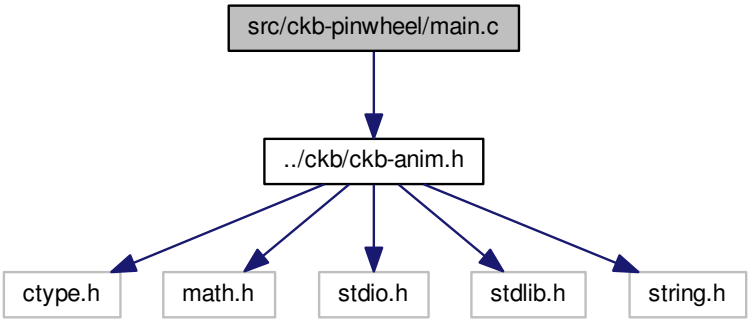
Definition at line 54 of file main.c.

Referenced by ckb_frame(), and getFreqDec().

13.35 src/ckb-pinwheel/main.c File Reference

```
#include "../ckb/ckb-anim.h"
```

Include dependency graph for main.c:



Macros

- #define ANGLE(theta) fmod((theta) + M_PI ∗ 2., M_PI ∗ 2.)

Functions

- void ckb_info ()
- void ckb_parameter (ckb_runctx ∗context, const char ∗name, const char ∗value)
- void ckb_init (ckb_runctx ∗context)
- void ckb_keypress (ckb_runctx ∗context, ckb_key ∗key, int x, int y, int state)
- void ckb_start (ckb_runctx ∗context, int state)
- void ckb_time (ckb_runctx ∗context, double delta)
- int ckb_frame (ckb_runctx ∗context)

Variables

- ckb_gradient animcolor = { 0 }
- double animlength = 0.
- int symmetric = 0
- int counter_clock = 0
- double x_offset = 0
- double y_offset = 0
- double frame = -1.
- float x
- float y

13.35.1 Macro Definition Documentation

13.35.1.1 #define ANGLE(*theta*) fmod((theta) + M_PI ∗ 2., M_PI ∗ 2.)

Definition at line 68 of file main.c.

Referenced by ckb_frame().

13.35.2 Function Documentation

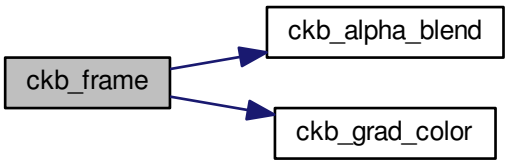
13.35.2.1 int ckb_frame (ckb_runctx * *context*)

Definition at line 86 of file main.c.

References ANGLE, animlength, ckb_alpha_blend(), ckb_grad_color(), CKB_KEYCLEAR, counter_clock, frame, ckb_runctx::keycount, ckb_runctx::keys, M_PI, symmetric, x, and y.

```
86                                  {
87      CKB_KEYCLEAR(context);
88      if(frame < 0.)
89          return 0;
90      // Color each key according to its angle from the center
91      float position;
92      if(counter_clock)
93          position = ANGLE(frame * M_PI * 2.);
94      else
95          position = ANGLE(-frame * M_PI * 2.);
96      unsigned count = context->keycount;
97      ckb_key* keys = context->keys;
98      for(ckb_key* key = keys; key < keys + count; key++){
99          float theta;
100         if(key->x == x && key->y == y)
101             // Dead center = 0°
102             theta = 0.f;
103         else
104             theta = ANGLE(ANGLE(atan2(x - key->x, y - key->y)) - position);
105         // If the animation is symmetric, mirror the second half
106         if(symmetric && theta > M_PI)
107             theta = M_PI * 2. - theta;
108         // Draw the gradient position that corresponds to this angle
109         if(theta < animlength){
110             float distance = theta / animlength;
111             float a, r, g, b;
112             ckb_grad_color(&a, &r, &g, &b, &animcolor, distance * 100.);
113             ckb_alpha_blend(key, a, r, g, b);
114         }
115     }
116     return 0;
117 }
```

Here is the call graph for this function:



13.35.2.2 void ckb_info ( )

Definition at line 3 of file main.c.

References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_NONE, CKB_KPMODE, CKB_LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, CKB_PARAM_BOOL, CKB_PARAM_DOUBLE, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_REPEAT, CKB_TIME_DURATION, CKB_TIMEMODE, CKB_VERSION, FALSE, and TRUE.

```
3               {
4       // Plugin info
```

```
5     CKB_NAME("Pinwheel");
6     CKB_VERSION("0.9");
7     CKB_COPYRIGHT("2014-2016", "MSC");
8     CKB_LICENSE("GPLv2");
9     CKB_GUID("{07551A90-D97A-4DD0-A770-E9E280A90891}");
10     CKB_DESCRIPTION("A spinning pinwheel effect.");
11
12     // Effect parameters
13     CKB_PARAM_AGRADIENT("color", "Wheel color:", "", "ffffffff");
14     CKB_PARAM_DOUBLE("length", "Wheel size:", "%", 100, 1., 100.);
15     CKB_PARAM_DOUBLE("x_offset", "X offset:", "%", 0, -100, 100);
16     CKB_PARAM_DOUBLE("y_offset", "Y offset:", "%", 0, -100, 100);
17     CKB_PARAM_BOOL("counter_clock", "Counter Clockwise", 0);
18     CKB_PARAM_BOOL("symmetric", "Symmetric", 0);
19
20     // Timing/input parameters
21     CKB_KPMODE(CKB_KP_NONE);
22     CKB_TIMEMODE(CKB_TIME_DURATION);
23     CKB_LIVEPARAMS(TRUE);
24     CKB_REPEAT(FALSE);
25
26     // Presets
27     CKB_PRESET_START("Search light");
28     CKB_PRESET_PARAM("duration", "2.0");
29     CKB_PRESET_PARAM("length", "50.0");
30     CKB_PRESET_PARAM("symmetric", "1");
31     CKB_PRESET_END;
32
33     CKB_PRESET_START("Rainbow");
34     CKB_PRESET_PARAM("color", "0:ffff0000 17:ffffff00 33:ff00ff00 50:ff00ffff 67:ff0000ff
       83:ffff00ff 100:ffff0000");
35     CKB_PRESET_PARAM("duration", "2.0");
36     CKB_PRESET_END;
37 }
```

13.35.2.3 void ckb_init (ckb_runctx * *context*)

Definition at line 57 of file main.c.

```
57                                  {
58     // Unused
59 }
```

13.35.2.4 void ckb_keypress (ckb_runctx * *context*, ckb_key * *key*, int *x*, int *y*, int *state*)

Definition at line 61 of file main.c.

```
61                                                                          {
62     // Unused
63 }
```

13.35.2.5 void ckb_parameter (ckb_runctx * *context*, const char * *name*, const char * *value*)

Definition at line 45 of file main.c.

References animlength, CKB_PARSE_AGRADIENT, CKB_PARSE_BOOL, CKB_PARSE_DOUBLE, counter_clock, M_PI, symmetric, x_offset, and y_offset.

```
45                                                                          {
46     CKB_PARSE_AGRADIENT("color", &animcolor){}
47     double len;
48     CKB_PARSE_DOUBLE("length", &len){
49         animlength = len / 100. * M_PI * 2.;
50     }
51     CKB_PARSE_BOOL("symmetric", &symmetric){}
52     CKB_PARSE_BOOL("counter_clock", &counter_clock){}
53     CKB_PARSE_DOUBLE("x_offset", &x_offset) {}
54     CKB_PARSE_DOUBLE("y_offset", &y_offset) {}
55 }
```

13.35.2.6 void ckb_start (ckb_runctx * *context*, int *state*)

Definition at line 70 of file main.c.

References frame, ckb_runctx::height, ckb_runctx::width, x, x_offset, y, and y_offset.

```
70                                                {
71     // Begin or end animation
72     frame = state ? 0. : -1.;
73     x = (context->width + (context->width * x_offset * 0.01)) / 2.f;
74     y = (context->height - (context->height * y_offset * 0.01)) / 2.f;
75 }
```

13.35.2.7 void ckb_time (ckb_runctx * *context*, double *delta*)

Definition at line 77 of file main.c.

References frame.

```
77                                                  {
78     if(frame < 0.)
79         return;
80     // Spin the wheel
81     frame += delta;
82     if(frame > 1.)
83         frame -= 1.;
84 }
```

13.35.3 Variable Documentation

13.35.3.1 ckb_gradient animcolor = { 0 }

Definition at line 39 of file main.c.

13.35.3.2 double animlength = 0.

Definition at line 40 of file main.c.

Referenced by anim_add(), ckb_frame(), ckb_parameter(), and ckb_time().

13.35.3.3 int counter_clock = 0

Definition at line 42 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.35.3.4 double frame = -1.

Definition at line 65 of file main.c.

Referenced by ckb_frame(), ckb_start(), ckb_time(), and AnimSettingDialog::hLine().

13.35.3.5 int symmetric = 0

Definition at line 41 of file main.c.

Referenced by anim_add(), ckb_frame(), and ckb_parameter().

13.35.3.6 float x

Definition at line 66 of file main.c.

Referenced by anim_add(), KeyMap::byPosition(), ckb_frame(), ckb_parameter(), ckb_start(), cmd_dpi(), drop_add(), GradientDialogWidget::fillRect(), main(), GradientDialog::makeIcon(), GradientDialogWidget::mousePressEvent(), ColorMap::operator==(), os_mousemove(), GradientDialogWidget::paintEvent(), KeyWidget::paintEvent(), Kb::readNotify(), unz64local_getLong(), unz64local_getLong64(), unz64local_getShort(), GradientButton::updateImage(), zip64local_getLong(), zip64local_getLong64(), and zip64local_getShort().

13.35.3.7 double x_offset = 0

Definition at line 43 of file main.c.

Referenced by ckb_parameter(), and ckb_start().

13.35.3.8 float y

Definition at line 66 of file main.c.

Referenced by anim_add(), KeyMap::byPosition(), ckb_frame(), ckb_parameter(), ckb_start(), cmd_dpi(), drop_add(), GradientDialogWidget::fillRect(), main(), GradientDialog::makeIcon(), GradientDialogWidget::mousePressEvent(), os_mousemove(), GradientDialogWidget::paintEvent(), KeyWidget::paintEvent(), and Kb::readNotify().

13.35.3.9 double y_offset = 0

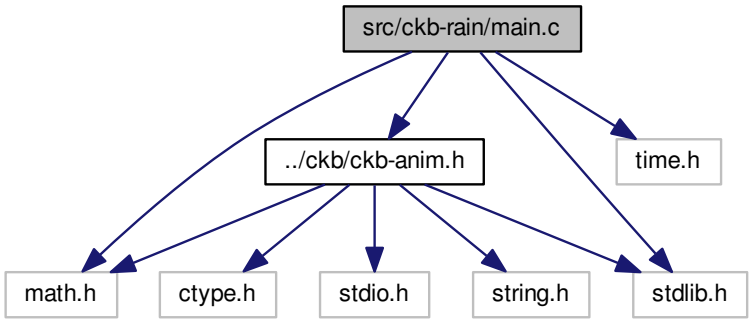
Definition at line 43 of file main.c.

Referenced by ckb_parameter(), and ckb_start().

13.36 src/ckb-rain/main.c File Reference

```
#include "../ckb/ckb-anim.h"
#include <math.h>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for main.c:



Macros

- #define DROP_MAX 1000

Functions

- void ckb_info ()
- void drop_add (float x, float y, int slow)
- void ckb_init (ckb_runctx *context)
- void ckb_parameter (ckb_runctx *context, const char *name, const char *value)
- void ckb_keypress (ckb_runctx *context, ckb_key *key, int x, int y, int state)
- void ckb_start (ckb_runctx *context, int state)
- void ckb_time (ckb_runctx *context, double delta)
- int ckb_frame (ckb_runctx *context)

Variables

- int spawn = 0
- double period = 0.1
- double maxsize = 100.
- double speed = 100.
- unsigned char ar = 255
- unsigned char ag = 255
- unsigned char ab = 255
- float aa = 1.
- struct {
  int **active**
  float **x**
  float **y**
  float **size**
  float **msize**
  } drop [1000]

- double tick = -1.

13.36.1 Macro Definition Documentation

13.36.1.1 #define DROP_MAX 1000

Definition at line 34 of file main.c.

Referenced by ckb_frame(), ckb_time(), and drop_add().

13.36.2 Function Documentation

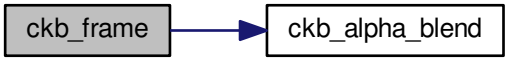
13.36.2.1 int ckb_frame (ckb_runctx * *context*)

Definition at line 120 of file main.c.

References aa, ab, ag, ar, ckb_alpha_blend(), CKB_KEYCLEAR, drop, DROP_MAX, ckb_runctx::keycount, and ckb_runctx::keys.

```
120                                 {
121     CKB_KEYCLEAR(context);
122     // Draw drops
123     for(unsigned i = 0; i < DROP_MAX; i++){
124         if(drop[i].active){
125             unsigned count = context->keycount;
126             ckb_key* keys = context->keys;
127             for(ckb_key* key = keys; key < keys + count; key++){
128                 // Calculate distance between key and drop, relative to the current drop size
129                 float distance = drop[i].size - sqrt(pow(key->x - drop[i].x, 2.f) + pow(
    key->y - drop[i].y, 2.f));
130                 // On the outside, cut the distance in half
131                 if(distance < 0.)
132                     distance = -distance / 2.;
133                 // Scale according to drop size (fade-out radius expands as the drop becomes larger)
134                 float scale = drop[i].size / 4.f;
135                 if(scale < 10.)
136                     scale = 10.;
137                 distance /= scale;
138                 if(distance <= 1.){
139                     // Scale alpha according to size divided by maximum size (drops fade out as they
     expand)
140                     float ascale = (1.f - drop[i].size / drop[i].msize);
141                     if(ascale > 1.f)
142                         ascale = 1.f;
143                     // Apply color
144                     ckb_alpha_blend(key, (1.f - distance) * ascale *
    aa * 255., ar, ag, ab);
145                 }
146             }
147         }
148     }
149     return 0;
150 }
```

Here is the call graph for this function:



13.36.2.2 void ckb_info ( )

Definition at line 6 of file main.c.

References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_POSITION, CKB_KPMODE, CKB_LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_DOUBLE, CKB_PARAM_LONG, CKB_PARAM_RGB, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_REPEAT, CKB_TIME_ABSOLUTE, CKB_TIMEMODE, CKB_VERSION, FALSE, and TRUE.

```
6               {
7     // Plugin info
8     CKB_NAME("Raindrop");
9     CKB_VERSION("0.10");
10     CKB_COPYRIGHT("2014-2016", "MSC");
11     CKB_LICENSE("GPLv2");
12     CKB_GUID("{5D6695AF-0496-41E2-BEE7-F7D0ABAA49E9}");
13     CKB_DESCRIPTION("A rain-like effect.");
14
15     // Effect parameters
16     CKB_PARAM_RGB("color", "Drop color:", "", 255, 255, 255);
17     CKB_PARAM_DOUBLE("intensity", "Intensity:", "%", 100, 0, 100);
18     CKB_PARAM_DOUBLE("size", "Size:", "%", 25, 1, 100);
19     CKB_PARAM_LONG("frequency", "Frequency:", "per second", 10, 1, 100);
20
21     // Timing/input parameters
22     CKB_KPMODE(CKB_KP_POSITION);
```

```
23      CKB_TIMEMODE(CKB_TIME_ABSOLUTE);
24      CKB_REPEAT(FALSE);
25      CKB_LIVEPARAMS(TRUE);
26
27      // Presets
28      CKB_PRESET_START("Default");
29      CKB_PRESET_PARAM("trigger", "1");
30      CKB_PRESET_PARAM("kptrigger", "1");
31      CKB_PRESET_END;
32  }
```

13.36.2.3 void ckb_init (ckb_runctx * *context*)

Definition at line 63 of file main.c.

```
63                                      {
64      srand(time(NULL));
65  }
```

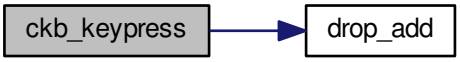
13.36.2.4 void ckb_keypress (ckb_runctx * *context*, ckb_key * *key*, int *x*, int *y*, int *state*)

Definition at line 86 of file main.c.

References drop_add().

```
86                                                                              {
87      // Add a drop on keypress
88      if(state)
89          drop_add(x, y, 0);
90  }
```

Here is the call graph for this function:



13.36.2.5 void ckb_parameter (ckb_runctx * *context*, const char * *name*, const char * *value*)

Definition at line 67 of file main.c.

References aa, ab, ag, ar, CKB_PARSE_BOOL, CKB_PARSE_DOUBLE, CKB_PARSE_LONG, CKB_PARSE_RGB, ckb_runctx::height, maxsize, period, spawn, speed, and ckb_runctx::width.

```
67                                                                              {
68      CKB_PARSE_RGB("color", &ar, &ag, &ab){}
69      double intensity, size;
70      long frequency;
71      CKB_PARSE_DOUBLE("intensity", &intensity){
72          aa = intensity / 100.;
73      }
74      CKB_PARSE_DOUBLE("size", &size){
75          double sizex = context->width / 2.;
76          double sizey = context->height / 2.;
77          maxsize = size / 100. * sqrt(sizex * sizex + sizey * sizey);
78          speed = maxsize;
79      }
```

```
80      CKB_PARSE_LONG("frequency", &frequency){
81          period = 1. / frequency;
82      }
83      CKB_PARSE_BOOL("trigger", &spawn){}
84  }
```

13.36.2.6 void ckb_start (ckb_runctx * *context*, int *state*)

Definition at line 94 of file main.c.

References tick.

```
94                                                  {
95      // Start or stop random spawning
96      tick = state ? 0. : -1.;
97  }
```

13.36.2.7 void ckb_time (ckb_runctx * *context*, double *delta*)

Definition at line 99 of file main.c.

References drop, drop_add(), DROP_MAX, ckb_runctx::height, period, spawn, speed, tick, and ckb_runctx::width.

```
99                                                    {
100     if(delta <= 0.)
101         return;
102     // Process existing drops
103     for(unsigned i = 0; i < DROP_MAX; i++){
104         if(drop[i].active){
105             drop[i].size += delta * speed;
106             if(drop[i].size > drop[i].msize)
107                 drop[i].active = 0;
108         }
109     }
110     if(tick >= 0.){
111         // Spawn a new randomly-placed drop, if the spawn time has passed
112         tick += delta;
113         if(tick > period && spawn){
114             drop_add(rand() / (double)RAND_MAX * context->width, rand() / (double)RAND_MAX *
      context->height, 1);
115             tick -= period;
116         }
117     }
118 }
```

Here is the call graph for this function:

13.36.2.8 void drop_add (float *x*, float *y*, int *slow*)

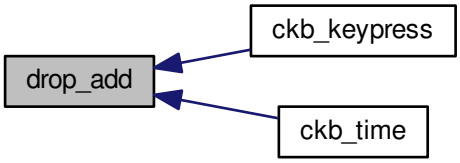
Definition at line 49 of file main.c.

References drop, DROP_MAX, maxsize, x, and y.

Referenced by ckb_keypress(), and ckb_time().

```
49                                      {
50     for(int i = 0; i < DROP_MAX; i++){
51         if(drop[i].active)
52             continue;
53         drop[i].active = 1;
54         drop[i].x = x;
55         drop[i].y = y;
56         float msize = maxsize * (0.9 + (rand() / (double)RAND_MAX * 0.2));
57         drop[i].size = -msize / 2. * slow;
58         drop[i].msize = msize;
59         return;
60     }
61 }
```

Here is the caller graph for this function:



13.36.3 Variable Documentation

13.36.3.1 float aa = 1.

Definition at line 41 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.36.3.2 unsigned char ab = 255

Definition at line 40 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.36.3.3 unsigned char ag = 255

Definition at line 40 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.36.3.4 unsigned char ar = 255

Definition at line 40 of file main.c.

Referenced by ckb_frame(), and ckb_parameter().

13.36.3.5 struct { ... } drop[1000]

Referenced by ckb_frame(), ckb_time(), and drop_add().

13.36.3.6 double maxsize = 100.

Definition at line 38 of file main.c.

Referenced by ckb_parameter(), ckb_time(), and drop_add().

13.36.3.7 double period = 0.1

Definition at line 37 of file main.c.

Referenced by ckb_parameter(), and ckb_time().

13.36.3.8 int spawn = 0

Definition at line 36 of file main.c.

Referenced by ckb_parameter(), and ckb_time().

13.36.3.9 double speed = 100.

Definition at line 39 of file main.c.

Referenced by ckb_parameter(), and ckb_time().

13.36.3.10 double tick = -1.

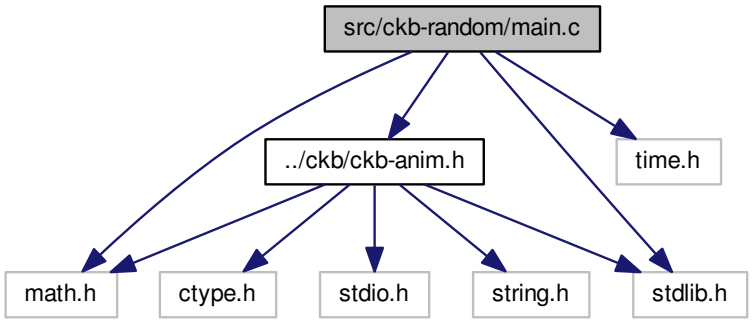
Definition at line 92 of file main.c.

Referenced by ckb_start(), and ckb_time().

13.37 src/ckb-random/main.c File Reference

```
#include "../ckb/ckb-anim.h"
#include <math.h>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for main.c:



Data Structures

- struct rgb

Functions

- void ckb_info ()
- void ckb_parameter (ckb_runctx *context, const char *name, const char *value)
- void ckb_keypress (ckb_runctx *context, ckb_key *key, int x, int y, int state)
- void newtarget (rgb *data, unsigned count)
- void ckb_init (ckb_runctx *context)
- void ckb_start (ckb_runctx *context, int state)
- void ckb_time (ckb_runctx *context, double delta)
- int ckb_frame (ckb_runctx *context)

Variables

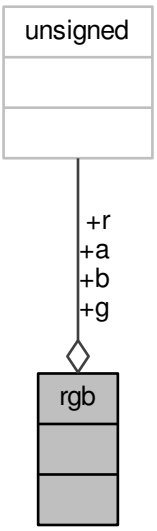
- int fadein = 0
- int useopacity = 0
- rgb * current = 0
- rgb * target = 0
- double phase = -1.

13.37.1 Data Structure Documentation

13.37.1.1 struct rgb

Definition at line 42 of file main.c.

Collaboration diagram for rgb:



Data Fields

unsigned char	a	
unsigned char	b	
unsigned char	g	
unsigned char	r	

13.37.2 Function Documentation

13.37.2.1 int ckb_frame (ckb_runctx * *context*)

Definition at line 109 of file main.c.

References ckb_key::a, ckb_key::b, CKB_KEYCLEAR, ckb_key::g, ckb_runctx::keycount, ckb_runctx::keys, phase, and ckb_key::r.

```
109                                 {
110     if(phase < 0.){
111         CKB_KEYCLEAR(context);
112         return 0;
113     }
114     ckb_key* keys = context->keys;
115     unsigned count = context->keycount;
116     for(unsigned i = 0; i < count; i++){
117         // Color each key according to the position between the last color set and the new color set
118         ckb_key* key = keys + i;
119         key->a = round(current[i].a * (1. - phase) + target[i].a *
    phase);
120         key->r = round(current[i].r * (1. - phase) + target[i].r *
    phase);
121         key->g = round(current[i].g * (1. - phase) + target[i].g *
    phase);
122         key->b = round(current[i].b * (1. - phase) + target[i].b *
    phase);
123     }
124     return 0;
125 }
```

13.37.2.2 void ckb_info ( )

Definition at line 6 of file main.c.

References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_NONE, CKB_KPMODE, CKB_LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_BOOL, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_REPEAT, CKB_TIME_DURATION, CKB_TIMEMODE, CKB_VERSION, FALSE, and TRUE.

```
6               {
7     // Plugin info
8     CKB_NAME("Random");
9     CKB_VERSION("0.9");
10     CKB_COPYRIGHT("2014-2016", "MSC");
11     CKB_LICENSE("GPLv2");
12     CKB_GUID("{22418DA4-A181-4B93-A4D3-03682BA283D2}");
13     CKB_DESCRIPTION("An effect that changes key colors randomly.");
14
15     // Effect parameters
16     CKB_PARAM_BOOL("fade", "Fade in", 0);
17     CKB_PARAM_BOOL("useopacity", "Randomize opacity", 0);
18
19     // Timing/input parameters
20     CKB_KPMODE(CKB_KP_NONE);
21     CKB_TIMEMODE(CKB_TIME_DURATION);
22     CKB_LIVEPARAMS(TRUE);
23     CKB_REPEAT(FALSE);
24
25     // Presets
26     CKB_PRESET_START("Default");
27     CKB_PRESET_PARAM("duration", "1.");
28     CKB_PRESET_END;
29 }
```

13.37.2.3 void ckb_init (ckb_runctx * *context*)

Definition at line 61 of file main.c.

References ckb_runctx::keycount.

```
61                                  {
62     unsigned count = context->keycount;
63     current = malloc(count * sizeof(rgb));
64     target = malloc(count * sizeof(rgb));
65     srand(time(NULL));
66 }
```

13.37.2.4 void ckb_keypress (ckb_runctx * *context,* ckb_key * *key,* int *x,* int *y,* int *state*)

Definition at line 38 of file main.c.

```
38                                                                          {
39     // Unused
40 }
```

13.37.2.5 void ckb_parameter (ckb_runctx * *context,* const char * *name,* const char * *value*)

Definition at line 33 of file main.c.

References CKB_PARSE_BOOL, fadein, and useopacity.

```
33                                                                          {
34     CKB_PARSE_BOOL("fade", &fadein) {}
35     CKB_PARSE_BOOL("useopacity", &useopacity) {}
36 }
```

13.37.2.6 void ckb_start (ckb_runctx * *context,* int *state*)

Definition at line 68 of file main.c.

References rgb::a, ckb_key::a, rgb::b, ckb_key::b, fadein, rgb::g, ckb_key::g, ckb_runctx::keycount, ckb_runctx::keys, newtarget(), phase, rgb::r, and ckb_key::r.

```
68                                          {
69     if(state == 0){
70         // Stop animation
71         phase = -1.;
72         return;
73     }
74     // Start animation
75     phase = 0.;
76     ckb_key* keys = context->keys;
77     unsigned count = context->keycount;
78     // Randomize starting colors and pick a random target
79     newtarget(current, count);
80     // Over the course of the animation, the keys fade between the current pattern and the target pattern
81     newtarget(target, count);
82     // Set all keys to current
83     for(unsigned i = 0; i < count; i++){
84         ckb_key* key = keys + i;
85         if(fadein)
86             current[i].a = 0;
87         key->a = current[i].a;
88         key->r = current[i].r;
89         key->g = current[i].g;
90         key->b = current[i].b;
91     }
92 }
```

Here is the call graph for this function:

13.37.2.7 void ckb_time (ckb_runctx * *context,* double *delta*)

Definition at line 94 of file main.c.

References current, ckb_runctx::keycount, newtarget(), phase, and target.

```
94                                                  {
95      if(phase < 0.)
96          return;
97      // Advance animation
98      phase += delta;
99      if(phase > 1.){
100         // If the animation is complete, pick a new target pattern and start again
101         phase -= 1.;
102         rgb* temp = target;
103         target = current;
104         current = temp;
105         newtarget(target, context->keycount);
106     }
107 }
```

Here is the call graph for this function:

13.37.2.8 void newtarget (rgb * *data,* unsigned *count*)

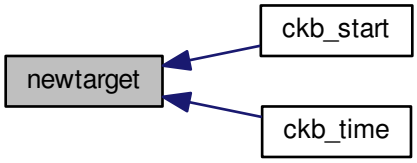
Definition at line 51 of file main.c.

References rgb::a, rgb::b, rgb::g, rgb::r, and useopacity.

Referenced by ckb_start(), and ckb_time().

```
51                                              {
52      for(unsigned i = 0; i < count; i++){
53          rgb* key = data + i;
54          key->a = useopacity ? rand() % 256 : 255;
55          key->r = rand() % 256;
56          key->g = rand() % 256;
57          key->b = rand() % 256;
58      }
59 }
```

Here is the caller graph for this function:



13.37.3 Variable Documentation

13.37.3.1 rgb* current = 0

Definition at line 46 of file main.c.

Referenced by CkbSettings::childGroups(), CkbSettings::childKeys(), ckb_time(), KbProfile::KbProfile(), Kb::load(), GradientDialog::on_presetDelete_clicked(), KbWidget::profileChanged(), KbProfileDialog::repopulate(), QuaZip::setCurrentFile(), and GradientDialog::setPreset().

13.37.3.2 int fadein = 0

Definition at line 31 of file main.c.

Referenced by ckb_parameter(), and ckb_start().

13.37.3.3 double phase = -1.

Definition at line 48 of file main.c.

Referenced by ckb_frame(), ckb_start(), ckb_time(), kiss_fft_alloc(), and kiss_fftr_alloc().

13.37.3.4 rgb* target = 0

Definition at line 47 of file main.c.

Referenced by ckb_time().

13.37.3.5 int useopacity = 0

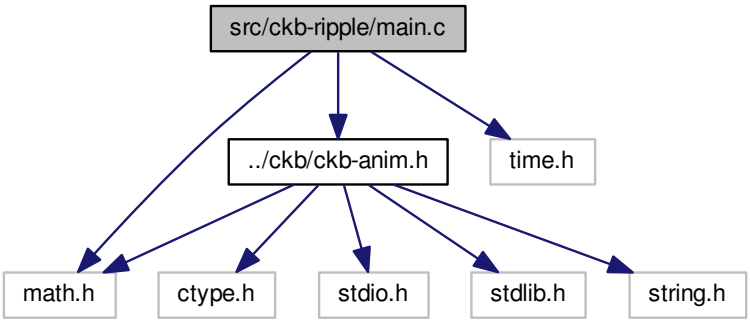
Definition at line 31 of file main.c.

Referenced by ckb_parameter(), and newtarget().

13.38 src/ckb-ripple/main.c File Reference

```
#include "../ckb/ckb-anim.h"
#include <math.h>
#include <time.h>
```

Include dependency graph for main.c:



Macros

- #define ANIM_MAX (144 ∗ 2)

Functions

- void ckb_info ()
- void ckb_init (ckb_runctx ∗context)
- void ckb_parameter (ckb_runctx ∗context, const char ∗name, const char ∗value)
- void anim_add (float x, float y, float width, float height)
- void anim_remove (float x, float y)
- void ckb_keypress (ckb_runctx ∗context, ckb_key ∗key, int x, int y, int state)
- void ckb_start (ckb_runctx ∗context, int state)
- void ckb_time (ckb_runctx ∗context, double delta)
- int ckb_frame (ckb_runctx ∗context)

Variables

- float kbsize = 0.f
- ckb_gradient animcolor = { 0 }
- int symmetric = 0
- int kprelease = 0
- int randomize = 0
- double animlength = 0.
- double x_offset = 0
- double y_offset = 0
- struct {
  int **active**
  float **x**
  float **y**
  float **maxsize**
  float **cursize**
  float **choice**
  } anim [(144 ∗2)] = { }

13.38.1 Macro Definition Documentation

13.38.1.1 #define ANIM_MAX (144 * 2)

Definition at line 75 of file main.c.

Referenced by anim_add(), anim_remove(), ckb_frame(), and ckb_time().

13.38.2 Function Documentation

13.38.2.1 void anim_add (float *x*, float *y*, float *width*, float *height*)

Definition at line 84 of file main.c.

References anim, ANIM_MAX, animlength, symmetric, x, and y.

```
84                                                          {
85     for(int i = 0; i < ANIM_MAX; i++){
86         if(anim[i].active)
87             continue;
88         anim[i].active = 1;
89         anim[i].x = x;
90         anim[i].y = y;
91         float sizex = fmax(x, width - x);
92         float sizey = fmax(y, height - y);
93         anim[i].maxsize = sqrt(sizex * sizex + sizey * sizey) + animlength;
94         anim[i].cursize = (symmetric) ? -animlength : 0;
95         anim[i].choice = (float)rand()/(float)(RAND_MAX);
96         return;
97     }
98 }
```

13.38.2.2 void anim_remove (float *x*, float *y*)

Definition at line 100 of file main.c.

References anim, and ANIM_MAX.

```
100                                    {
101     for(int i = 0; i < ANIM_MAX; i++){
102         if(anim[i].active && anim[i].x == x && anim[i].y == y)
103             anim[i].active = 0;
104     }
105 }
```

13.38.2.3 int ckb_frame (ckb_runctx * *context*)

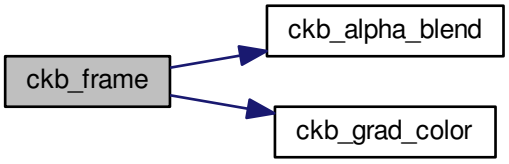
Definition at line 134 of file main.c.

References anim, ANIM_MAX, animlength, ckb_alpha_blend(), ckb_grad_color(), CKB_KEYCLEAR, ckb_runctx::keycount, ckb_runctx::keys, randomize, and symmetric.

```
134                                    {
135     CKB_KEYCLEAR(context);
136     // Draw rings
137     unsigned count = context->keycount;
138     ckb_key* keys = context->keys;
139     for(unsigned i = 0; i < ANIM_MAX; i++){
140         if(anim[i].active){
141             for(ckb_key* key = keys; key < keys + count; key++){
142                 // Calculate distance between this key and the ring
143                 float distance = anim[i].cursize - sqrt(pow(key->x - anim[i].x, 2.f) + pow(
      key->y - anim[i].y, 2.f));
144                 // Divide distance by ring size (use absolute distance if symmetric)
145                 distance /= animlength;
146                 if(symmetric)
147                     distance = fabs(distance);
148                 else if(distance >= -0.005f && distance < 0.f)
```

```
149                     // If not symmetric, round down values very close to zero
150                     distance = 0.f;
151                 if(distance > 1.f && distance <= 1.005f)
152                     // Round values close to 1
153                     distance = 1.f;
154
155                 // Blend color gradient according to position
156                 if(distance >= 0. && distance <= 1.f){
157                     float a, r, g, b;
158                     float gradChoice = randomize ? anim[i].choice : distance;
159                     ckb_grad_color(&a, &r, &g, &b, &animcolor, gradChoice * 100.);
160                     ckb_alpha_blend(key, a, r, g, b);
161                 }
162             }
163         }
164     }
165     return 0;
166 }
```

Here is the call graph for this function:



13.38.2.4 void ckb_info ( )

Definition at line 5 of file main.c.

References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_POSITION, CKB_KPMODE, CKB_-LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, CKB_PARAM_BOOL, CKB_PARAM_-DOUBLE, CKB_PREEMPT, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_TIME_-DURATION, CKB_TIMEMODE, CKB_VERSION, and TRUE.

```
5               {
6     // Plugin info
7     CKB_NAME("Ripple");
8     CKB_VERSION("0.10");
9     CKB_COPYRIGHT("2014-2016", "MSC");
10     CKB_LICENSE("GPLv2");
11     CKB_GUID("{097D69F0-70B2-48B8-AFE2-25A1CDB02C9D}");
12     CKB_DESCRIPTION("An expanding ring effect.");
13
14     // Effect parameters
15     CKB_PARAM_AGRADIENT("color", "Ripple color:", "", "ffffffff");
16     CKB_PARAM_DOUBLE("length", "Ring length:", "%", 100, 1, 100);
17     CKB_PARAM_DOUBLE("x_offset", "X offset:", "%", 0, -100, 100);
18     CKB_PARAM_DOUBLE("y_offset", "Y offset:", "%", 0, -100, 100);
19     CKB_PARAM_BOOL("symmetric", "Symmetric", 0);
20     CKB_PARAM_BOOL("randomize", "Randomly select from gradient", 0);
21
22     // Timing/input parameters
23     CKB_KPMODE(CKB_KP_POSITION);
24     CKB_TIMEMODE(CKB_TIME_DURATION);
25     CKB_LIVEPARAMS(TRUE);
26     CKB_PREEMPT(TRUE);
27
28     // Presets
29     CKB_PRESET_START("Shimmer");
30     CKB_PRESET_PARAM("duration", "2.0");
31     CKB_PRESET_PARAM("length", "50.0");
32     CKB_PRESET_PARAM("symmetric", "1");
```

```
33     CKB_PRESET_END;
34
35     CKB_PRESET_START("Rainbow");
36     CKB_PRESET_PARAM("color", "0:ffff0000 17:ffffff00 33:ff00ff00 50:ff00ffff 67:ff0000ff
       83:ffff00ff 100:ffff0000");
37     CKB_PRESET_PARAM("duration", "2.0");
38     CKB_PRESET_END;
39
40     CKB_PRESET_START("Type lighting");
41     CKB_PRESET_PARAM("duration", "1.0");
42     CKB_PRESET_PARAM("length", "25.0");
43     CKB_PRESET_PARAM("symmetric", "1");
44     CKB_PRESET_PARAM("trigger", "0");
45     CKB_PRESET_PARAM("kptrigger", "1");
46     CKB_PRESET_END;
47 }
```

13.38.2.5 void ckb_init (ckb_runctx * *context*)

Definition at line 55 of file main.c.

References ckb_runctx::height, kbsize, and ckb_runctx::width.

```
55                                  {
56     kbsize = sqrt(context->width * context->width / 4.f + context->
       height * context->height / 4.f);
57     srand((unsigned)time(NULL));
58 }
```

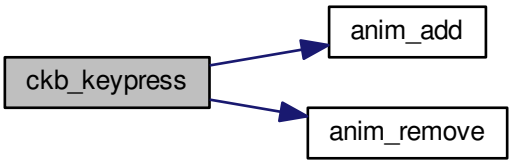
13.38.2.6 void ckb_keypress (ckb_runctx * *context,* ckb_key * *key,* int *x,* int *y,* int *state*)

Definition at line 107 of file main.c.

References anim_add(), anim_remove(), ckb_runctx::height, kprelease, and ckb_runctx::width.

```
107                                                                                 {
108     // Add or remove a ring on this key
109     if(state)
110         anim_add(x, y, context->width, context->height);
111     else if(kprelease)
112         anim_remove(x, y);
113 }
```

Here is the call graph for this function:



13.38.2.7 void ckb_parameter (ckb_runctx * *context,* const char * *name,* const char * *value*)

Definition at line 60 of file main.c.

References animlength, CKB_PARSE_AGRADIENT, CKB_PARSE_BOOL, CKB_PARSE_DOUBLE, ckb_runctx::height, kprelease, randomize, symmetric, ckb_runctx::width, x_offset, and y_offset.

```
60                                                                                  {
61     CKB_PARSE_AGRADIENT("color", &animcolor){}
62     double len;
63     CKB_PARSE_DOUBLE("length", &len){
64         double sizex = context->width / 2.;
65         double sizey = context->height / 2.;
66         animlength = len / 100. * sqrt(sizex * sizex + sizey * sizey);
67     }
68     CKB_PARSE_BOOL("symmetric", &symmetric){}
69     CKB_PARSE_BOOL("kprelease", &kprelease){}
70     CKB_PARSE_BOOL("randomize", &randomize){}
71     CKB_PARSE_DOUBLE("x_offset", &x_offset) {}
72     CKB_PARSE_DOUBLE("y_offset", &y_offset) {}
73 }
```

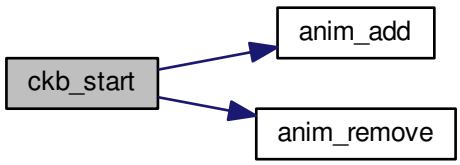
13.38.2.8 void ckb_start (ckb_runctx * *context*, int *state*)

Definition at line 115 of file main.c.

References anim_add(), anim_remove(), ckb_runctx::height, ckb_runctx::width, x_offset, and y_offset.

```
115                                                 {
116     // Add or remove a ring in the center of the keyboard
117     if(state)
118         anim_add((context->width + (context->width * x_offset * 0.01)) / 2.f, (
      context->height - (context->height * y_offset * 0.01)) / 2.f, context->
      width, context->height);
119     else
120         anim_remove(context->width / 2.f, context->height / 2.f);
121 }
```

Here is the call graph for this function:



13.38.2.9 void ckb_time (ckb_runctx * *context*, double *delta*)

Definition at line 123 of file main.c.

References anim, ANIM_MAX, kbsize, and maxsize.

```
123                                                 {
124     // Advance animation on all rings
125     for(unsigned i = 0; i < ANIM_MAX; i++){
126         if(anim[i].active){
127             anim[i].cursize += kbsize * delta;
128             if(anim[i].cursize > anim[i].maxsize)
129                 anim[i].active = 0;
130         }
131     }
132 }
```

13.38.3 Variable Documentation

13.38.3.1 struct { ... } anim[(144 * 2)]

13.38.3.2 ckb_gradient animcolor = { 0 }

Definition at line 50 of file main.c.

13.38.3.3 double animlength = 0.

Definition at line 52 of file main.c.

13.38.3.4 float kbsize = 0.f

Definition at line 49 of file main.c.

Referenced by ckb_init(), and ckb_time().

13.38.3.5 int kprelease = 0

Definition at line 51 of file main.c.

13.38.3.6 int randomize = 0

Definition at line 51 of file main.c.

13.38.3.7 int symmetric = 0

Definition at line 51 of file main.c.

13.38.3.8 double x_offset = 0

Definition at line 53 of file main.c.

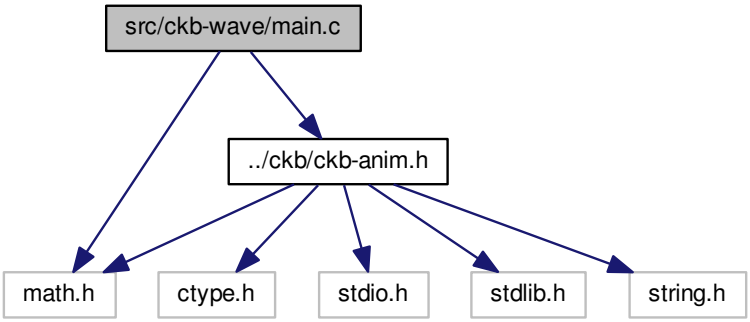
13.38.3.9 double y_offset = 0

Definition at line 53 of file main.c.

13.39 src/ckb-wave/main.c File Reference

```
#include "../ckb/ckb-anim.h"
#include <math.h>
```

Include dependency graph for main.c:



Macros

- #define ANIM_MAX (144 ∗ 2)

Functions

- void ckb_info ()
- void ckb_init (ckb_runctx ∗context)
- void ckb_parameter (ckb_runctx ∗context, const char ∗name, const char ∗value)
- void anim_add (float x, float y)
- void anim_remove (float x, float y)
- void ckb_keypress (ckb_runctx ∗context, ckb_key ∗key, int x, int y, int state)
- void ckb_start (ckb_runctx ∗context, int state)
- void ckb_time (ckb_runctx ∗context, double delta)
- int ckb_frame (ckb_runctx ∗context)

Variables

- ckb_gradient animcolor = { 0 }
- int symmetric = 0
- int kprelease = 0
- double angle = 0.
- double left = 0.
- double top = 0.
- double animlength = 0.
- double width = 0.
- struct {
  int **active**
  float **x**
  float **y**
  float **curx**
  } anim [(144 ∗2)] = { }

13.39.1 Macro Definition Documentation

13.39.1.1 #define ANIM_MAX (144 ∗ 2)

Definition at line 98 of file main.c.

Referenced by anim_add(), anim_remove(), ckb_frame(), and ckb_time().

13.39.2 Function Documentation

13.39.2.1 void anim_add (float *x*, float *y*)

Definition at line 105 of file main.c.

References anim, ANIM_MAX, animlength, symmetric, width, x, and y.

```
105                                {
106     for(int i = 0; i < ANIM_MAX; i++){
107         if(anim[i].active)
108             continue;
109         anim[i].active = 1;
110         anim[i].x = x;
111         anim[i].y = y;
112         anim[i].curx = symmetric ? -animlength * width : 0.f;
113         return;
114     }
115 }
```

13.39.2.2 void anim_remove (float *x*, float *y*)

Definition at line 117 of file main.c.

References anim, and ANIM_MAX.

```
117                                   {
118     for(int i = 0; i < ANIM_MAX; i++){
119         if(anim[i].active && anim[i].x == x && anim[i].y == y)
120             anim[i].active = 0;
121     }
122 }
```

13.39.2.3 int ckb_frame (ckb_runctx ∗ *context*)

Definition at line 152 of file main.c.

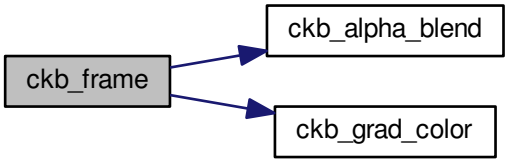
References angle, anim, ANIM_MAX, animlength, ckb_alpha_blend(), ckb_grad_color(), CKB_KEYCLEAR, ckb_runctx::keycount, ckb_runctx::keys, symmetric, width, x, and y.

Referenced by main().

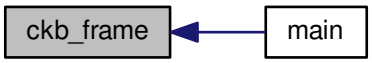
```
152                                   {
153     CKB_KEYCLEAR(context);
154     // Draw keys
155     double length = animlength * width;
156     unsigned count = context->keycount;
157     ckb_key* keys = context->keys;
158     for(unsigned i = 0; i < ANIM_MAX; i++){
159         if(anim[i].active){
160             for(ckb_key* key = keys; key < keys + count; key++){
161                 // Translate and rotate the key position into the animation's coordinate system
162                 float x = key->x - anim[i].x, y = key->y - anim[i].y;
163                 // Distance is the current X minus the key's X
164                 float distance = anim[i].curx - (x * cos(angle) - y * sin(
    angle));
165                 distance /= length;
166                 // If symmetric, use absolute distance
167                 if(symmetric)
168                     distance = fabs(distance);
```

```
169                 else if(distance >= -0.005f && distance < 0.f)
170                     // If not symmetric, round values close to 0
171                     distance = 0.f;
172                 if(distance > 1.f && distance <= 1.005f)
173                     // Round values close to 1
174                     distance = 1.f;
175                 // Pick gradient position based on distance
176                 if(distance <= 1.f && distance >= 0.){
177                     float a, r, g, b;
178                     ckb_grad_color(&a, &r, &g, &b, &animcolor, distance * 100.);
179                     ckb_alpha_blend(key, a, r, g, b);
180                 }
181             }
182         }
183     }
184     return 0;
185 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.39.2.4 void ckb_info ( )

Definition at line 4 of file main.c.

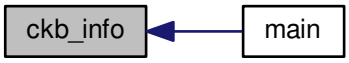
References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_POSITION, CKB_KPMODE, CKB_-LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, CKB_PARAM_ANGLE, CKB_PARAM-_BOOL, CKB_PARAM_DOUBLE, CKB_PREEMPT, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESE-T_START, CKB_TIME_DURATION, CKB_TIMEMODE, CKB_VERSION, and TRUE.

Referenced by main().

```
4               {
5     // Plugin info
6     CKB_NAME("Wave");
7     CKB_VERSION("0.10");
8     CKB_COPYRIGHT("2014-2016", "MSC");
9     CKB_LICENSE("GPLv2");
10     CKB_GUID("{E0BBA19E-C328-4C0E-8E3C-A06D5722B4FC}");
11     CKB_DESCRIPTION("A moving wave effect.");
```

```
12
13      // Effect parameters
14      CKB_PARAM_ANGLE("angle", "Angle:", "", 90);
15      CKB_PARAM_AGRADIENT("color", "Wave color:", "", "ffffffff");
16      CKB_PARAM_DOUBLE("length", "Wave length:", "%", 100, 1., 100.);
17      CKB_PARAM_BOOL("symmetric", "Symmetric", 0);
18
19      // Timing/input parameters
20      CKB_KPMODE(CKB_KP_POSITION);
21      CKB_TIMEMODE(CKB_TIME_DURATION);
22      CKB_LIVEPARAMS(TRUE);
23      CKB_PREEMPT(TRUE);
24
25      // Presets
26      CKB_PRESET_START("Shimmer");
27      CKB_PRESET_PARAM("duration", "2.0");
28      CKB_PRESET_PARAM("length", "50.0");
29      CKB_PRESET_PARAM("symmetric", "1");
30      CKB_PRESET_END;
31
32      CKB_PRESET_START("Rainbow");
33      CKB_PRESET_PARAM("color", "0:ffff0000 17:ffffff00 33:ff00ff00 50:ff00ffff 67:ff0000ff
        83:ffff00ff 100:ffff0000");
34      CKB_PRESET_PARAM("duration", "2.0");
35      CKB_PRESET_END;
36
37      CKB_PRESET_START("Vertical rainbow");
38      CKB_PRESET_PARAM("color", "0:ffff0000 17:ffffff00 33:ff00ff00 50:ff00ffff 67:ff0000ff
        83:ffff00ff 100:ffff0000");
39      CKB_PRESET_PARAM("duration", "2.0");
40      CKB_PRESET_PARAM("angle", "180");
41      CKB_PRESET_END;
42  }
```

Here is the caller graph for this function:



13.39.2.5 void ckb_init (ckb_runctx * *context*)

Definition at line 44 of file main.c.

Referenced by main().

```
44                                  {
45      // Nothing to do
46  }
```

Here is the caller graph for this function:

13.39.2.6 void ckb_keypress (ckb_runctx * *context,* ckb_key * *key,* int *x,* int *y,* int *state*)

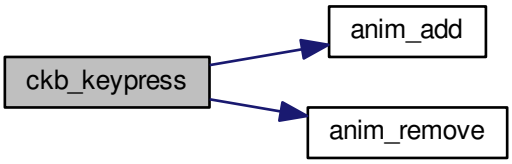
Definition at line 124 of file main.c.

References anim_add(), and anim_remove().

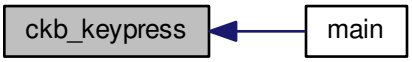
Referenced by main().

```
124                                                                                 {
125     // Start or stop animation on a key
126     if(state)
127         anim_add(x, y);
128     else
129         anim_remove(x, y);
130 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.39.2.7 void ckb_parameter (ckb_runctx * *context,* const char * *name,* const char * *value*)

Definition at line 54 of file main.c.

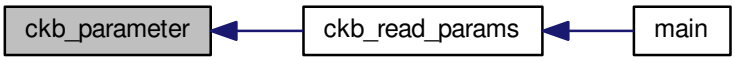
References angle, animlength, CKB_PARSE_AGRADIENT, CKB_PARSE_ANGLE, CKB_PARSE_BOOL, CKB_PARSE_DOUBLE, CKB_REAL_ANGLE, ckb_runctx::height, kprelease, left, symmetric, top, width, ckb_runctx::width, x, and y.

Referenced by ckb_read_params().

```
54                                                                                  {
55     CKB_PARSE_AGRADIENT("color", &animcolor){}
56     double len;
57     CKB_PARSE_DOUBLE("length", &len){
58         animlength = len / 100.;
59     }
60     CKB_PARSE_BOOL("symmetric", &symmetric){}
61     CKB_PARSE_BOOL("kprelease", &kprelease){}
62     long _angle;
```

```
63      CKB_PARSE_ANGLE("angle", &_angle){
64          angle = CKB_REAL_ANGLE(_angle);
65          // Get each of the four corners of the keyboard, relative to the center
66          double wOver2 = context->width / 2., hOver2 = context->height / 2.;
67          double x[4] = {
68              -wOver2, wOver2,
69              -wOver2, wOver2
70          };
71          double y[4] = {
72              -hOver2, -hOver2,
73              hOver2, hOver2
74          };
75          // Rotate them in the direction that the animation will travel
76          for(int i = 0; i < 4; i++){
77              double x2 = x[i] * cos(angle) - y[i] * sin(angle);
78              y[i] = x[i] * sin(angle) + y[i] * cos(angle);
79              x[i] = x2;
80          }
81          // Determine the leftmost and rightmost points
82          double min_x = INFINITY, max_x = -INFINITY;
83          for(int i = 0; i < 4; i++){
84              if(x[i] < min_x)
85                  min_x = x[i];
86              if(x[i] > max_x)
87                  max_x = x[i];
88          }
89          // The leftmost point is where the animation will begin when started with the mode,
90          // and the difference between the two is the animation width.
91          // Animations always travel left to right in their own coordinate system.
92          left = min_x * cos(-angle) + wOver2;
93          top = min_x * sin(-angle) + hOver2;
94          width = max_x - min_x;
95      }
96  }
```

Here is the caller graph for this function:



13.39.2.8 void ckb_start (ckb_runctx * *context*, int *state*)

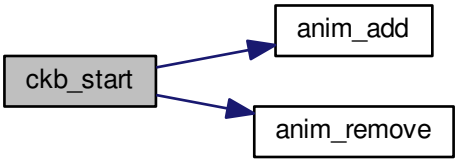
Definition at line 132 of file main.c.

References anim_add(), anim_remove(), kprelease, left, and top.

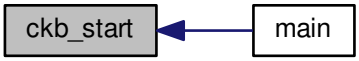
Referenced by main().

```
132                                             {
133      // Start or stop animation on the edge of the keyboard
134      if(state)
135          anim_add(left, top);
136      else if(kprelease)
137          anim_remove(left, top);
138  }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.39.2.9 void ckb_time (ckb_runctx * *context*, double *delta*)

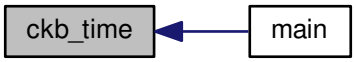
Definition at line 140 of file main.c.

References anim, ANIM_MAX, animlength, and width.

Referenced by main().

```
140                                                 {
141     // Advance all waves, remove them if they pass the edge
142     double length = animlength * width;
143     for(unsigned i = 0; i < ANIM_MAX; i++){
144         if(anim[i].active){
145             anim[i].curx += width * delta;
146             if(anim[i].curx > width + length)
147                 anim[i].active = 0;
148         }
149     }
150 }
```

Here is the caller graph for this function:



13.39.3 Variable Documentation

13.39.3.1 double angle = 0.

Definition at line 50 of file main.c.

Referenced by angleFlip(), ckb_frame(), and ckb_parameter().

13.39.3.2 struct { ... } anim[(144 * 2)]

13.39.3.3 ckb_gradient animcolor = { 0 }

Definition at line 48 of file main.c.

13.39.3.4 double animlength = 0.

Definition at line 52 of file main.c.

13.39.3.5 int kprelease = 0

Definition at line 49 of file main.c.

13.39.3.6 double left = 0.

Definition at line 51 of file main.c.

Referenced by _cmd_macro(), ckb_parameter(), ckb_start(), cmd_dpi(), GradientDialogWidget::mouseMoveEvent(), readcmd(), and right().

13.39.3.7 int symmetric = 0

Definition at line 49 of file main.c.

13.39.3.8 double top = 0.

Definition at line 51 of file main.c.

Referenced by ckb_parameter(), ckb_start(), and GradientDialogWidget::mouseMoveEvent().

13.39.3.9 double width = 0.

Definition at line 52 of file main.c.

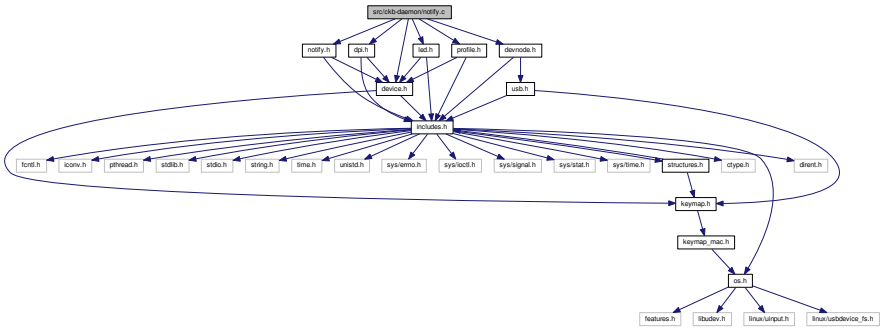
Referenced by anim_add(), ckb_frame(), ckb_parameter(), ckb_time(), KeyWidget::drawInfo(), GradientDialogWidget::fillRect(), KeyWidget::map(), and KeyWidget::paintEvent().

13.40 src/ckb-daemon/notify.c File Reference

```
#include "device.h"
```

```
#include "devnode.h"
#include "dpi.h"
#include "led.h"
#include "notify.h"
#include "profile.h"
```

Include dependency graph for notify.c:



Macros

- #define HWMODE_OR_RETURN(kb, index)
- #define HW_STANDARD

Functions

- void nprintf (usbdevice *kb, int nodenumber, usbmode *mode, const char *format,...)
- void nprintkey (usbdevice *kb, int nnumber, int keyindex, int down)
- void nprintind (usbdevice *kb, int nnumber, int led, int on)
- void cmd_notify (usbdevice *kb, usbmode *mode, int nnumber, int keyindex, const char *toggle)
- static void _cmd_get (usbdevice *kb, usbmode *mode, int nnumber, const char *setting)
- void cmd_get (usbdevice *kb, usbmode *mode, int nnumber, int dummy, const char *setting)
- int restart ()
- void cmd_restart (usbdevice *kb, usbmode *mode, int nnumber, int dummy, const char *content)

13.40.1 Macro Definition Documentation

13.40.1.1 #define HW_STANDARD

Value:

```
if(!kb->hw)                                     \
        return;                                 \
    unsigned index = INDEX_OF(mode, profile->mode); \
    /* Make sure the mode number is valid */    \
    HWMODE_OR_RETURN(kb, index)
```

Definition at line 83 of file notify.c.

Referenced by _cmd_get().

13.40.1.2 #define HWMODE_OR_RETURN(*kb, index*)

Value:

```
if(IS_K95(kb)){                 \
        if((index) >= HWMODE_K95)   \
            return;                 \
    } else {                        \
        if((index) >= HWMODE_K70)   \
            return;                 \
    }
```

Definition at line 73 of file notify.c.

13.40.2 Function Documentation

13.40.2.1 static void _cmd_get (usbdevice * *kb,* usbmode * *mode,* int *nnumber,* const char * *setting*) [static]

Definition at line 90 of file notify.c.

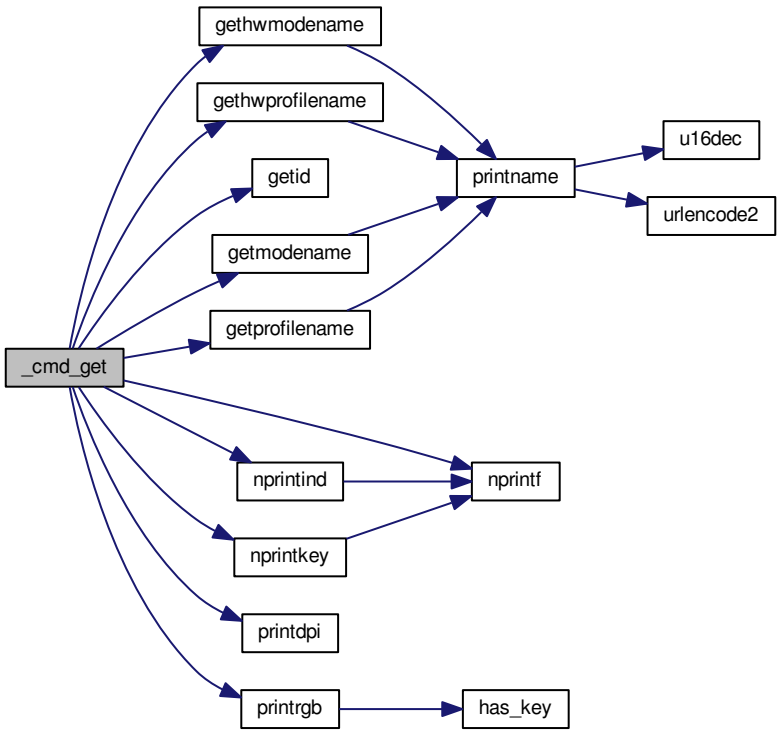
References dpiset::current, usbmode::dpi, hwprofile::dpi, gethwmodename(), gethwprofilename(), getid(), getmodename(), getprofilename(), usbdevice::hw, usbdevice::hw_ileds, HW_STANDARD, I_CAPS, I_NUM, I_SCROLL, usbmode::id, usbprofile::id, hwprofile::id, usbdevice::input, keymap, usbinput::keys, dpiset::lift, usbmode::light, hwprofile::light, usbid::modified, N_KEYS_INPUT, nprintf(), nprintind(), nprintkey(), printdpi(), printrgb(), usbdevice::profile, and dpiset::snap.

Referenced by cmd_get().

```
90                                                                                  {
91      usbprofile* profile = kb->profile;
92      if(!strcmp(setting, ":mode")){
93          // Get the current mode number
94          nprintf(kb, nnumber, mode, "switch\n");
95          return;
96      } else if(!strcmp(setting, ":rgb")){
97          // Get the current RGB settings
98          char* rgb = printrgb(&mode->light, kb);
99          nprintf(kb, nnumber, mode, "rgb %s\n", rgb);
100         free(rgb);
101         return;
102     } else if(!strcmp(setting, ":hwrgb")){
103         // Get the current hardware RGB settings
104         HW_STANDARD;
105         char* rgb = printrgb(kb->hw->light + index, kb);
106         nprintf(kb, nnumber, mode, "hwrgb %s\n", rgb);
107         free(rgb);
108         return;
109     } else if(!strcmp(setting, ":profilename")){
110         // Get the current profile name
111         char* name = getprofilename(profile);
112         nprintf(kb, nnumber, 0, "profilename %s\n", name[0] ? name : "Unnamed");
113         free(name);
114     } else if(!strcmp(setting, ":name")){
115         // Get the current mode name
116         char* name = getmodename(mode);
117         nprintf(kb, nnumber, mode, "name %s\n", name[0] ? name : "Unnamed");
118         free(name);
119     } else if(!strcmp(setting, ":hwprofilename")){
120         // Get the current hardware profile name
121         if(!kb->hw)
122             return;
123         char* name = gethwprofilename(kb->hw);
124         nprintf(kb, nnumber, 0, "hwprofilename %s\n", name[0] ? name : "Unnamed");
125         free(name);
126     } else if(!strcmp(setting, ":hwname")){
127         // Get the current hardware mode name
128         HW_STANDARD;
129         char* name = gethwmodename(kb->hw, index);
130         nprintf(kb, nnumber, mode, "hwname %s\n", name[0] ? name : "Unnamed");
131         free(name);
132     } else if(!strcmp(setting, ":profileid")){
133         // Get the current profile ID
134         char* guid = getid(&profile->id);
135         int modified;
136         memcpy(&modified, &profile->id.modified, sizeof(modified));
137         nprintf(kb, nnumber, 0, "profileid %s %x\n", guid, modified);
```

```
138         free(guid);
139     } else if(!strcmp(setting, ":id")){
140         // Get the current mode ID
141         char* guid = getid(&mode->id);
142         int modified;
143         memcpy(&modified, &mode->id.modified, sizeof(modified));
144         nprintf(kb, nnumber, mode, "id %s %x\n", guid, modified);
145         free(guid);
146     } else if(!strcmp(setting, ":hwprofileid")){
147         // Get the current hardware profile ID
148         if(!kb->hw)
149             return;
150         char* guid = getid(&kb->hw->id[0]);
151         int modified;
152         memcpy(&modified, &kb->hw->id[0].modified, sizeof(modified));
153         nprintf(kb, nnumber, 0, "hwprofileid %s %x\n", guid, modified);
154         free(guid);
155     } else if(!strcmp(setting, ":hwid")){
156         // Get the current hardware mode ID
157         HW_STANDARD;
158         char* guid = getid(&kb->hw->id[index + 1]);
159         int modified;
160         memcpy(&modified, &kb->hw->id[index + 1].modified, sizeof(modified));
161         nprintf(kb, nnumber, mode, "hwid %s %x\n", guid, modified);
162         free(guid);
163     } else if(!strcmp(setting, ":keys")){
164         // Get the current state of all keys
165         for(int i = 0; i < N_KEYS_INPUT; i++){
166             if(!keymap[i].name)
167                 continue;
168             int byte = i / 8, bit = 1 << (i & 7);
169             uchar state = kb->input.keys[byte] & bit;
170             if(state)
171                 nprintkey(kb, nnumber, i, 1);
172         }
173     } else if(!strcmp(setting, ":i")){
174         // Get the current state of all indicator LEDs
175         if(kb->hw_ileds & I_NUM) nprintind(kb, nnumber,
    I_NUM, 1);
176         if(kb->hw_ileds & I_CAPS) nprintind(kb, nnumber,
    I_CAPS, 1);
177         if(kb->hw_ileds & I_SCROLL) nprintind(kb, nnumber,
    I_SCROLL, 1);
178     } else if(!strcmp(setting, ":dpi")){
179         // Get the current DPI levels
180         char* dpi = printdpi(&mode->dpi, kb);
181         nprintf(kb, nnumber, mode, "dpi %s\n", dpi);
182         free(dpi);
183         return;
184     } else if(!strcmp(setting, ":hwdpi")){
185         // Get the current hardware DPI levels
186         HW_STANDARD;
187         char* dpi = printdpi(kb->hw->dpi + index, kb);
188         nprintf(kb, nnumber, mode, "hwdpi %s\n", dpi);
189         free(dpi);
190         return;
191     } else if(!strcmp(setting, ":dpisel")){
192         // Get the currently-selected DPI
193         nprintf(kb, nnumber, mode, "dpisel %d\n", mode->dpi.current);
194     } else if(!strcmp(setting, ":hwdpisel")){
195         // Get the currently-selected hardware DPI
196         HW_STANDARD;
197         nprintf(kb, nnumber, mode, "hwdpisel %d\n", kb->hw->dpi[index].
    current);
198     } else if(!strcmp(setting, ":lift")){
199         // Get the mouse lift height
200         nprintf(kb, nnumber, mode, "lift %d\n", mode->dpi.lift);
201     } else if(!strcmp(setting, ":hwlift")){
202         // Get the hardware lift height
203         HW_STANDARD;
204         nprintf(kb, nnumber, mode, "hwlift %d\n", kb->hw->dpi[index].
    lift);
205     } else if(!strcmp(setting, ":snap")){
206         // Get the angle snap status
207         nprintf(kb, nnumber, mode, "snap %s\n", mode->dpi.snap ? "on" : "off");
208     } else if(!strcmp(setting, ":hwsnap")){
209         // Get the hardware angle snap status
210         HW_STANDARD;
211         nprintf(kb, nnumber, mode, "hwsnap %s\n", kb->hw->dpi[index].
    snap ? "on" : "off");
212     }
213 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

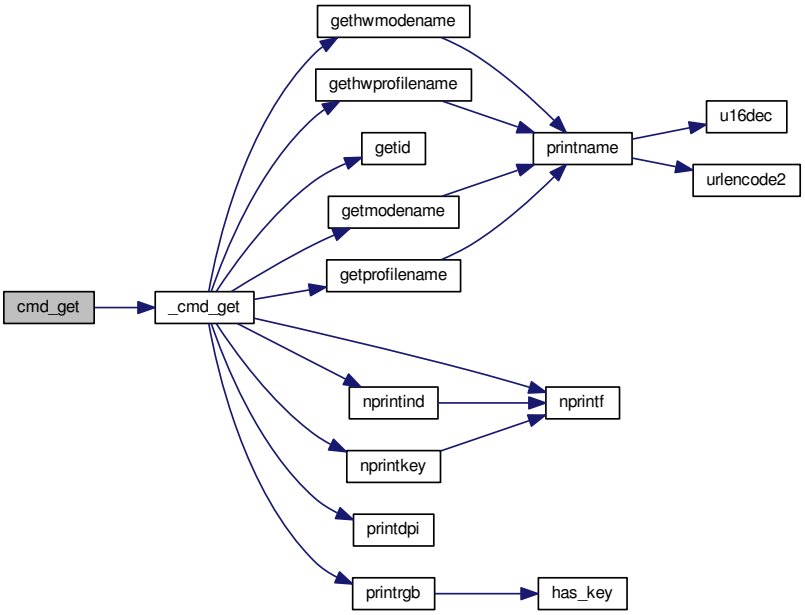
13.40.2.2 void cmd_get (usbdevice * *kb*, usbmode * *mode*, int *nnumber*, int *dummy*, const char * *setting*)

Definition at line 215 of file notify.c.

References _cmd_get(), and imutex.

```
215                                                                                 {
216     (void)dummy;
217
218     pthread_mutex_lock(imutex(kb));
219     _cmd_get(kb, mode, nnumber, setting);
220     pthread_mutex_unlock(imutex(kb));
221 }
```

Here is the call graph for this function:



13.40.2.3 void cmd_notify (usbdevice * *kb*, usbmode * *mode*, int *nnumber*, int *keyindex*, const char * *toggle*)

Definition at line 61 of file notify.c.

References CLEAR_KEYBIT, imutex, N_KEYS_INPUT, usbmode::notify, and SET_KEYBIT.

```
61                                                                                                  {
62     if(keyindex >= N_KEYS_INPUT)
63         return;
64     pthread_mutex_lock(imutex(kb));
65     if(!strcmp(toggle, "on") || *toggle == 0)
66         SET_KEYBIT(mode->notify[nnumber], keyindex);
67     else if(!strcmp(toggle, "off"))
68         CLEAR_KEYBIT(mode->notify[nnumber], keyindex);
69     pthread_mutex_unlock(imutex(kb));
70 }
```

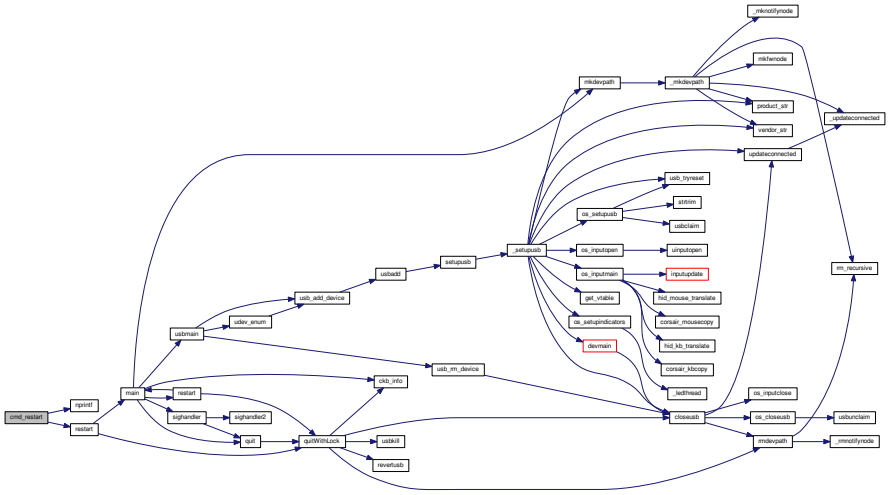
13.40.2.4 void cmd_restart (usbdevice * *kb*, usbmode * *mode*, int *nnumber*, int *dummy*, const char * *content*)

Definition at line 225 of file notify.c.

References ckb_info, nprintf(), and restart().

```
225                                                                                                 {
226     (void)mode;
227     (void)nnumber;
228     (void)dummy;
229
230     ckb_info("RESTART called with %s\n", content);
231     nprintf(kb, -1, 0, "RESTART called with %s\n", content);
232     restart();
233 }
```

Here is the call graph for this function:



13.40.2.5 void nprintf (usbdevice * *kb*, int *nodenumber*, usbmode * *mode*, const char * *format*, ...)

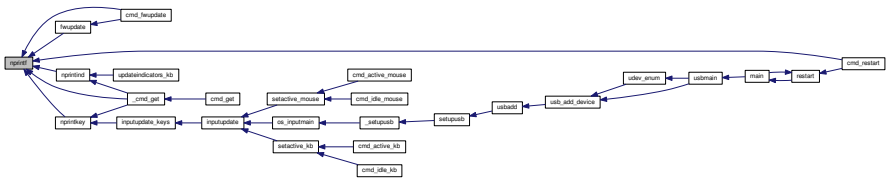
Definition at line 8 of file notify.c.

References INDEX_OF, usbprofile::mode, usbdevice::outfifo, OUTFIFO_MAX, and usbdevice::profile.

Referenced by _cmd_get(), cmd_fwupdate(), cmd_restart(), fwupdate(), nprintind(), and nprintkey().

```
8                                                                                   {
9       if(!kb)
10          return;
11      usbprofile* profile = kb->profile;
12      va_list va_args;
13      int fifo;
14      if(nodenumber >= 0){
15          // If node number was given, print to that node (if open)
16          if((fifo = kb->outfifo[nodenumber] - 1) != -1){
17              va_start(va_args, format);
18              if(mode)
19                  dprintf(fifo, "mode %d ", INDEX_OF(mode, profile->mode) + 1);
20              vdprintf(fifo, format, va_args);
21          }
22          return;
23      }
24      // Otherwise, print to all nodes
25      for(int i = 0; i < OUTFIFO_MAX; i++){
26          if((fifo = kb->outfifo[i] - 1) != -1){
27              va_start(va_args, format);
28              if(mode)
29                  dprintf(fifo, "mode %d ", INDEX_OF(mode, profile->mode) + 1);
30              vdprintf(fifo, format, va_args);
31          }
32      }
33  }
```

Here is the caller graph for this function:



13.40.2.6 void nprintind (usbdevice * kb, int nnumber, int led, int on)

Definition at line 43 of file notify.c.

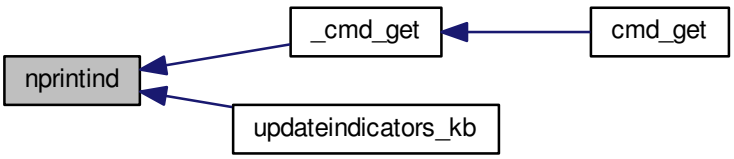
References I_CAPS, I_NUM, I_SCROLL, and nprintf().

Referenced by _cmd_get(), and updateindicators_kb().

```
43                                                          {
44     const char* name = 0;
45     switch(led){
46     case I_NUM:
47         name = "num";
48         break;
49     case I_CAPS:
50         name = "caps";
51         break;
52     case I_SCROLL:
53         name = "scroll";
54         break;
55     default:
56         return;
57     }
58     nprintf(kb, nnumber, 0, "i %c%s\n", on ? '+' : '-', name);
59 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.40.2.7 void nprintkey (usbdevice * kb, int nnumber, int keyindex, int down)

Definition at line 35 of file notify.c.

References keymap, key::name, and nprintf().

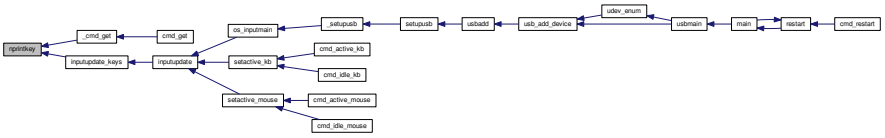
Referenced by _cmd_get(), and inputupdate_keys().

```
35                                                                  {
36     const key* map = keymap + keyindex;
37     if(map->name)
```

```
38         nprintf(kb, nnumber, 0, "key %c%s\n", down ? '+' : '-', map->name);
39     else
40         nprintf(kb, nnumber, 0, "key %c#%d\n", down ? '+' : '-', keyindex);
41 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.40.2.8 int restart ( )

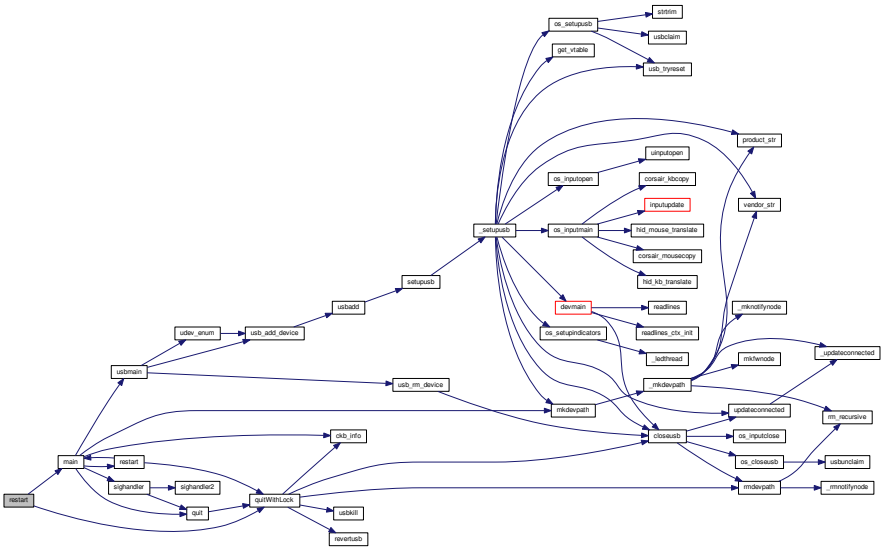
Definition at line 228 of file main.c.

References ckb_err, main(), main_ac, main_av, and quitWithLock().

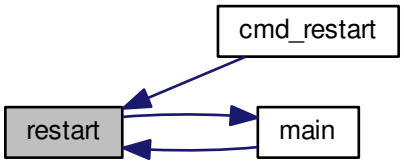
Referenced by cmd_restart(), and main().

```
228             {
229     ckb_err("restart called, running quit without mutex-lock.\n");
230     quitWithLock(0);
231     return main(main_ac, main_av);
232 }
```

Here is the call graph for this function:



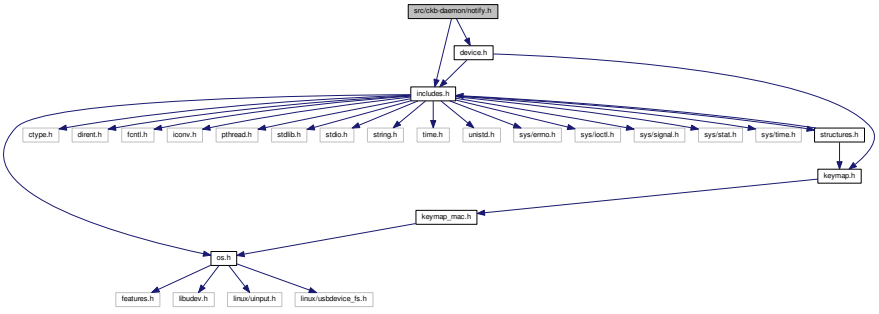
Here is the caller graph for this function:



13.41 src/ckb-daemon/notify.h File Reference

```
#include "includes.h"
#include "device.h"
```

Include dependency graph for notify.h:



This graph shows which files directly or indirectly include this file:

Functions

- void nprintf (usbdevice *kb, int nodenumber, usbmode *mode, const char *format,...)
- void nprintkey (usbdevice *kb, int nnumber, int keyindex, int down)
- void nprintind (usbdevice *kb, int nnumber, int led, int on)
- void cmd_notify (usbdevice *kb, usbmode *mode, int nnumber, int keyindex, const char *toggle)
- void cmd_get (usbdevice *kb, usbmode *mode, int nnumber, int dummy, const char *setting)
- void cmd_restart (usbdevice *kb, usbmode *mode, int nnumber, int dummy, const char *content)

13.41.1 Function Documentation

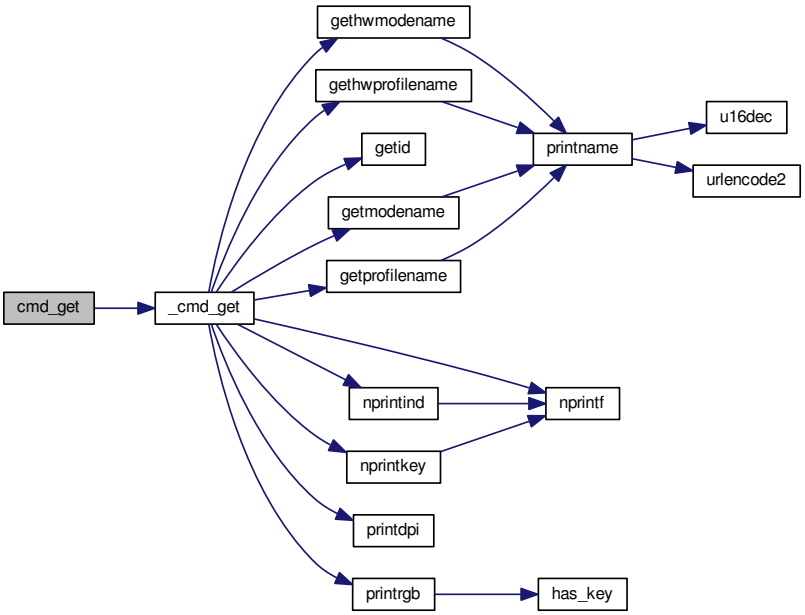
13.41.1.1 void cmd_get (usbdevice * *kb*, usbmode * *mode*, int *nnumber*, int *dummy*, const char * *setting*)

Definition at line 215 of file notify.c.

References _cmd_get(), and imutex.

```
215                                                                                                 {
216     (void)dummy;
217
218     pthread_mutex_lock(imutex(kb));
219     _cmd_get(kb, mode, nnumber, setting);
220     pthread_mutex_unlock(imutex(kb));
221 }
```

Here is the call graph for this function:



13.41.1.2 void cmd_notify (usbdevice * kb, usbmode * mode, int nnumber, int keyindex, const char * toggle)

Definition at line 61 of file notify.c.

References CLEAR_KEYBIT, imutex, N_KEYS_INPUT, usbmode::notify, and SET_KEYBIT.

```
61                                                                                                  {
62     if(keyindex >= N_KEYS_INPUT)
63         return;
64     pthread_mutex_lock(imutex(kb));
65     if(!strcmp(toggle, "on") || *toggle == 0)
66         SET_KEYBIT(mode->notify[nnumber], keyindex);
67     else if(!strcmp(toggle, "off"))
68         CLEAR_KEYBIT(mode->notify[nnumber], keyindex);
69     pthread_mutex_unlock(imutex(kb));
70 }
```

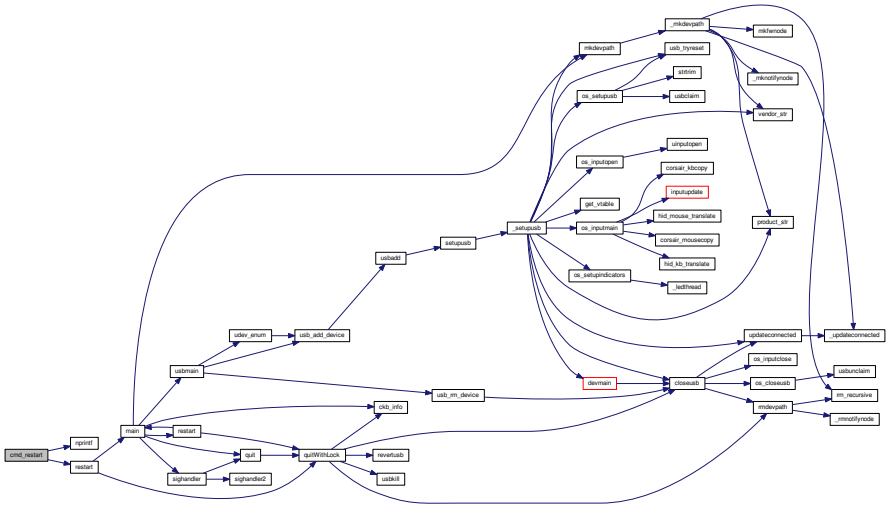
13.41.1.3 void cmd_restart (usbdevice * kb, usbmode * mode, int nnumber, int dummy, const char * content)

Definition at line 225 of file notify.c.

References ckb_info, nprintf(), and restart().

```
225                                                                                                 {
226     (void)mode;
227     (void)nnumber;
228     (void)dummy;
229
230     ckb_info("RESTART called with %s\n", content);
231     nprintf(kb, -1, 0, "RESTART called with %s\n", content);
232     restart();
233 }
```

Here is the call graph for this function:



13.41.1.4 void nprintf (usbdevice * kb, int nodenumber, usbmode * mode, const char * format, ...)

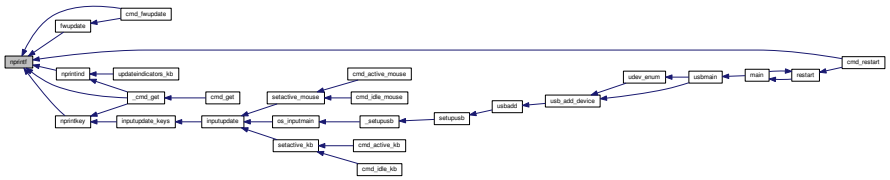
Definition at line 8 of file notify.c.

References INDEX_OF, usbprofile::mode, usbdevice::outfifo, OUTFIFO_MAX, and usbdevice::profile.

Referenced by _cmd_get(), cmd_fwupdate(), cmd_restart(), fwupdate(), nprintind(), and nprintkey().

```
8                                                                                                   {
9       if(!kb)
10          return;
11      usbprofile* profile = kb->profile;
12      va_list va_args;
13      int fifo;
14      if(nodenumber >= 0){
15          // If node number was given, print to that node (if open)
16          if((fifo = kb->outfifo[nodenumber] - 1) != -1){
17              va_start(va_args, format);
18              if(mode)
19                  dprintf(fifo, "mode %d ", INDEX_OF(mode, profile->mode) + 1);
20              vdprintf(fifo, format, va_args);
21          }
22          return;
23      }
24      // Otherwise, print to all nodes
25      for(int i = 0; i < OUTFIFO_MAX; i++){
26          if((fifo = kb->outfifo[i] - 1) != -1){
27              va_start(va_args, format);
28              if(mode)
29                  dprintf(fifo, "mode %d ", INDEX_OF(mode, profile->mode) + 1);
30              vdprintf(fifo, format, va_args);
31          }
32      }
33  }
```

Here is the caller graph for this function:



13.41.1.5 void nprintind (usbdevice * *kb,* int *nnumber,* int *led,* int *on*)

Definition at line 43 of file notify.c.

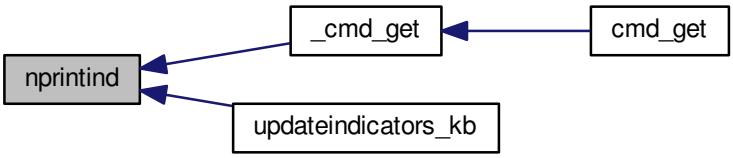
References I_CAPS, I_NUM, I_SCROLL, and nprintf().

Referenced by _cmd_get(), and updateindicators_kb().

```
43                                                                  {
44      const char* name = 0;
45      switch(led){
46      case I_NUM:
47          name = "num";
48          break;
49      case I_CAPS:
50          name = "caps";
51          break;
52      case I_SCROLL:
53          name = "scroll";
54          break;
55      default:
56          return;
57      }
58      nprintf(kb, nnumber, 0, "i %c%s\n", on ? '+' : '-', name);
59  }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.41.1.6 void nprintkey (usbdevice * *kb*, int *nnumber*, int *keyindex*, int *down*)

Definition at line 35 of file notify.c.

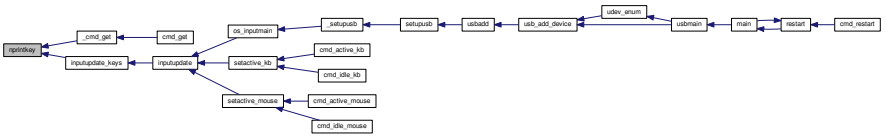
References keymap, key::name, and nprintf().

Referenced by _cmd_get(), and inputupdate_keys().

```
35                                                                  {
36     const key* map = keymap + keyindex;
37     if(map->name)
38         nprintf(kb, nnumber, 0, "key %c%s\n", down ? '+' : '-', map->name);
39     else
40         nprintf(kb, nnumber, 0, "key %c#%d\n", down ? '+' : '-', keyindex);
41 }
```

Here is the call graph for this function:

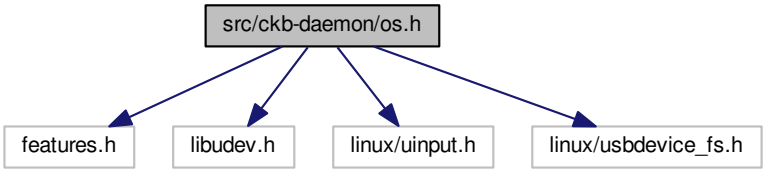
Here is the caller graph for this function:



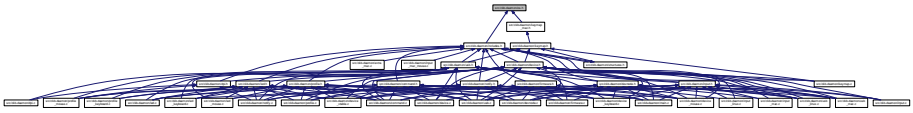
13.42 src/ckb-daemon/os.h File Reference

```
#include <features.h>
#include <libudev.h>
#include <linux/uinput.h>
#include <linux/usbdevice_fs.h>
```

Include dependency graph for os.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define _DEFAULT_SOURCE
- #define _GNU_SOURCE
- #define UINPUT_VERSION 2
- #define euid_guard_start
- #define euid_guard_stop

13.42.1 Macro Definition Documentation

13.42.1.1 #define _DEFAULT_SOURCE

Definition at line 22 of file os.h.

13.42.1.2 #define _GNU_SOURCE

Definition at line 26 of file os.h.

13.42.1.3 #define euid_guard_start

Definition at line 40 of file os.h.

Referenced by mkdevpath(), mknotifynode(), rmdevpath(), rmnotifynode(), and updateconnected().

13.42.1.4 #define euid_guard_stop

Definition at line 41 of file os.h.

Referenced by mkdevpath(), mknotifynode(), rmdevpath(), rmnotifynode(), and updateconnected().

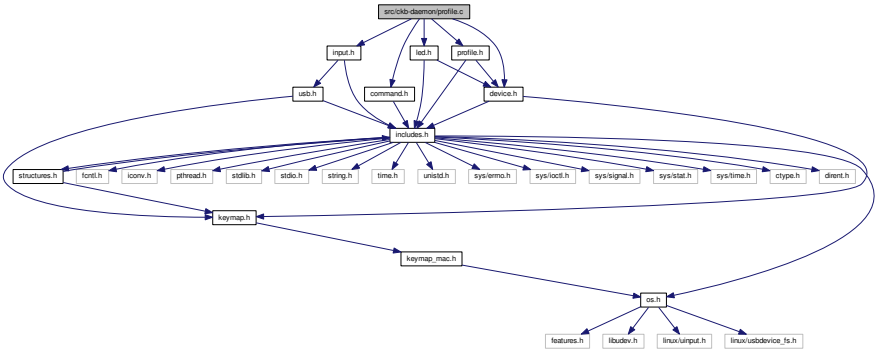
13.42.1.5 #define UINPUT_VERSION 2

Definition at line 35 of file os.h.

13.43 src/ckb-daemon/profile.c File Reference

```
#include "command.h"
#include "device.h"
#include "input.h"
#include "led.h"
#include "profile.h"
```

Include dependency graph for profile.c:



Functions

- void urldecode2 (char *dst, const char *src)
- void urlencode2 (char *dst, const char *src)
- int setid (usbid *id, const char *guid)
- char * getid (usbid *id)
- void u16enc (char *in, ushort *out, size_t *srclen, size_t *dstlen)
- void u16dec (ushort *in, char *out, size_t *srclen, size_t *dstlen)
- void cmd_name (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *name)
- void cmd_profilename (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *name)

- char * printname (ushort *name, int length)
- char * getmodename (usbmode *mode)
- char * getprofilename (usbprofile *profile)
- char * gethwmodename (hwprofile *profile, int index)
- char * gethwprofilename (hwprofile *profile)
- void cmd_id (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *id)
- void cmd_profileid (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *id)
- static void initmode (usbmode *mode)
- void allocprofile (usbdevice *kb)
- int loadprofile (usbdevice *kb)
- static void freemode (usbmode *mode)
- void cmd_erase (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *dummy3)
- static void _freeprofile (usbdevice *kb)
- void cmd_eraseprofile (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- void freeprofile (usbdevice *kb)
- void hwtonative (usbprofile *profile, hwprofile *hw, int modecount)
- void nativetohw (usbprofile *profile, hwprofile *hw, int modecount)

Variables

- static iconv_t utf8to16 = 0
- static iconv_t utf16to8 = 0

13.43.1 Function Documentation

13.43.1.1 static void _freeprofile (usbdevice * *kb*) [static]

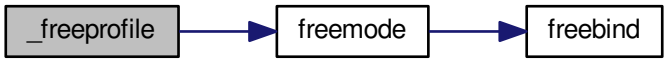
Definition at line 230 of file profile.c.

References freemode(), usbprofile::mode, MODE_COUNT, and usbdevice::profile.

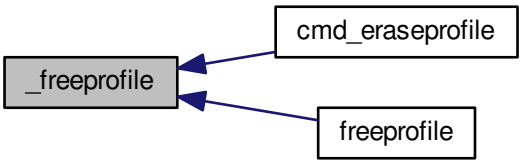
Referenced by cmd_eraseprofile(), and freeprofile().

```
230                                         {
231     usbprofile* profile = kb->profile;
232     if(!profile)
233         return;
234     // Clear all mode data
235     for(int i = 0; i < MODE_COUNT; i++)
236         freemode(profile->mode + i);
237     free(profile);
238     kb->profile = 0;
239 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.43.1.2 void allocprofile (usbdevice * *kb*)

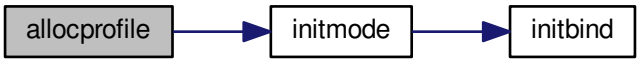
Definition at line 198 of file profile.c.

References usbprofile::currentmode, dpiset::forceupdate, lighting::forceupdate, initmode(), usbprofile::lastdpi, usbprofile::lastlight, usbprofile::mode, MODE_COUNT, and usbdevice::profile.

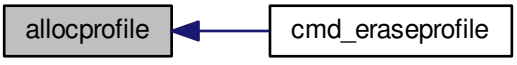
Referenced by cmd_eraseprofile().

```
198                                {
199     if(kb->profile)
200         return;
201     usbprofile* profile = kb->profile = calloc(1, sizeof(
      usbprofile));
202     for(int i = 0; i < MODE_COUNT; i++)
203         initmode(profile->mode + i);
204     profile->currentmode = profile->mode;
205     profile->lastlight.forceupdate = profile->lastdpi.
      forceupdate = 1;
206 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



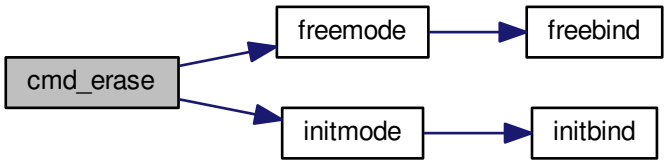
13.43.1.3 void cmd_erase (usbdevice * kb, usbmode * mode, int dummy1, int dummy2, const char * dummy3)

Definition at line 219 of file profile.c.

References freemode(), imutex, and initmode().

```
219                                                                                 {
220     (void)dummy1;
221     (void)dummy2;
222     (void)dummy3;
223
224     pthread_mutex_lock(imutex(kb));
225     freemode(mode);
226     initmode(mode);
227     pthread_mutex_unlock(imutex(kb));
228 }
```

Here is the call graph for this function:



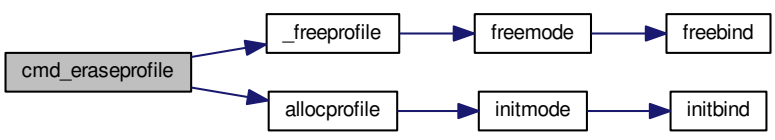
13.43.1.4 void cmd_eraseprofile (usbdevice * kb, usbmode * dummy1, int dummy2, int dummy3, const char * dummy4)

Definition at line 241 of file profile.c.

References _freeprofile(), allocprofile(), and imutex.

```
241                                                                                 {
242     (void)dummy1;
243     (void)dummy2;
244     (void)dummy3;
245     (void)dummy4;
246
247     pthread_mutex_lock(imutex(kb));
248     _freeprofile(kb);
249     allocprofile(kb);
250     pthread_mutex_unlock(imutex(kb));
251 }
```

Here is the call graph for this function:



13.43.1.5 void cmd_id (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *id*)

Definition at line 168 of file profile.c.

References usbmode::id, usbid::modified, and setid().

```
168                                                                                 {
169     (void)kb;
170     (void)dummy1;
171     (void)dummy2;
172
173     // ID is either a GUID or an 8-digit hex number
174     int newmodified;
175     if(!setid(&mode->id, id) && sscanf(id, "%08x", &newmodified) == 1)
176         memcpy(mode->id.modified, &newmodified, sizeof(newmodified));
177 }
```

Here is the call graph for this function:

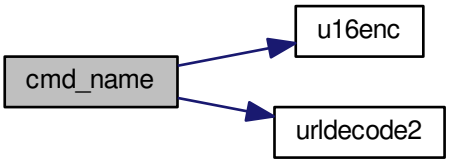
13.43.1.6 void cmd_name (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *name*)

Definition at line 117 of file profile.c.

References MD_NAME_LEN, usbmode::name, u16enc(), and urldecode2().

```
117                                                                                 {
118     (void)kb;
119     (void)dummy1;
120     (void)dummy2;
121
122     char decoded[strlen(name) + 1];
123     urldecode2(decoded, name);
124     size_t srclen = strlen(decoded), dstlen = MD_NAME_LEN;
125     u16enc(decoded, mode->name, &srclen, &dstlen);
126 }
```

Here is the call graph for this function:



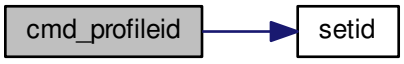
13.43.1.7 void cmd_profileid (usbdevice * kb, usbmode * mode, int dummy1, int dummy2, const char * id)

Definition at line 179 of file profile.c.

References usbprofile::id, usbid::modified, usbdevice::profile, and setid().

```
179                                                                                                 {
180     (void)mode;
181     (void)dummy1;
182     (void)dummy2;
183
184     usbprofile* profile = kb->profile;
185     int newmodified;
186     if(!setid(&profile->id, id) && sscanf(id, "%08x", &newmodified) == 1)
187         memcpy(profile->id.modified, &newmodified, sizeof(newmodified));
188
189 }
```

Here is the call graph for this function:



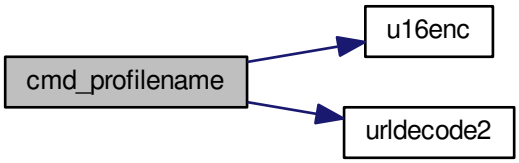
13.43.1.8 void cmd_profilename (usbdevice * kb, usbmode * dummy1, int dummy2, int dummy3, const char * name)

Definition at line 128 of file profile.c.

References usbprofile::name, PR_NAME_LEN, usbdevice::profile, u16enc(), and urldecode2().

```
128                                                                                                 {
129     (void)dummy1;
130     (void)dummy2;
131     (void)dummy3;
132
133     usbprofile* profile = kb->profile;
134     char decoded[strlen(name) + 1];
135     urldecode2(decoded, name);
136     size_t srclen = strlen(decoded), dstlen = PR_NAME_LEN;
137     u16enc(decoded, profile->name, &srclen, &dstlen);
138 }
```

Here is the call graph for this function:



13.43.1.9 static void freemode (usbmode * *mode*) [static]

Definition at line 214 of file profile.c.

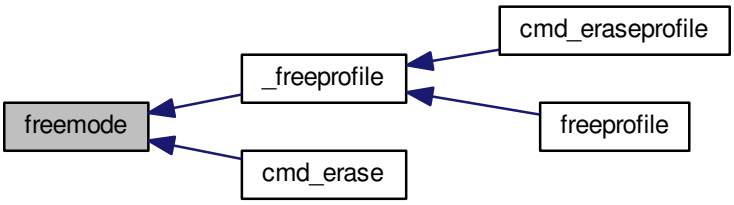
References usbmode::bind, and freebind().

Referenced by _freeprofile(), and cmd_erase().

```
214                                 {
215     freebind(&mode->bind);
216     memset(mode, 0, sizeof(*mode));
217 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



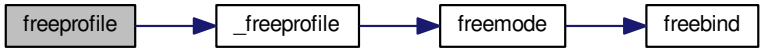
13.43.1.10 void freeprofile (usbdevice * *kb*)

Definition at line 253 of file profile.c.

References _freeprofile(), and usbdevice::hw.

```
253                                 {
254     _freeprofile(kb);
255     // Also free HW profile
256     free(kb->hw);
257     kb->hw = 0;
258 }
```

Here is the call graph for this function:



13.43.1.11 char* gethwmodename (hwprofile * *profile*, int *index*)

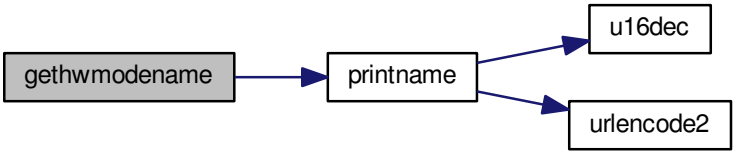
Definition at line 160 of file profile.c.

References MD_NAME_LEN, hwprofile::name, and printname().

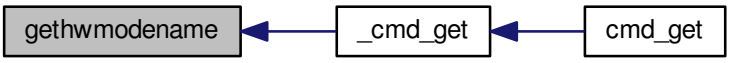
Referenced by _cmd_get().

```
160                                                 {
161     return printname(profile->name[index + 1], MD_NAME_LEN);
162 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.43.1.12 char* gethwprofilename (hwprofile * *profile*)

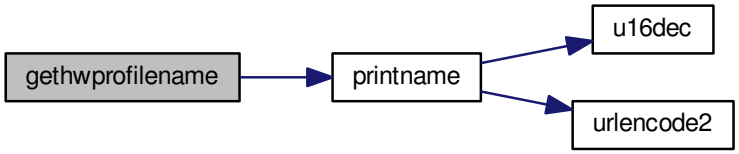
Definition at line 164 of file profile.c.

References MD_NAME_LEN, hwprofile::name, and printname().

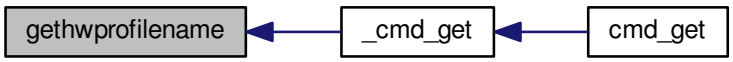
Referenced by _cmd_get().

```
164                                         {
165     return printname(profile->name[0], MD_NAME_LEN);
166 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.43.1.13 char* getid (usbid * *id*)

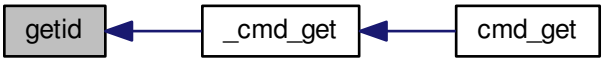
Definition at line 79 of file profile.c.

References usbid::guid.

Referenced by _cmd_get().

```
79                         {
80     int32_t data1;
81     int16_t data2, data3, data4a;
82     char data4b[6];
83     memcpy(&data1, id->guid + 0x0, 4);
84     memcpy(&data2, id->guid + 0x4, 2);
85     memcpy(&data3, id->guid + 0x6, 2);
86     memcpy(&data4a, id->guid + 0x8, 2);
87     memcpy(data4b, id->guid + 0xA, 6);
88     char* guid = malloc(39);
89     snprintf(guid, 39, "{%08X-%04hX-%04hX-%04hX-%02hhX%02hhX%02hhX%02hhX%02hhX%02hhX}",
90              data1, data2, data3, data4a, data4b[0], data4b[1], data4b[2], data4b[3], data4b[4], data4b[5])
     ;
91     return guid;
92 }
```

Here is the caller graph for this function:



13.43.1.14 char* getmodename (usbmode * *mode*)

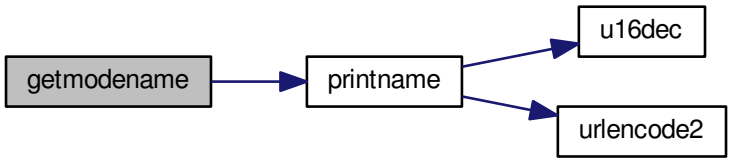
Definition at line 152 of file profile.c.

References MD_NAME_LEN, usbmode::name, and printname().

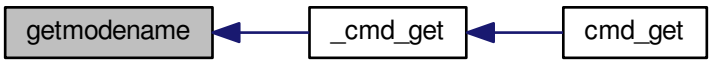
Referenced by _cmd_get().

```
152                                 {
153     return printname(mode->name, MD_NAME_LEN);
154 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.43.1.15 char* getprofilename (usbprofile * *profile*)

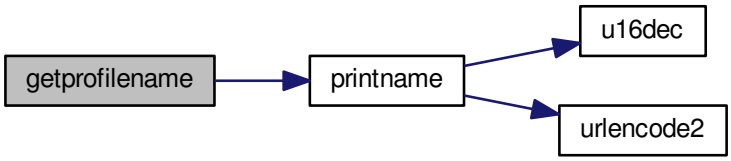
Definition at line 156 of file profile.c.

References usbprofile::name, PR_NAME_LEN, and printname().

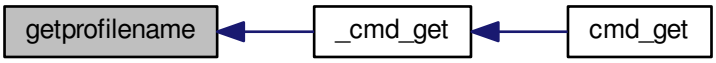
Referenced by _cmd_get().

```
156                                         {
157     return printname(profile->name, PR_NAME_LEN);
158 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.43.1.16 void hwtonative (usbprofile * *profile,* hwprofile * *hw,* int *modecount*)

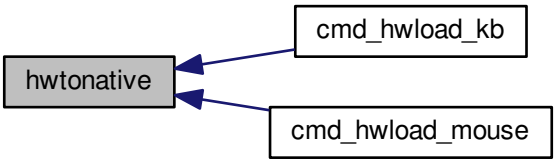
Definition at line 260 of file profile.c.

References usbmode::dpi, hwprofile::dpi, dpiset::forceupdate, lighting::forceupdate, usbmode::id, usbprofile::id, hwprofile::id, usbprofile::lastdpi, usbprofile::lastlight, usbmode::light, hwprofile::light, MD_NAME_LEN, usbprofile::mode, usbmode::name, usbprofile::name, hwprofile::name, and PR_NAME_LEN.

Referenced by cmd_hwload_kb(), and cmd_hwload_mouse().

```
260                                                                         {
261     // Copy the profile name and ID
262     memcpy(profile->name, hw->name[0], PR_NAME_LEN * 2);
263     memcpy(&profile->id, hw->id, sizeof(usbid));
264     // Copy the mode settings
265     for(int i = 0; i < modecount; i++){
266         usbmode* mode = profile->mode + i;
267         memcpy(mode->name, hw->name[i + 1], MD_NAME_LEN * 2);
268         memcpy(&mode->id, hw->id + i + 1, sizeof(usbid));
269         memcpy(&mode->light, hw->light + i, sizeof(lighting));
270         memcpy(&mode->dpi, hw->dpi + i, sizeof(dpiset));
271         // Set a force update on the light/DPI since they've been overwritten
272         mode->light.forceupdate = mode->dpi.forceupdate = 1;
273     }
274     profile->lastlight.forceupdate = profile->lastdpi.
    forceupdate = 1;
275 }
```

Here is the caller graph for this function:



13.43.1.17 static void initmode (usbmode * *mode*) [static]

Definition at line 191 of file profile.c.

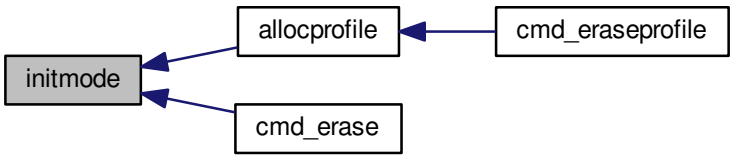
References usbmode::bind, usbmode::dpi, dpiset::forceupdate, lighting::forceupdate, initbind(), and usbmode::light.

Referenced by allocprofile(), and cmd_erase().

```
191                                     {
192     memset(mode, 0, sizeof(*mode));
193     mode->light.forceupdate = 1;
194     mode->dpi.forceupdate = 1;
195     initbind(&mode->bind);
196 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.43.1.18 int loadprofile (usbdevice * *kb*)

Definition at line 208 of file profile.c.

References hwloadprofile.

```
208                                {
209     if(hwloadprofile(kb, 1))
210         return -1;
211     return 0;
212 }
```

13.43.1.19 void nativetohw (usbprofile * *profile,* hwprofile * *hw,* int *modecount*)

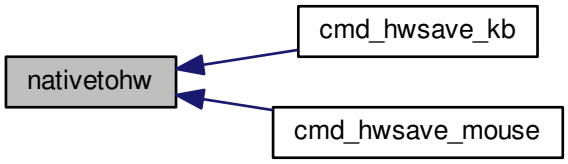
Definition at line 277 of file profile.c.

References usbmode::dpi, hwprofile::dpi, usbmode::id, usbprofile::id, hwprofile::id, usbmode::light, hwprofile::light, MD_NAME_LEN, usbprofile::mode, usbmode::name, usbprofile::name, hwprofile::name, and PR_NAME_LEN.

Referenced by cmd_hwsave_kb(), and cmd_hwsave_mouse().

```
277                                                                    {
278     // Copy name and ID
279     memcpy(hw->name[0], profile->name, PR_NAME_LEN * 2);
280     memcpy(hw->id, &profile->id, sizeof(usbid));
281     // Copy the mode settings
282     for(int i = 0; i < modecount; i++){
283         usbmode* mode = profile->mode + i;
284         memcpy(hw->name[i + 1], mode->name, MD_NAME_LEN * 2);
285         memcpy(hw->id + i + 1, &mode->id, sizeof(usbid));
286         memcpy(hw->light + i, &mode->light, sizeof(lighting));
287         memcpy(hw->dpi + i, &mode->dpi, sizeof(dpiset));
288     }
289 }
```

Here is the caller graph for this function:



13.43.1.20 char* printname (ushort * *name,* int *length*)

Definition at line 140 of file profile.c.

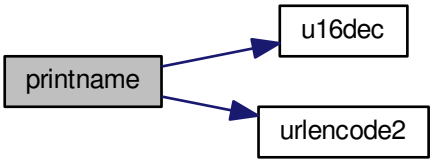
References u16dec(), and urlencode2().

Referenced by gethwmodename(), gethwprofilename(), getmodename(), and getprofilename().

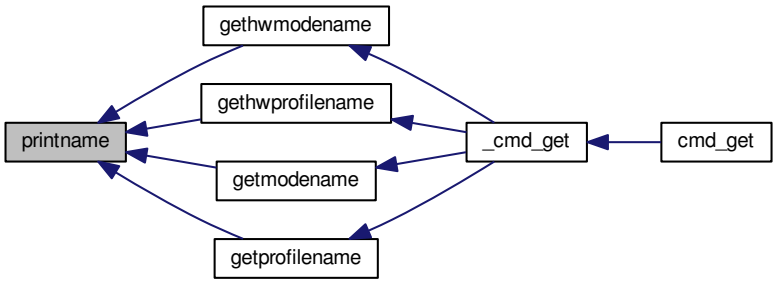
```
140                                        {
141     // Convert the name to UTF-8
142     char* buffer = calloc(1, length * 4 - 3);
143     size_t srclen = length, dstlen = length * 4 - 4;
144     u16dec(name, buffer, &srclen, &dstlen);
145     // URL-encode it
```

```
146     char* buffer2 = malloc(strlen(buffer) * 3 + 1);
147     urlencode2(buffer2, buffer);
148     free(buffer);
149     return buffer2;
150 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.43.1.21 int setid (usbid * *id*, const char * *guid*)

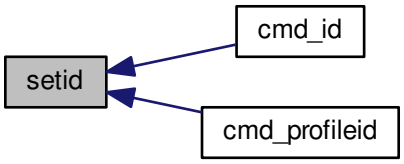
Definition at line 64 of file profile.c.

References usbid::guid.

Referenced by cmd_id(), and cmd_profileid().

```
64                                      {
65     int32_t data1;
66     int16_t data2, data3, data4a;
67     char data4b[6];
68     if(sscanf(guid, "{%08X-%04hX-%04hX-%04hX-%02hhX%02hhX%02hhX%02hhX%02hhX%02hhX}",
69               &data1, &data2, &data3, &data4a, data4b, data4b + 1, data4b + 2, data4b + 3, data4b + 4,
      data4b + 5) != 10)
70         return 0;
71     memcpy(id->guid + 0x0, &data1, 4);
72     memcpy(id->guid + 0x4, &data2, 2);
73     memcpy(id->guid + 0x6, &data3, 2);
74     memcpy(id->guid + 0x8, &data4a, 2);
75     memcpy(id->guid + 0xA, data4b, 6);
76     return 1;
77 }
```

Here is the caller graph for this function:



13.43.1.22 void u16dec (ushort * *in*, char * *out*, size_t * *srclen*, size_t * *dstlen*)

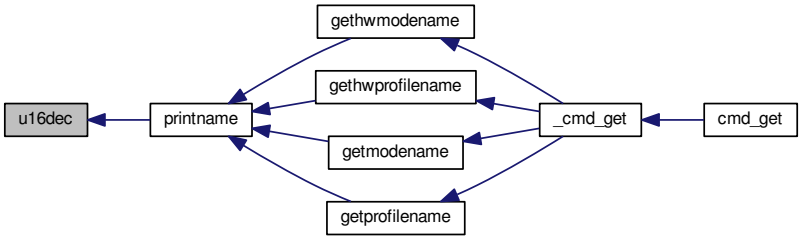
Definition at line 105 of file profile.c.

References utf16to8.

Referenced by printname().

```
105                                                                     {
106     if(!utf16to8)
107         utf16to8 = iconv_open("UTF-8", "UTF-16LE");
108     size_t srclen2 = 0, srclenmax = *srclen;
109     for(; srclen2 < srclenmax; srclen2++){
110         if(!in[srclen2])
111             break;
112     }
113     *srclen = srclen2 * 2;
114     iconv(utf16to8, (char**)&in, srclen, &out, dstlen);
115 }
```

Here is the caller graph for this function:



13.43.1.23 void u16enc (char * *in*, ushort * *out*, size_t * *srclen*, size_t * *dstlen*)

Definition at line 97 of file profile.c.

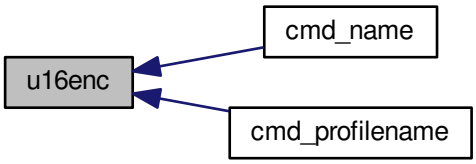
References utf8to16.

Referenced by cmd_name(), and cmd_profilename().

```
97                                                                      {
```

```
98      if(!utf8to16)
99          utf8to16 = iconv_open("UTF-16LE", "UTF-8");
100      memset(out, 0, *dstlen * 2);
101      *dstlen = *dstlen * 2 - 2;
102      iconv(utf8to16, &in, srclen, (char**)&out, dstlen);
103  }
```

Here is the caller graph for this function:



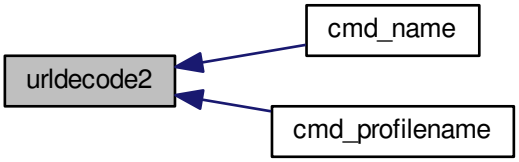
13.43.1.24 void urldecode2 (char * *dst*, const char * *src*)

Definition at line 8 of file profile.c.

Referenced by cmd_name(), and cmd_profilename().

```
8                                           {
9       char a, b;
10      char s;
11      while((s = *src)){
12          if((s == '%') &&
13                  ((a = src[1]) && (b = src[2])) &&
14                  (isxdigit(a) && isxdigit(b))){
15              if(a >= 'a')
16                  a -= 'a'-'A';
17              if(a >= 'A')
18                  a -= 'A' - 10;
19              else
20                  a -= '0';
21              if(b >= 'a')
22                  b -= 'a'-'A';
23              if(b >= 'A')
24                  b -= 'A' - 10;
25              else
26                  b -= '0';
27              *dst++ = 16 * a + b;
28              src += 3;
29          } else {
30              *dst++ = s;
31              src++;
32          }
33      }
34      *dst = '\0';
35  }
```

Here is the caller graph for this function:



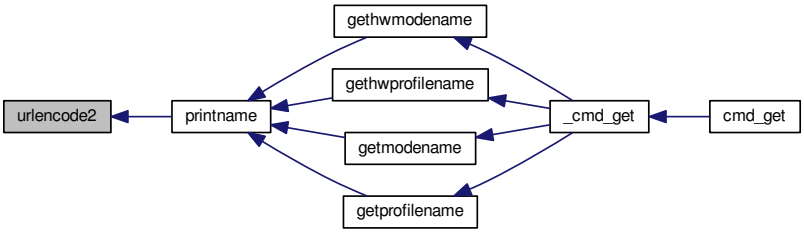
13.43.1.25 void urlencode2 (char * *dst*, const char * *src*)

Definition at line 37 of file profile.c.

Referenced by printname().

```
37                                              {
38     char s;
39     while((s = *src++)){
40         if(s <= ',' || s == '/' ||
41                 (s >= ':' && s <= '@') ||
42                 s == '[' || s == ']' ||
43                 s >= 0x7F){
44             char a = s >> 4, b = s & 0xF;
45             if(a >= 10)
46                 a += 'A' - 10;
47             else
48                 a += '0';
49             if(b >= 10)
50                 b += 'A' - 10;
51             else
52                 b += '0';
53             dst[0] = '%';
54             dst[1] = a;
55             dst[2] = b;
56             dst += 3;
57         } else
58             *dst++ = s;
59     }
60     *dst = '\0';
61 }
```

Here is the caller graph for this function:



13.43.2 Variable Documentation

13.43.2.1 iconv_t utf16to8 = 0 [static]

Definition at line 95 of file profile.c.

Referenced by u16dec().

13.43.2.2 iconv_t utf8to16 = 0 [static]

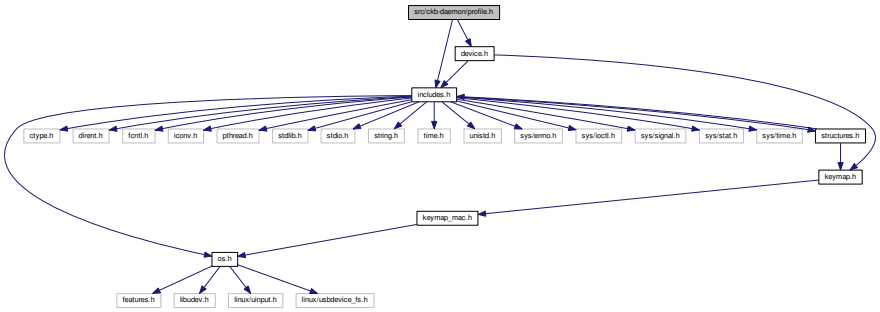
Definition at line 95 of file profile.c.

Referenced by u16enc().

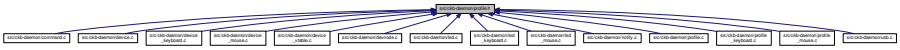
13.44 src/ckb-daemon/profile.h File Reference

```
#include "includes.h"
#include "device.h"
```

Include dependency graph for profile.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define hwloadprofile(kb, apply) (kb)->vtable->hwload(kb, 0, 0, apply, 0)

Functions

- void allocprofile (usbdevice *kb)
- int loadprofile (usbdevice *kb)
- void freeprofile (usbdevice *kb)
- void cmd_erase (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *dummy3)
- void cmd_eraseprofile (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- void cmd_name (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *name)
- void cmd_profilename (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *name)
- char * getmodename (usbmode *mode)
- char * getprofilename (usbprofile *profile)
- char * gethwmodename (hwprofile *profile, int index)

- char * gethwprofilename (hwprofile *profile)
- int setid (usbid *id, const char *guid)
- char * getid (usbid *id)
- void hwtonative (usbprofile *profile, hwprofile *hw, int modecount)
- void nativetohw (usbprofile *profile, hwprofile *hw, int modecount)
- void cmd_id (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *id)
- void cmd_profileid (usbdevice *kb, usbmode *mode, int dummy1, int dummy2, const char *id)
- int cmd_hwload_kb (usbdevice *kb, usbmode *dummy1, int dummy2, int apply, const char *dummy3)
- int cmd_hwload_mouse (usbdevice *kb, usbmode *dummy1, int dummy2, int apply, const char *dummy3)
- int cmd_hwsave_kb (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)
- int cmd_hwsave_mouse (usbdevice *kb, usbmode *dummy1, int dummy2, int dummy3, const char *dummy4)

13.44.1 Macro Definition Documentation

13.44.1.1 #define hwloadprofile(*kb, apply*) (kb)->vtable->hwload(kb, 0, 0, apply, 0)

Definition at line 52 of file profile.h.

Referenced by _start_dev(), and loadprofile().

13.44.2 Function Documentation

13.44.2.1 void allocprofile (usbdevice * *kb*)

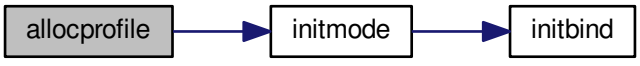
Definition at line 198 of file profile.c.

References usbprofile::currentmode, dpiset::forceupdate, lighting::forceupdate, initmode(), usbprofile::lastdpi, usbprofile::lastlight, usbprofile::mode, MODE_COUNT, and usbdevice::profile.

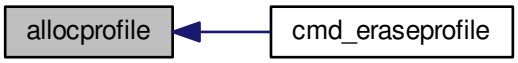
Referenced by cmd_eraseprofile().

```
198                                {
199     if(kb->profile)
200         return;
201     usbprofile* profile = kb->profile = calloc(1, sizeof(
      usbprofile));
202     for(int i = 0; i < MODE_COUNT; i++)
203         initmode(profile->mode + i);
204     profile->currentmode = profile->mode;
205     profile->lastlight.forceupdate = profile->lastdpi.
      forceupdate = 1;
206 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



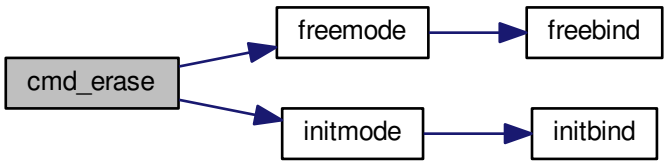
13.44.2.2 void cmd_erase (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *dummy3*)

Definition at line 219 of file profile.c.

References freemode(), imutex, and initmode().

```
219                                                                                 {
220     (void)dummy1;
221     (void)dummy2;
222     (void)dummy3;
223
224     pthread_mutex_lock(imutex(kb));
225     freemode(mode);
226     initmode(mode);
227     pthread_mutex_unlock(imutex(kb));
228 }
```

Here is the call graph for this function:



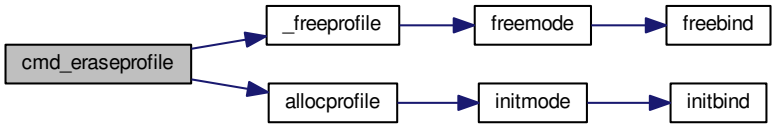
13.44.2.3 void cmd_eraseprofile (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 241 of file profile.c.

References _freeprofile(), allocprofile(), and imutex.

```
241                                                                                 {
242     (void)dummy1;
243     (void)dummy2;
244     (void)dummy3;
245     (void)dummy4;
246
247     pthread_mutex_lock(imutex(kb));
248     _freeprofile(kb);
249     allocprofile(kb);
250     pthread_mutex_unlock(imutex(kb));
251 }
```

Here is the call graph for this function:



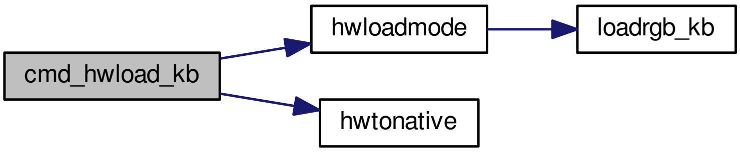
13.44.2.4 int cmd_hwload_kb (usbdevice * kb, usbmode * dummy1, int dummy2, int apply, const char * dummy3)

Definition at line 16 of file profile_keyboard.c.

References DELAY_LONG, usbdevice::hw, hwloadmode(), HWMODE_K70, HWMODE_K95, hwtonative(), hwprofile::id, IS_K95, MSG_SIZE, hwprofile::name, PR_NAME_LEN, usbdevice::profile, and usbrecv.

```
16                                                                                                  {
17      (void)dummy1;
18      (void)dummy2;
19      (void)dummy3;
20
21      DELAY_LONG(kb);
22      hwprofile* hw = calloc(1, sizeof(hwprofile));
23      // Ask for profile and mode IDs
24      uchar data_pkt[2][MSG_SIZE] = {
25          { 0x0e, 0x15, 0x01, 0 },
26          { 0x0e, 0x16, 0x01, 0 }
27      };
28      uchar in_pkt[MSG_SIZE];
29      int modes = (IS_K95(kb) ? HWMODE_K95 : HWMODE_K70);
30      for(int i = 0; i <= modes; i++){
31          data_pkt[0][3] = i;
32          if(!usbrecv(kb, data_pkt[0], in_pkt)){
33              free(hw);
34              return -1;
35          }
36          memcpy(hw->id + i, in_pkt + 4, sizeof(usbid));
37      }
38      // Ask for profile name
39      if(!usbrecv(kb, data_pkt[1], in_pkt)){
40          free(hw);
41          return -1;
42      }
43      memcpy(hw->name[0], in_pkt + 4, PR_NAME_LEN * 2);
44      // Load modes
45      for(int i = 0; i < modes; i++){
46          if(hwloadmode(kb, hw, i)){
47              free(hw);
48              return -1;
49          }
50      }
51      // Make the profile active (if requested)
52      if(apply)
53          hwtonative(kb->profile, hw, modes);
54      // Free the existing profile (if any)
55      free(kb->hw);
56      kb->hw = hw;
57      DELAY_LONG(kb);
58      return 0;
59  }
```

Here is the call graph for this function:



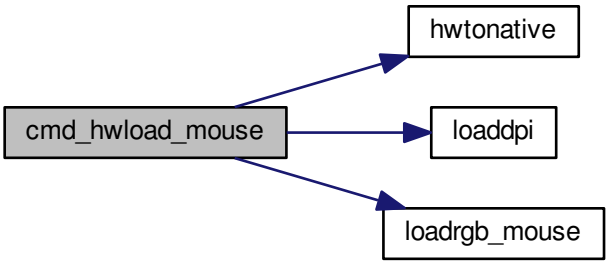
13.44.2.5 int cmd_hwload_mouse (usbdevice ∗ *kb*, usbmode ∗ *dummy1*, int *dummy2*, int *apply*, const char ∗ *dummy3*)

Definition at line 6 of file profile_mouse.c.

References DELAY_LONG, hwprofile::dpi, usbdevice::hw, hwtonative(), hwprofile::id, hwprofile::light, loaddpi(), loadrgb_mouse(), MSG_SIZE, hwprofile::name, PR_NAME_LEN, usbdevice::profile, and usbrecv.

```
6                                                                                                    {
7       (void)dummy1;
8       (void)dummy2;
9       (void)dummy3;
10
11      DELAY_LONG(kb);
12      hwprofile* hw = calloc(1, sizeof(hwprofile));
13      // Ask for profile and mode IDs
14      uchar data_pkt[2][MSG_SIZE] = {
15          { 0x0e, 0x15, 0x01, 0 },
16          { 0x0e, 0x16, 0x01, 0 }
17      };
18      uchar in_pkt[MSG_SIZE];
19      for(int i = 0; i <= 1; i++){
20          data_pkt[0][3] = i;
21          if(!usbrecv(kb, data_pkt[0], in_pkt)){
22              free(hw);
23              return -1;
24          }
25          memcpy(hw->id + i, in_pkt + 4, sizeof(usbid));
26      }
27      // Ask for profile and mode names
28      for(int i = 0; i <= 1; i++){
29          data_pkt[1][3] = i;
30          if(!usbrecv(kb, data_pkt[1],in_pkt)){
31              free(hw);
32              return -1;
33          }
34          memcpy(hw->name[i], in_pkt + 4, PR_NAME_LEN * 2);
35      }
36
37      // Load the RGB and DPI settings
38      if(loadrgb_mouse(kb, hw->light, 0)
39              || loaddpi(kb, hw->dpi, hw->light)){
40          free(hw);
41          return -1;
42      }
43
44      // Make the profile active (if requested)
45      if(apply)
46          hwtonative(kb->profile, hw, 1);
47      // Free the existing profile (if any)
48      free(kb->hw);
49      kb->hw = hw;
50      DELAY_LONG(kb);
51      return 0;
52 }
```

Here is the call graph for this function:



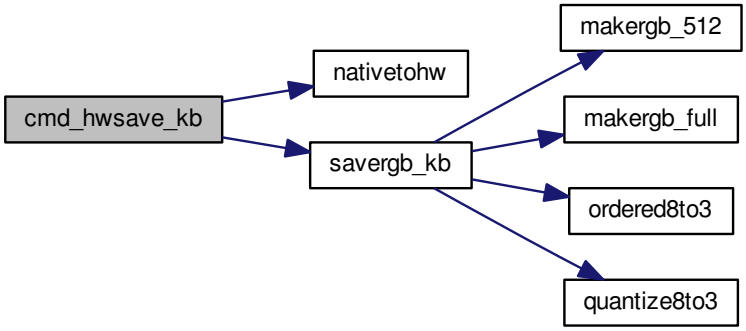
13.44.2.6 int cmd_hwsave_kb (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 61 of file profile_keyboard.c.

References DELAY_LONG, usbdevice::hw, HWMODE_K70, HWMODE_K95, hwprofile::id, IS_K95, hwprofile::light, MD_NAME_LEN, MSG_SIZE, hwprofile::name, nativetohw(), usbdevice::profile, savergb_kb(), and usbsend.

```
61                                                                                                  {
62      (void)dummy1;
63      (void)dummy2;
64      (void)dummy3;
65      (void)dummy4;
66
67      DELAY_LONG(kb);
68      hwprofile* hw = kb->hw;
69      if(!hw)
70          hw = kb->hw = calloc(1, sizeof(hwprofile));
71      int modes = (IS_K95(kb) ? HWMODE_K95 : HWMODE_K70);
72      nativetohw(kb->profile, hw, modes);
73      // Save the profile and mode names
74      uchar data_pkt[2][MSG_SIZE] = {
75          { 0x07, 0x16, 0x01, 0 },
76          { 0x07, 0x15, 0x01, 0 },
77      };
78      // Save the mode names
79      for(int i = 0; i <= modes; i++){
80          data_pkt[0][3] = i;
81          memcpy(data_pkt[0] + 4, hw->name[i], MD_NAME_LEN * 2);
82          if(!usbsend(kb, data_pkt[0], 1))
83              return -1;
84      }
85      // Save the IDs
86      for(int i = 0; i <= modes; i++){
87          data_pkt[1][3] = i;
88          memcpy(data_pkt[1] + 4, hw->id + i, sizeof(usbid));
89          if(!usbsend(kb, data_pkt[1], 1))
90              return -1;
91      }
92      // Save the RGB data
93      for(int i = 0; i < modes; i++){
94          if(savergb_kb(kb, hw->light + i, i))
95              return -1;
96      }
97      DELAY_LONG(kb);
98      return 0;
99  }
```

Here is the call graph for this function:



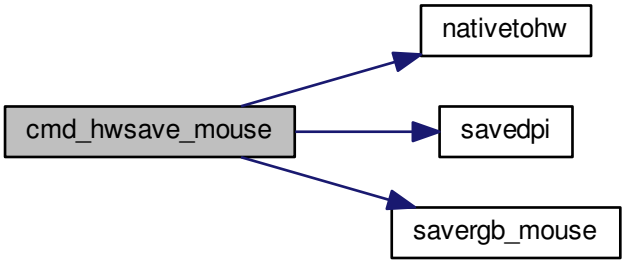
13.44.2.7 int cmd_hwsave_mouse (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 54 of file profile_mouse.c.

References DELAY_LONG, hwprofile::dpi, usbdevice::hw, hwprofile::id, hwprofile::light, MD_NAME_LEN, MSG_SIZE, hwprofile::name, nativetohw(), usbdevice::profile, savedpi(), savergb_mouse(), and usbsend.

```
54                                                                                  {
55     (void)dummy1;
56     (void)dummy2;
57     (void)dummy3;
58     (void)dummy4;
59
60     DELAY_LONG(kb);
61     hwprofile* hw = kb->hw;
62     if(!hw)
63         hw = kb->hw = calloc(1, sizeof(hwprofile));
64     nativetohw(kb->profile, hw, 1);
65     // Save the profile and mode names
66     uchar data_pkt[2][MSG_SIZE] = {
67         { 0x07, 0x16, 0x01, 0 },
68         { 0x07, 0x15, 0x01, 0 },
69     };
70     for(int i = 0; i <= 1; i++){
71         data_pkt[0][3] = i;
72         memcpy(data_pkt[0] + 4, hw->name[i], MD_NAME_LEN * 2);
73         if(!usbsend(kb, data_pkt[0], 1))
74             return -1;
75     }
76     // Save the IDs
77     for(int i = 0; i <= 1; i++){
78         data_pkt[1][3] = i;
79         memcpy(data_pkt[1] + 4, hw->id + i, sizeof(usbid));
80         if(!usbsend(kb, data_pkt[1], 1))
81             return -1;
82     }
83     // Save the RGB data for the non-DPI zones
84     if(savergb_mouse(kb, hw->light, 0))
85         return -1;
86     // Save the DPI data (also saves RGB for those states)
87     if(savedpi(kb, hw->dpi, hw->light))
88         return -1;
89     DELAY_LONG(kb);
90     return 0;
91 }
```

Here is the call graph for this function:



13.44.2.8 void cmd_id (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *id*)

Definition at line 168 of file profile.c.

References usbmode::id, usbid::modified, and setid().

```
168                                                                                 {
169     (void)kb;
170     (void)dummy1;
171     (void)dummy2;
172
173     // ID is either a GUID or an 8-digit hex number
174     int newmodified;
175     if(!setid(&mode->id, id) && sscanf(id, "%08x", &newmodified) == 1)
176         memcpy(mode->id.modified, &newmodified, sizeof(newmodified));
177 }
```

Here is the call graph for this function:

13.44.2.9 void cmd_name (usbdevice * *kb*, usbmode * *mode*, int *dummy1*, int *dummy2*, const char * *name*)

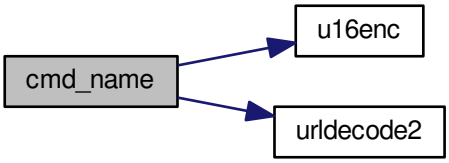
Definition at line 117 of file profile.c.

References MD_NAME_LEN, usbmode::name, u16enc(), and urldecode2().

```
117                                                                                 {
118     (void)kb;
119     (void)dummy1;
120     (void)dummy2;
121
122     char decoded[strlen(name) + 1];
123     urldecode2(decoded, name);
```

```
124     size_t srclen = strlen(decoded), dstlen = MD_NAME_LEN;
125     u16enc(decoded, mode->name, &srclen, &dstlen);
126 }
```

Here is the call graph for this function:



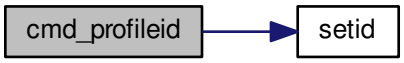
13.44.2.10 void cmd_profileid (usbdevice * kb, usbmode * mode, int dummy1, int dummy2, const char * id)

Definition at line 179 of file profile.c.

References usbprofile::id, usbid::modified, usbdevice::profile, and setid().

```
179                                                                                 {
180     (void)mode;
181     (void)dummy1;
182     (void)dummy2;
183
184     usbprofile* profile = kb->profile;
185     int newmodified;
186     if(!setid(&profile->id, id) && sscanf(id, "%08x", &newmodified) == 1)
187         memcpy(profile->id.modified, &newmodified, sizeof(newmodified));
188
189 }
```

Here is the call graph for this function:



13.44.2.11 void cmd_profilename (usbdevice * kb, usbmode * dummy1, int dummy2, int dummy3, const char * name)

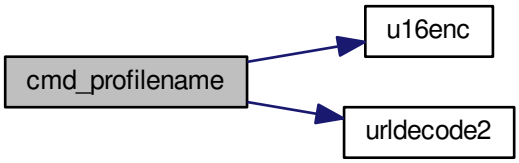
Definition at line 128 of file profile.c.

References usbprofile::name, PR_NAME_LEN, usbdevice::profile, u16enc(), and urldecode2().

```
128                                                                                 {
129     (void)dummy1;
130     (void)dummy2;
131     (void)dummy3;
132
```

```
133     usbprofile* profile = kb->profile;
134     char decoded[strlen(name) + 1];
135     urldecode2(decoded, name);
136     size_t srclen = strlen(decoded), dstlen = PR_NAME_LEN;
137     u16enc(decoded, profile->name, &srclen, &dstlen);
138 }
```

Here is the call graph for this function:



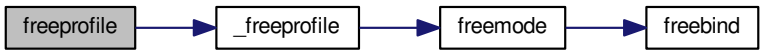
13.44.2.12 void freeprofile (usbdevice * *kb*)

Definition at line 253 of file profile.c.

References _freeprofile(), and usbdevice::hw.

```
253                                 {
254     _freeprofile(kb);
255     // Also free HW profile
256     free(kb->hw);
257     kb->hw = 0;
258 }
```

Here is the call graph for this function:



13.44.2.13 char* gethwmodename (hwprofile * *profile*, int *index*)

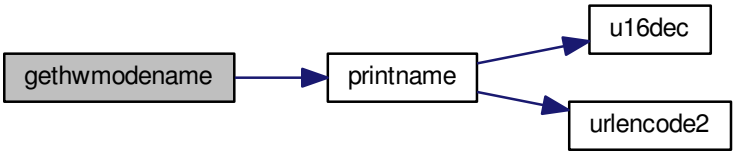
Definition at line 160 of file profile.c.

References MD_NAME_LEN, hwprofile::name, and printname().

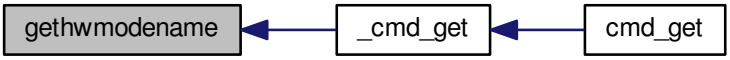
Referenced by _cmd_get().

```
160                                                 {
161     return printname(profile->name[index + 1], MD_NAME_LEN);
162 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.44.2.14 char* gethwprofilename (hwprofile * *profile*)

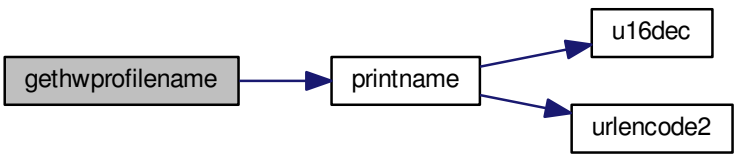
Definition at line 164 of file profile.c.

References MD_NAME_LEN, hwprofile::name, and printname().

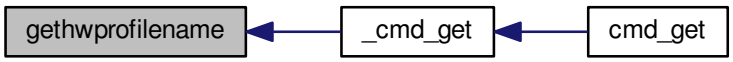
Referenced by _cmd_get().

```
164                                         {
165     return printname(profile->name[0], MD_NAME_LEN);
166 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.44.2.15 char* getid (usbid * *id*)

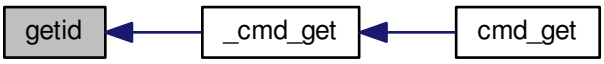
Definition at line 79 of file profile.c.

References usbid::guid.

Referenced by _cmd_get().

```
79                      {
80     int32_t data1;
81     int16_t data2, data3, data4a;
82     char data4b[6];
83     memcpy(&data1, id->guid + 0x0, 4);
84     memcpy(&data2, id->guid + 0x4, 2);
85     memcpy(&data3, id->guid + 0x6, 2);
86     memcpy(&data4a, id->guid + 0x8, 2);
87     memcpy(data4b, id->guid + 0xA, 6);
88     char* guid = malloc(39);
89     snprintf(guid, 39, "{%08X-%04hX-%04hX-%04hX-%02hhX%02hhX%02hhX%02hhX%02hhX%02hhX}",
90              data1, data2, data3, data4a, data4b[0], data4b[1], data4b[2], data4b[3], data4b[4], data4b[5])
    ;
91     return guid;
92 }
```

Here is the caller graph for this function:



13.44.2.16 char* getmodename (usbmode * *mode*)

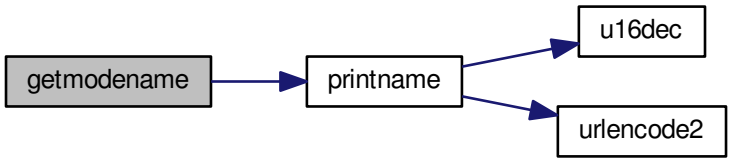
Definition at line 152 of file profile.c.

References MD_NAME_LEN, usbmode::name, and printname().

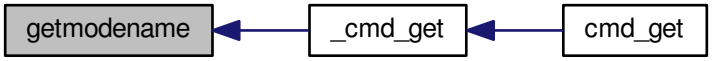
Referenced by _cmd_get().

```
152                                 {
153     return printname(mode->name, MD_NAME_LEN);
154 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.44.2.17 char* getprofilename (usbprofile * *profile*)

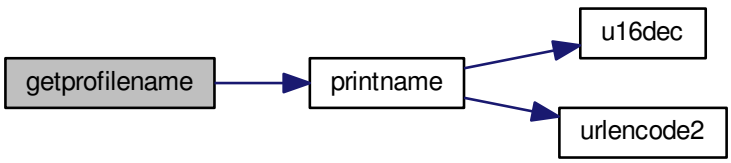
Definition at line 156 of file profile.c.

References usbprofile::name, PR_NAME_LEN, and printname().

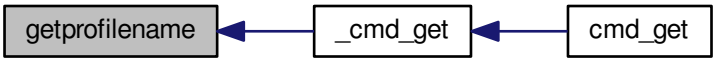
Referenced by _cmd_get().

```
156                                          {
157     return printname(profile->name, PR_NAME_LEN);
158 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.44.2.18 void hwtonative (usbprofile * *profile,* hwprofile * *hw,* int *modecount*)

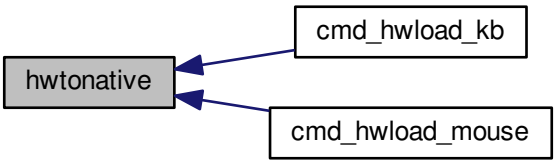
Definition at line 260 of file profile.c.

References usbmode::dpi, hwprofile::dpi, dpiset::forceupdate, lighting::forceupdate, usbmode::id, usbprofile::id, hwprofile::id, usbprofile::lastdpi, usbprofile::lastlight, usbmode::light, hwprofile::light, MD_NAME_LEN, usbprofile::mode, usbmode::name, usbprofile::name, hwprofile::name, and PR_NAME_LEN.

Referenced by cmd_hwload_kb(), and cmd_hwload_mouse().

```
260                                                                    {
261     // Copy the profile name and ID
262     memcpy(profile->name, hw->name[0], PR_NAME_LEN * 2);
263     memcpy(&profile->id, hw->id, sizeof(usbid));
264     // Copy the mode settings
265     for(int i = 0; i < modecount; i++){
266         usbmode* mode = profile->mode + i;
267         memcpy(mode->name, hw->name[i + 1], MD_NAME_LEN * 2);
268         memcpy(&mode->id, hw->id + i + 1, sizeof(usbid));
269         memcpy(&mode->light, hw->light + i, sizeof(lighting));
270         memcpy(&mode->dpi, hw->dpi + i, sizeof(dpiset));
271         // Set a force update on the light/DPI since they've been overwritten
272         mode->light.forceupdate = mode->dpi.forceupdate = 1;
273     }
274     profile->lastlight.forceupdate = profile->lastdpi.
      forceupdate = 1;
275 }
```

Here is the caller graph for this function:



13.44.2.19 int loadprofile (usbdevice * *kb*)

Definition at line 208 of file profile.c.

References hwloadprofile.

```
208                                 {
209     if(hwloadprofile(kb, 1))
210         return -1;
211     return 0;
212 }
```

13.44.2.20 void nativetohw (usbprofile * *profile,* hwprofile * *hw,* int *modecount*)

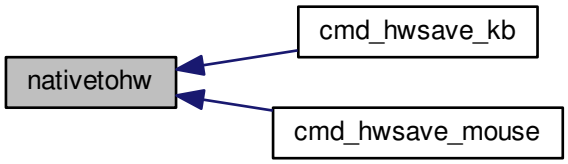
Definition at line 277 of file profile.c.

References usbmode::dpi, hwprofile::dpi, usbmode::id, usbprofile::id, hwprofile::id, usbmode::light, hwprofile::light, MD_NAME_LEN, usbprofile::mode, usbmode::name, usbprofile::name, hwprofile::name, and PR_NAME_LEN.

Referenced by cmd_hwsave_kb(), and cmd_hwsave_mouse().

```
277                                                                 {
278     // Copy name and ID
279     memcpy(hw->name[0], profile->name, PR_NAME_LEN * 2);
280     memcpy(hw->id, &profile->id, sizeof(usbid));
281     // Copy the mode settings
282     for(int i = 0; i < modecount; i++){
283         usbmode* mode = profile->mode + i;
284         memcpy(hw->name[i + 1], mode->name, MD_NAME_LEN * 2);
285         memcpy(hw->id + i + 1, &mode->id, sizeof(usbid));
286         memcpy(hw->light + i, &mode->light, sizeof(lighting));
287         memcpy(hw->dpi + i, &mode->dpi, sizeof(dpiset));
288     }
289 }
```

Here is the caller graph for this function:



13.44.2.21 int setid (usbid * *id,* const char * *guid*)

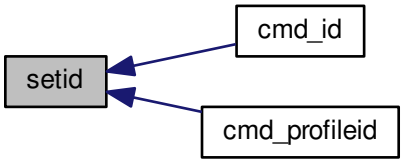
Definition at line 64 of file profile.c.

References usbid::guid.

Referenced by cmd_id(), and cmd_profileid().

```
64                                      {
65     int32_t data1;
66     int16_t data2, data3, data4a;
67     char data4b[6];
68     if(sscanf(guid, "{%08X-%04hX-%04hX-%04hX-%02hhX%02hhX%02hhX%02hhX%02hhX%02hhX}",
69               &data1, &data2, &data3, &data4a, data4b, data4b + 1, data4b + 2, data4b + 3, data4b + 4,
    data4b + 5) != 10)
70         return 0;
71     memcpy(id->guid + 0x0, &data1, 4);
72     memcpy(id->guid + 0x4, &data2, 2);
73     memcpy(id->guid + 0x6, &data3, 2);
74     memcpy(id->guid + 0x8, &data4a, 2);
75     memcpy(id->guid + 0xA, data4b, 6);
76     return 1;
77 }
```

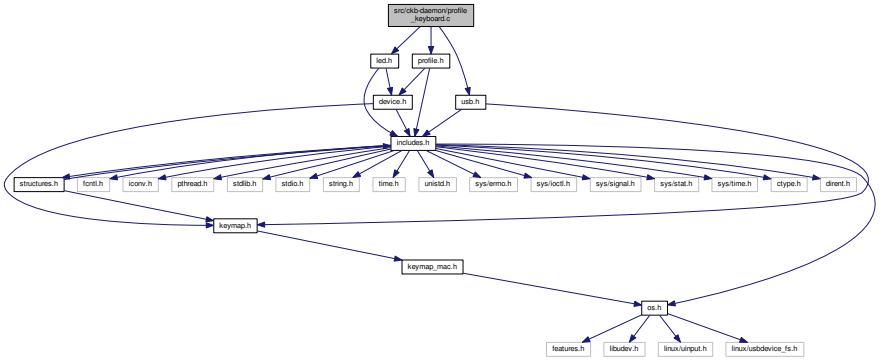
Here is the caller graph for this function:



13.45 src/ckb-daemon/profile_keyboard.c File Reference

```
#include "profile.h"
#include "usb.h"
#include "led.h"
```

Include dependency graph for profile_keyboard.c:



Functions

- static int hwloadmode (usbdevice ∗kb, hwprofile ∗hw, int mode)
- int cmd_hwload_kb (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int apply, const char ∗dummy3)
- int cmd_hwsave_kb (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)

13.45.1 Function Documentation

13.45.1.1 int cmd_hwload_kb (usbdevice ∗ *kb,* usbmode ∗ *dummy1,* int *dummy2,* int *apply,* const char ∗ *dummy3*)

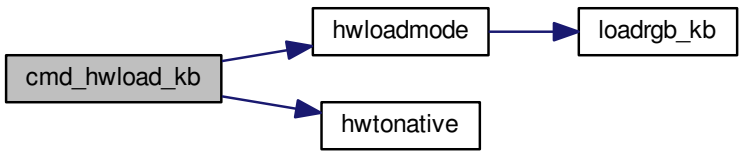
Definition at line 16 of file profile_keyboard.c.

References DELAY_LONG, usbdevice::hw, hwloadmode(), HWMODE_K70, HWMODE_K95, hwtonative(), hwprofile::id, IS_K95, MSG_SIZE, hwprofile::name, PR_NAME_LEN, usbdevice::profile, and usbrecv.

```
16                                                                      {
17     (void)dummy1;
18     (void)dummy2;
19     (void)dummy3;
```

```
20
21      DELAY_LONG(kb);
22      hwprofile* hw = calloc(1, sizeof(hwprofile));
23      // Ask for profile and mode IDs
24      uchar data_pkt[2][MSG_SIZE] = {
25          { 0x0e, 0x15, 0x01, 0 },
26          { 0x0e, 0x16, 0x01, 0 }
27      };
28      uchar in_pkt[MSG_SIZE];
29      int modes = (IS_K95(kb) ? HWMODE_K95 : HWMODE_K70);
30      for(int i = 0; i <= modes; i++){
31          data_pkt[0][3] = i;
32          if(!usbrecv(kb, data_pkt[0], in_pkt)){
33              free(hw);
34              return -1;
35          }
36          memcpy(hw->id + i, in_pkt + 4, sizeof(usbid));
37      }
38      // Ask for profile name
39      if(!usbrecv(kb, data_pkt[1], in_pkt)){
40          free(hw);
41          return -1;
42      }
43      memcpy(hw->name[0], in_pkt + 4, PR_NAME_LEN * 2);
44      // Load modes
45      for(int i = 0; i < modes; i++){
46          if(hwloadmode(kb, hw, i)){
47              free(hw);
48              return -1;
49          }
50      }
51      // Make the profile active (if requested)
52      if(apply)
53          hwtonative(kb->profile, hw, modes);
54      // Free the existing profile (if any)
55      free(kb->hw);
56      kb->hw = hw;
57      DELAY_LONG(kb);
58      return 0;
59  }
```

Here is the call graph for this function:



13.45.1.2 int cmd_hwsave_kb (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

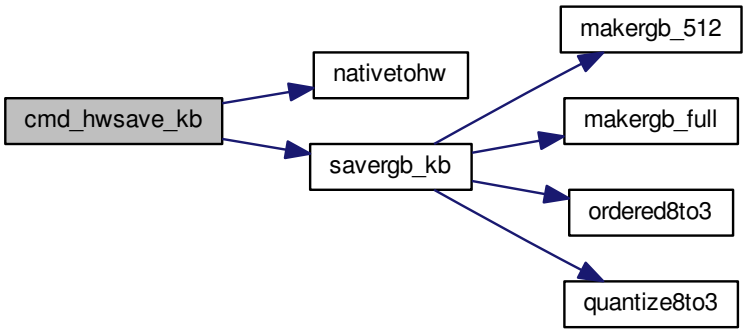
Definition at line 61 of file profile_keyboard.c.

References DELAY_LONG, usbdevice::hw, HWMODE_K70, HWMODE_K95, hwprofile::id, IS_K95, hwprofile::light, MD_NAME_LEN, MSG_SIZE, hwprofile::name, nativetohw(), usbdevice::profile, savergb_kb(), and usbsend.

```
61                                                                                  {
62      (void)dummy1;
63      (void)dummy2;
64      (void)dummy3;
65      (void)dummy4;
66
67      DELAY_LONG(kb);
68      hwprofile* hw = kb->hw;
69      if(!hw)
70          hw = kb->hw = calloc(1, sizeof(hwprofile));
71      int modes = (IS_K95(kb) ? HWMODE_K95 : HWMODE_K70);
```

```
72      nativetohw(kb->profile, hw, modes);
73      // Save the profile and mode names
74      uchar data_pkt[2][MSG_SIZE] = {
75          { 0x07, 0x16, 0x01, 0 },
76          { 0x07, 0x15, 0x01, 0 },
77      };
78      // Save the mode names
79      for(int i = 0; i <= modes; i++){
80          data_pkt[0][3] = i;
81          memcpy(data_pkt[0] + 4, hw->name[i], MD_NAME_LEN * 2);
82          if(!usbsend(kb, data_pkt[0], 1))
83              return -1;
84      }
85      // Save the IDs
86      for(int i = 0; i <= modes; i++){
87          data_pkt[1][3] = i;
88          memcpy(data_pkt[1] + 4, hw->id + i, sizeof(usbid));
89          if(!usbsend(kb, data_pkt[1], 1))
90              return -1;
91      }
92      // Save the RGB data
93      for(int i = 0; i < modes; i++){
94          if(savergb_kb(kb, hw->light + i, i))
95              return -1;
96      }
97      DELAY_LONG(kb);
98      return 0;
99  }
```

Here is the call graph for this function:



13.45.1.3 static int hwloadmode (usbdevice * *kb*, hwprofile * *hw*, int *mode*) [static]

Definition at line 5 of file profile_keyboard.c.

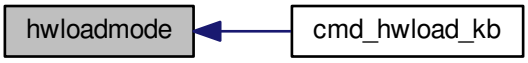
References hwprofile::light, loadrgb_kb(), MD_NAME_LEN, MSG_SIZE, hwprofile::name, and usbrecv.

Referenced by cmd_hwload_kb().

```
5                                                       {
6       // Ask for mode's name
7       uchar data_pkt[MSG_SIZE] = { 0x0e, 0x16, 0x01, mode + 1, 0 };
8       uchar in_pkt[MSG_SIZE];
9       if(!usbrecv(kb, data_pkt, in_pkt))
10          return -1;
11       memcpy(hw->name[mode + 1], in_pkt + 4, MD_NAME_LEN * 2);
12      // Load the RGB setting
13      return loadrgb_kb(kb, hw->light + mode, mode);
14  }
```

Here is the call graph for this function:

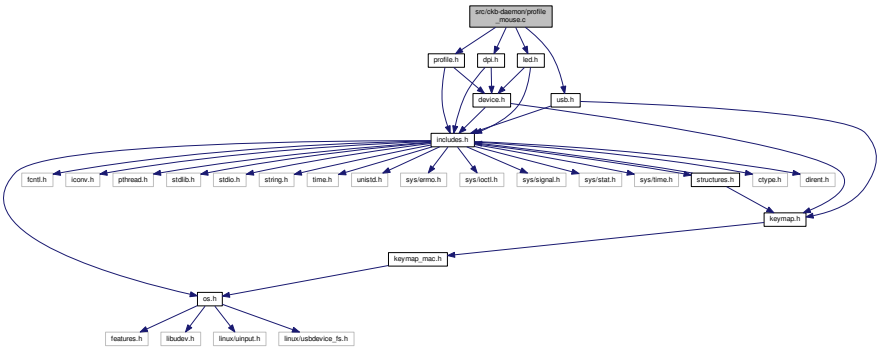
Here is the caller graph for this function:



13.46 src/ckb-daemon/profile_mouse.c File Reference

```
#include "dpi.h"
#include "profile.h"
#include "usb.h"
#include "led.h"
```

Include dependency graph for profile_mouse.c:



Functions

- int cmd_hwload_mouse (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int apply, const char ∗dummy3)
- int cmd_hwsave_mouse (usbdevice ∗kb, usbmode ∗dummy1, int dummy2, int dummy3, const char ∗dummy4)

13.46.1 Function Documentation

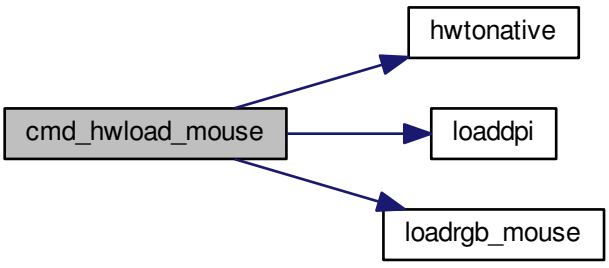
13.46.1.1 int cmd_hwload_mouse (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *apply*, const char * *dummy3*)

Definition at line 6 of file profile_mouse.c.

References DELAY_LONG, hwprofile::dpi, usbdevice::hw, hwtonative(), hwprofile::id, hwprofile::light, loaddpi(), loadrgb_mouse(), MSG_SIZE, hwprofile::name, PR_NAME_LEN, usbdevice::profile, and usbrecv.

```
6                                                                                                                   {
7       (void)dummy1;
8       (void)dummy2;
9       (void)dummy3;
10
11      DELAY_LONG(kb);
12      hwprofile* hw = calloc(1, sizeof(hwprofile));
13      // Ask for profile and mode IDs
14      uchar data_pkt[2][MSG_SIZE] = {
15          { 0x0e, 0x15, 0x01, 0 },
16          { 0x0e, 0x16, 0x01, 0 }
17      };
18      uchar in_pkt[MSG_SIZE];
19      for(int i = 0; i <= 1; i++){
20          data_pkt[0][3] = i;
21          if(!usbrecv(kb, data_pkt[0], in_pkt)){
22              free(hw);
23              return -1;
24          }
25          memcpy(hw->id + i, in_pkt + 4, sizeof(usbid));
26      }
27      // Ask for profile and mode names
28      for(int i = 0; i <= 1; i++){
29          data_pkt[1][3] = i;
30          if(!usbrecv(kb, data_pkt[1],in_pkt)){
31              free(hw);
32              return -1;
33          }
34          memcpy(hw->name[i], in_pkt + 4, PR_NAME_LEN * 2);
35      }
36
37      // Load the RGB and DPI settings
38      if(loadrgb_mouse(kb, hw->light, 0)
39              || loaddpi(kb, hw->dpi, hw->light)){
40          free(hw);
41          return -1;
42      }
43
44      // Make the profile active (if requested)
45      if(apply)
46          hwtonative(kb->profile, hw, 1);
47      // Free the existing profile (if any)
48      free(kb->hw);
49      kb->hw = hw;
50      DELAY_LONG(kb);
51      return 0;
52  }
```

Here is the call graph for this function:



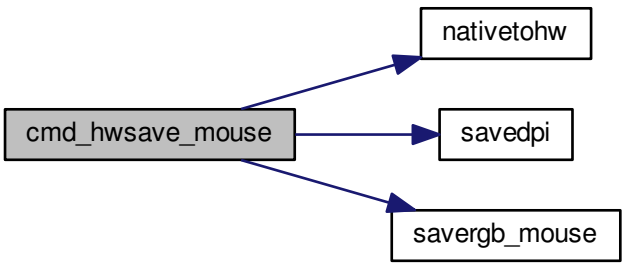
13.46.1.2 int cmd_hwsave_mouse (usbdevice * *kb*, usbmode * *dummy1*, int *dummy2*, int *dummy3*, const char * *dummy4*)

Definition at line 54 of file profile_mouse.c.

References DELAY_LONG, hwprofile::dpi, usbdevice::hw, hwprofile::id, hwprofile::light, MD_NAME_LEN, MSG_SIZE, hwprofile::name, nativetohw(), usbdevice::profile, savedpi(), savergb_mouse(), and usbsend.

```
54                                                                                                  {
55     (void)dummy1;
56     (void)dummy2;
57     (void)dummy3;
58     (void)dummy4;
59
60     DELAY_LONG(kb);
61     hwprofile* hw = kb->hw;
62     if(!hw)
63         hw = kb->hw = calloc(1, sizeof(hwprofile));
64     nativetohw(kb->profile, hw, 1);
65     // Save the profile and mode names
66     uchar data_pkt[2][MSG_SIZE] = {
67         { 0x07, 0x16, 0x01, 0 },
68         { 0x07, 0x15, 0x01, 0 },
69     };
70     for(int i = 0; i <= 1; i++){
71         data_pkt[0][3] = i;
72         memcpy(data_pkt[0] + 4, hw->name[i], MD_NAME_LEN * 2);
73         if(!usbsend(kb, data_pkt[0], 1))
74             return -1;
75     }
76     // Save the IDs
77     for(int i = 0; i <= 1; i++){
78         data_pkt[1][3] = i;
79         memcpy(data_pkt[1] + 4, hw->id + i, sizeof(usbid));
80         if(!usbsend(kb, data_pkt[1], 1))
81             return -1;
82     }
83     // Save the RGB data for the non-DPI zones
84     if(savergb_mouse(kb, hw->light, 0))
85         return -1;
86     // Save the DPI data (also saves RGB for those states)
87     if(savedpi(kb, hw->dpi, hw->light))
88         return -1;
89     DELAY_LONG(kb);
90     return 0;
91 }
```

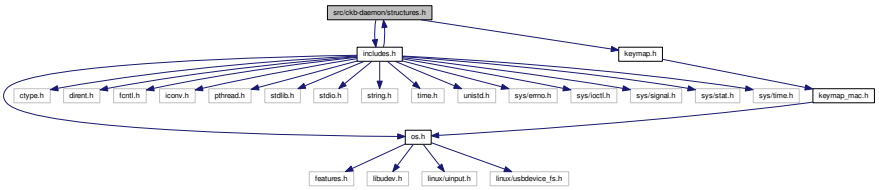
Here is the call graph for this function:



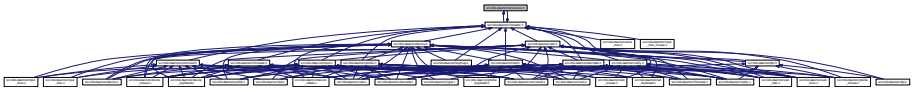
13.47 src/ckb-daemon/structures.h File Reference

```
#include "includes.h"
#include "keymap.h"
```

Include dependency graph for structures.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct usbid
- struct macroaction
- struct keymacro
- struct binding
- struct dpiset
- struct lighting
- struct usbmode
- struct usbprofile
- struct hwprofile
- struct usbinput
- struct usbdevice

Macros

- #define SET_KEYBIT(array, index) do { (array)[(index) / 8] |= 1 << ((index) % 8); } while(0)
- #define CLEAR_KEYBIT(array, index) do { (array)[(index) / 8] &= ~(1 << ((index) % 8)); } while(0)
- #define I_NUM 1
- #define I_CAPS 2
- #define I_SCROLL 4
- #define OUTFIFO_MAX 10
- #define MACRO_MAX 1024
- #define DPI_COUNT 6
- #define LIFT_MIN 1
- #define LIFT_MAX 5
- #define MD_NAME_LEN 16
- #define PR_NAME_LEN 16
- #define MODE_COUNT 6
- #define HWMODE_K70 1
- #define HWMODE_K95 3
- #define HWMODE_MAX 3
- #define FEAT_RGB 0x001
- #define FEAT_MONOCHROME 0x002
- #define FEAT_POLLRATE 0x004

- #define FEAT_ADJRATE 0x008
- #define FEAT_BIND 0x010
- #define FEAT_NOTIFY 0x020
- #define FEAT_FWVERSION 0x040
- #define FEAT_FWUPDATE 0x080
- #define FEAT_HWLOAD 0x100
- #define FEAT_ANSI 0x200
- #define FEAT_ISO 0x400
- #define FEAT_MOUSEACCEL 0x800
- #define FEAT_COMMON (FEAT_BIND | FEAT_NOTIFY | FEAT_FWVERSION | FEAT_MOUSEACCEL | FEAT_HWLOAD)
- #define FEAT_STD_RGB (FEAT_COMMON | FEAT_RGB | FEAT_POLLRATE | FEAT_FWUPDATE)
- #define FEAT_STD_NRGB (FEAT_COMMON)
- #define FEAT_LMASK (FEAT_ANSI | FEAT_ISO)
- #define HAS_FEATURES(kb, feat) (((kb)->features & (feat)) == (feat))
- #define HAS_ANY_FEATURE(kb, feat) (!!((kb)->features & (feat)))
- #define NEEDS_FW_UPDATE(kb) ((kb)->fwversion == 0 && HAS_FEATURES((kb), FEAT_FWUPDATE | FEAT_FWVERSION))
- #define SCROLL_ACCELERATED 0
- #define SCROLL_MIN 1
- #define SCROLL_MAX 10
- #define KB_NAME_LEN 40
- #define SERIAL_LEN 34
- #define MSG_SIZE 64
- #define IFACE_MAX 4

Variables

- const union devcmd vtable_keyboard

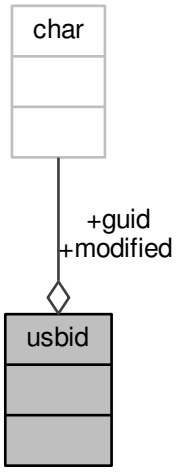
  RGB keyboard vtable holds functions for each device type.
- const union devcmd vtable_keyboard_nonrgb
- const union devcmd vtable_mouse

13.47.1 Data Structure Documentation

13.47.1.1 struct usbid

Definition at line 8 of file structures.h.

Collaboration diagram for usbid:



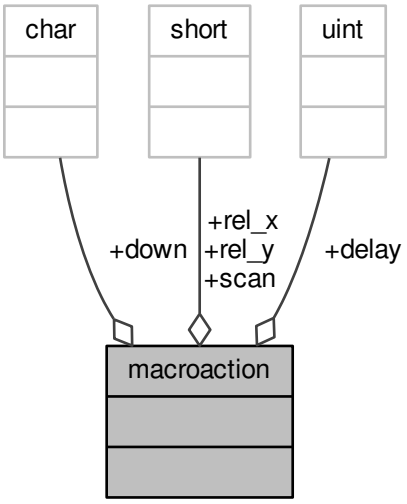
Data Fields

char	guid[16]	
char	modified[4]	

13.47.1.2 struct macroaction

Definition at line 27 of file structures.h.

Collaboration diagram for macroaction:



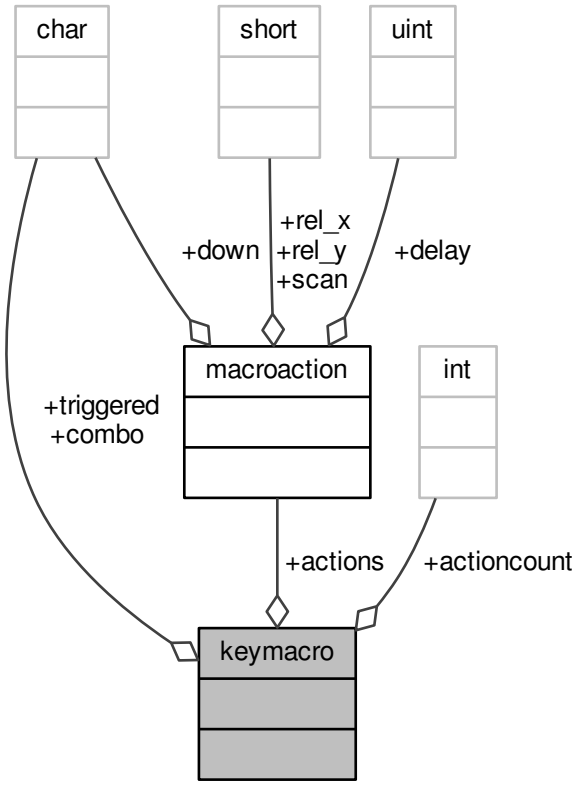
Data Fields

uint	delay	
char	down	
short	rel_x	
short	rel_y	
short	scan	

13.47.1.3 struct keymacro

Definition at line 35 of file structures.h.

Collaboration diagram for keymacro:



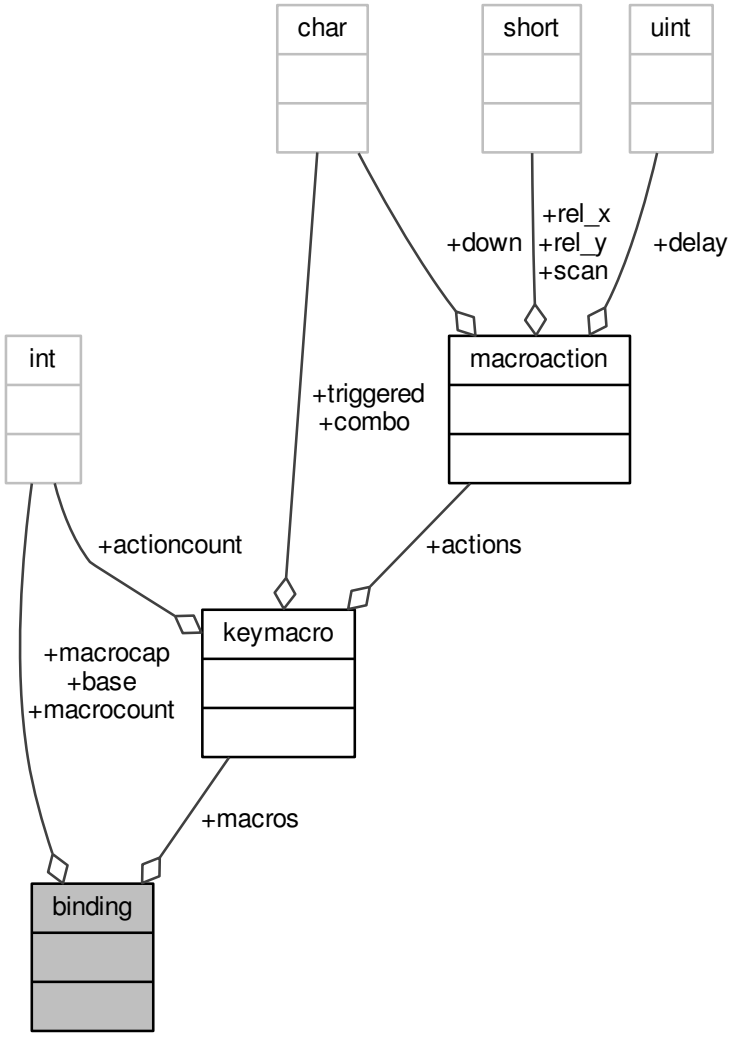
Data Fields

int	actioncount	
macroaction *	actions	
uchar	combo[(((((152+22+12)+25)+7)/8)]	
char	triggered	

13.47.1.4 struct binding

Definition at line 43 of file structures.h.

Collaboration diagram for binding:



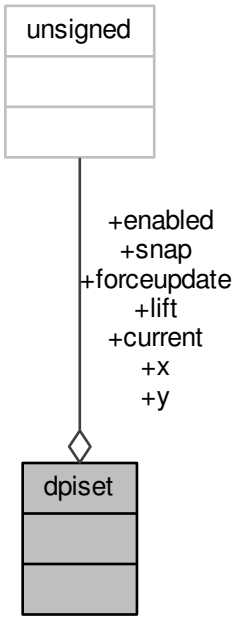
Data Fields

int	base[((152+22+12)+25)]	
int	macrocap	
int	macrocount	
keymacro *	macros	

13.47.1.5 struct dpiset

Definition at line 57 of file structures.h.

Collaboration diagram for dpiset:



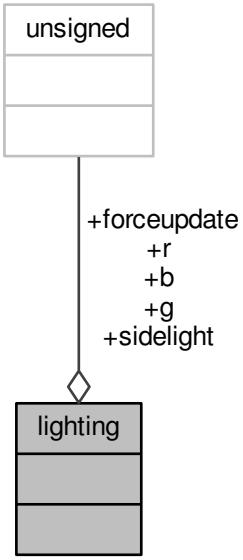
Data Fields

uchar	current	
uchar	enabled	
uchar	forceupdate	
uchar	lift	
uchar	snap	
ushort	x[6]	
ushort	y[6]	

13.47.1.6 struct lighting

Definition at line 73 of file structures.h.

Collaboration diagram for lighting:



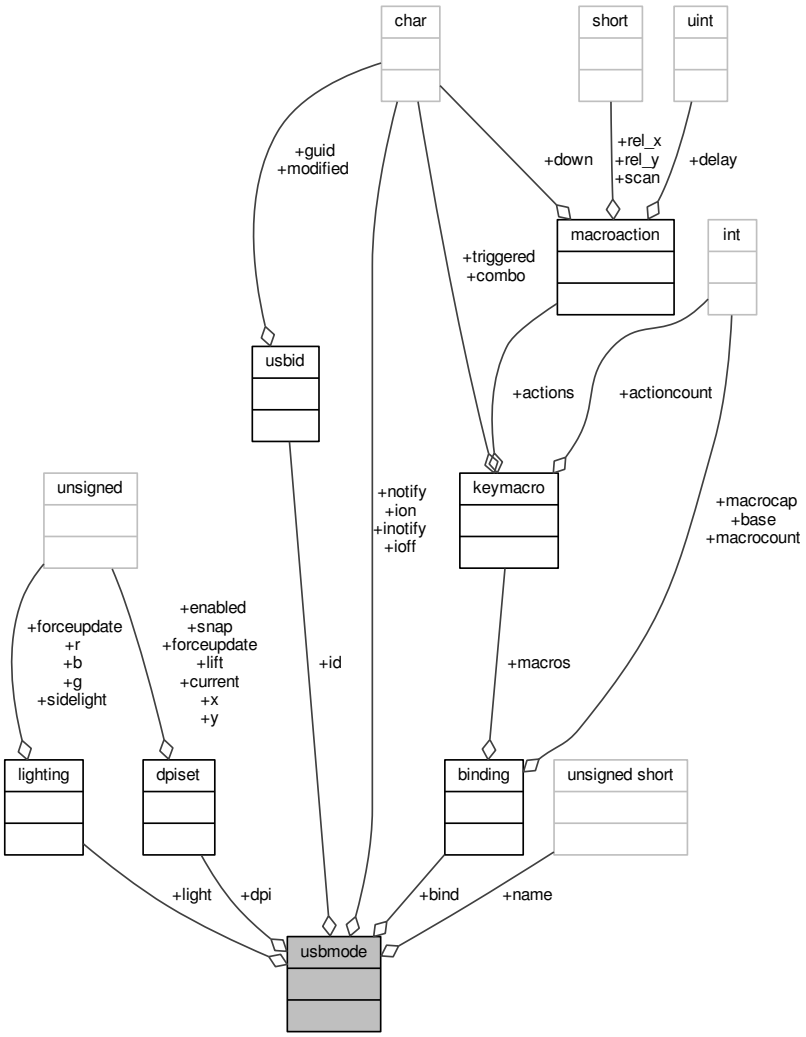
Data Fields

uchar	b[152+12]	
uchar	forceupdate	
uchar	g[152+12]	
uchar	r[152+12]	
uchar	sidelight	

13.47.1.7 struct usbmode

Definition at line 83 of file structures.h.

Collaboration diagram for usbmode:



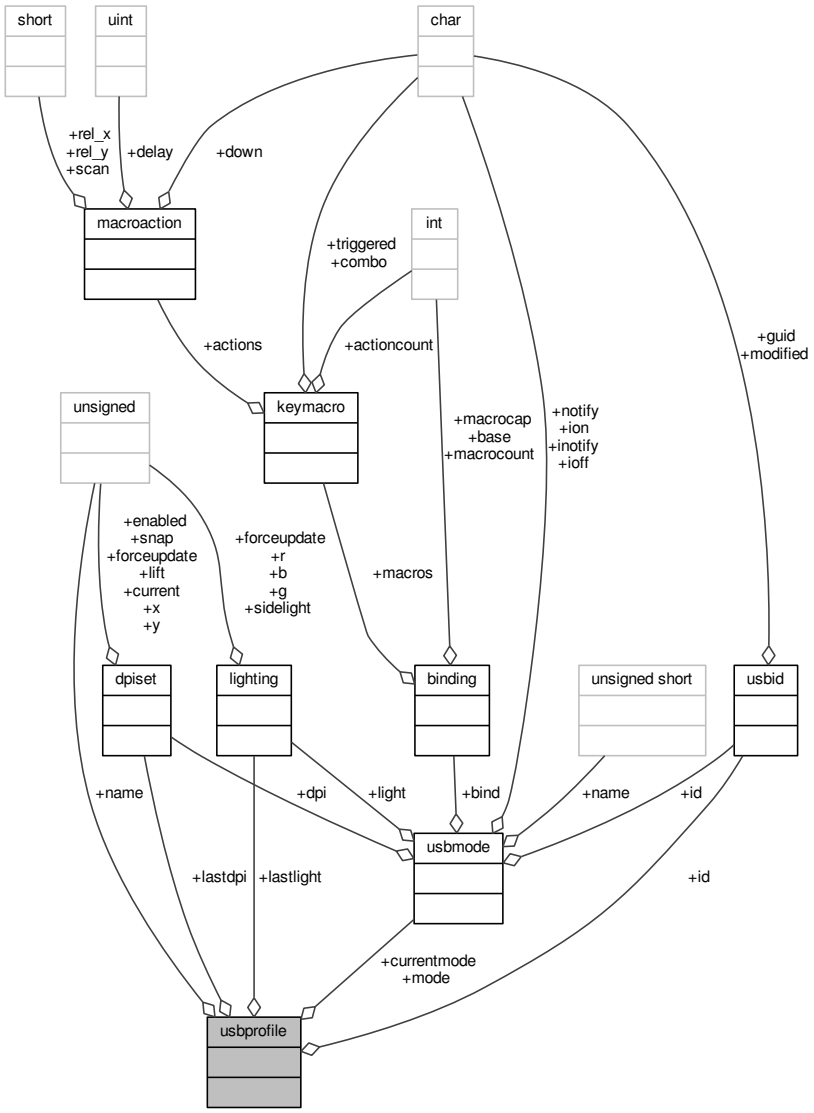
Data Fields

binding	bind	
dpiset	dpi	
usbid	id	
uchar	inotify[10]	
uchar	ioff	
uchar	ion	
lighting	light	
ushort	name[16]	
uchar	notify[10][((((152+22+12)+25)+7)/8)]	

13.47.1.8 struct usbprofile

Definition at line 101 of file structures.h.

Collaboration diagram for usbprofile:



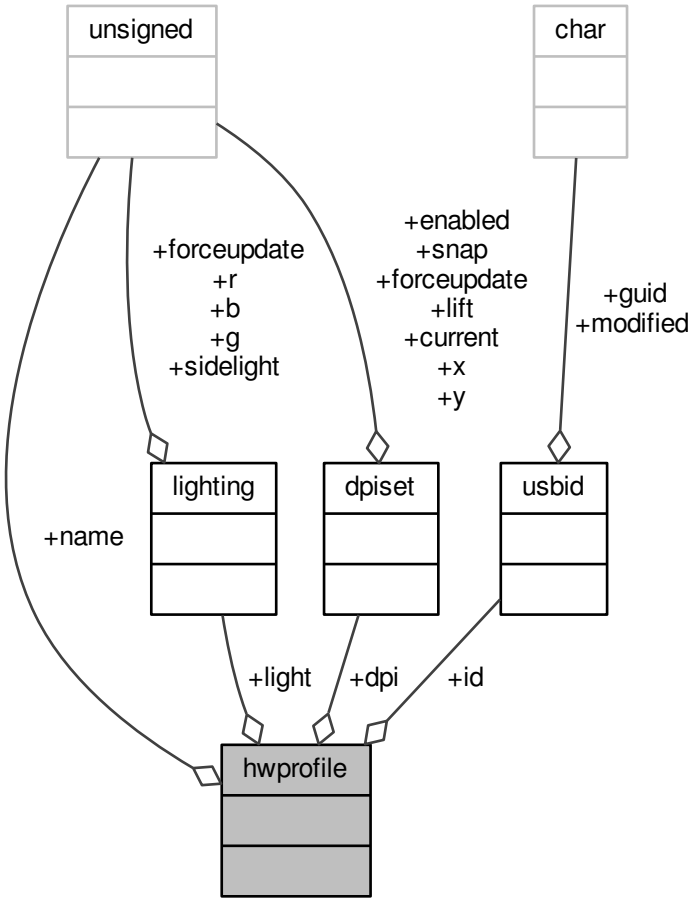
Data Fields

usbmode *	currentmode	
usbid	id	
dpiset	lastdpi	
lighting	lastlight	
usbmode	mode[6]	
ushort	name[16]	

13.47.1.9 struct hwprofile

Definition at line 118 of file structures.h.

Collaboration diagram for hwprofile:



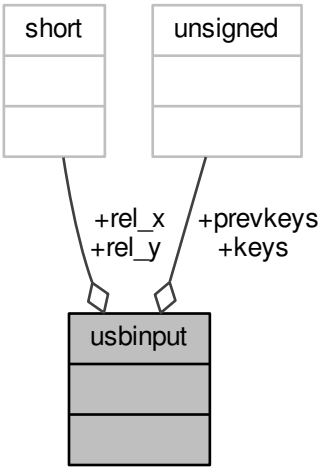
Data Fields

dpiset	dpi[3]	
usbid	id[3+1]	
lighting	light[3]	
ushort	name[3+1][16]	

13.47.1.10 struct usbinput

Definition at line 129 of file structures.h.

Collaboration diagram for usbinput:



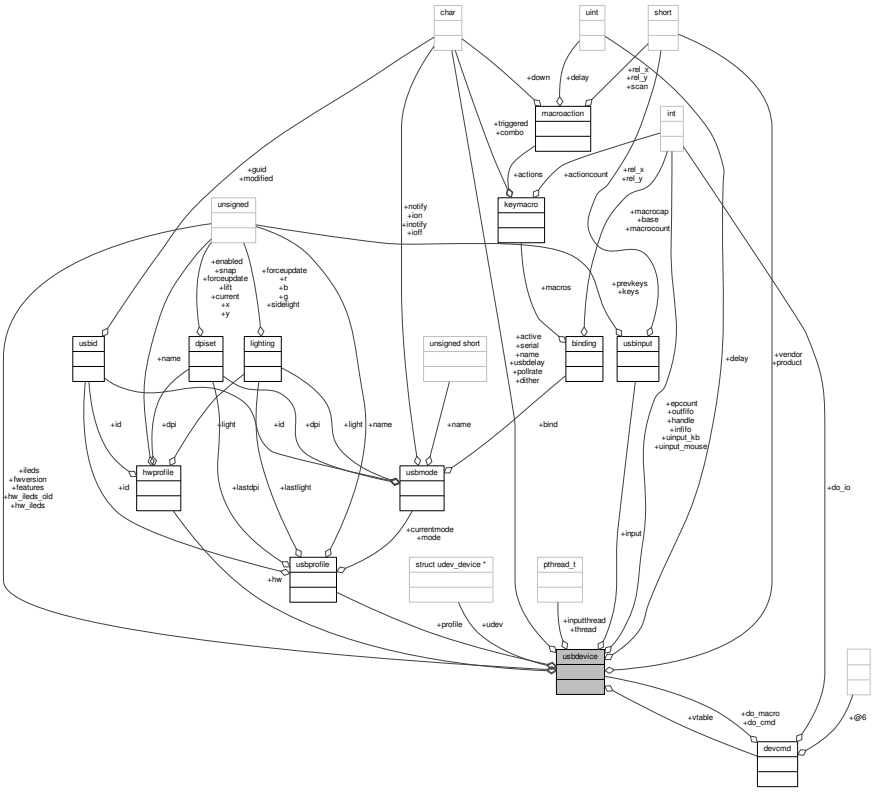
Data Fields

uchar	keys[((((152+22+12)+25)+7)/8)]
uchar	prevkeys[((((152+22+12)+25)+7)/8)]
short	rel_x
short	rel_y

13.47.1.11 struct usbdevice

Definition at line 178 of file structures.h.

Collaboration diagram for usbdevice:



Data Fields

char	active	
uint	delay	
char	dither	
int	epcount	
ushort	features	
ushort	fwversion	
int	handle	
hwprofile *	hw	
uchar	hw_ileds	
uchar	hw_ileds_old	
uchar	ileds	
int	infifo	
usbinput	input	
pthread_t	inputthread	
char	name[40+1]	
int	outfifo[10]	
char	pollrate	
short	product	

usbprofile *	profile	
char	serial[34]	
pthread_t	thread	
struct udev_device *	udev	
int	uinput_kb	
int	uinput_mouse	
char	usbdelay	
short	vendor	
const union devcmd *	vtable	

13.47.2 Macro Definition Documentation

13.47.2.1 #define CLEAR_KEYBIT(*array, index*) do { (array)[(index) / 8] &= ~(1 << ((index) % 8)); } while(0)

Definition at line 16 of file structures.h.

Referenced by cmd_notify(), corsair_mousecopy(), hid_kb_translate(), and hid_mouse_translate().

13.47.2.2 #define DPI_COUNT 6

Definition at line 54 of file structures.h.

Referenced by cmd_dpi(), cmd_dpisel(), loaddpi(), printdpi(), savedpi(), and updatedpi().

13.47.2.3 #define FEAT_ADJRATE 0x008

Definition at line 139 of file structures.h.

Referenced by _mkdevpath(), _setupusb(), and _start_dev().

13.47.2.4 #define FEAT_ANSI 0x200

Definition at line 146 of file structures.h.

Referenced by readcmd().

13.47.2.5 #define FEAT_BIND 0x010

Definition at line 140 of file structures.h.

Referenced by _mkdevpath(), main(), and readcmd().

13.47.2.6 #define FEAT_COMMON (FEAT_BIND | FEAT_NOTIFY | FEAT_FWVERSION | FEAT_MOUSEACCEL | FEAT_HWLOAD)

Definition at line 151 of file structures.h.

13.47.2.7 #define FEAT_FWUPDATE 0x080

Definition at line 143 of file structures.h.

Referenced by _mkdevpath(), _start_dev(), and cmd_fwupdate().

13.47.2.8 #define FEAT_FWVERSION 0x040

Definition at line 142 of file structures.h.

Referenced by _mkdevpath(), and _start_dev().

13.47.2.9 #define FEAT_HWLOAD 0x100

Definition at line 144 of file structures.h.

Referenced by _start_dev().

13.47.2.10 #define FEAT_ISO 0x400

Definition at line 147 of file structures.h.

Referenced by readcmd().

13.47.2.11 #define FEAT_LMASK (FEAT_ANSI | FEAT_ISO)

Definition at line 154 of file structures.h.

Referenced by readcmd().

13.47.2.12 #define FEAT_MONOCHROME 0x002

Definition at line 137 of file structures.h.

Referenced by _mkdevpath(), and _setupusb().

13.47.2.13 #define FEAT_MOUSEACCEL 0x800

Definition at line 148 of file structures.h.

Referenced by main(), and readcmd().

13.47.2.14 #define FEAT_NOTIFY 0x020

Definition at line 141 of file structures.h.

Referenced by _mkdevpath(), main(), and readcmd().

13.47.2.15 #define FEAT_POLLRATE 0x004

Definition at line 138 of file structures.h.

Referenced by _mkdevpath(), _start_dev(), and getfwversion().

13.47.2.16 #define FEAT_RGB 0x001

Definition at line 136 of file structures.h.

Referenced by _mkdevpath(), _start_dev(), revertusb(), and usbunclaim().

13.47.2.17 #define FEAT_STD_NRGB (FEAT_COMMON)

Definition at line 153 of file structures.h.

Referenced by _setupusb().

13.47.2.18 #define FEAT_STD_RGB (FEAT_COMMON | FEAT_RGB | FEAT_POLLRATE | FEAT_FWUPDATE)

Definition at line 152 of file structures.h.

Referenced by _setupusb().

13.47.2.19 #define HAS_ANY_FEATURE(*kb, feat*) (!!((kb)->features & (feat)))

Definition at line 158 of file structures.h.

13.47.2.20 #define HAS_FEATURES(*kb, feat*) (((kb)->features & (feat)) == (feat))

Definition at line 157 of file structures.h.

Referenced by _mkdevpath(), _start_dev(), cmd_fwupdate(), readcmd(), revertusb(), and usbunclaim().

13.47.2.21 #define HWMODE_K70 1

Definition at line 115 of file structures.h.

Referenced by cmd_hwload_kb(), and cmd_hwsave_kb().

13.47.2.22 #define HWMODE_K95 3

Definition at line 116 of file structures.h.

Referenced by cmd_hwload_kb(), and cmd_hwsave_kb().

13.47.2.23 #define HWMODE_MAX 3

Definition at line 117 of file structures.h.

13.47.2.24 #define I_CAPS 2

Definition at line 20 of file structures.h.

Referenced by _cmd_get(), iselect(), nprintind(), and updateindicators_kb().

13.47.2.25 #define I_NUM 1

Definition at line 19 of file structures.h.

Referenced by _cmd_get(), iselect(), nprintind(), and updateindicators_kb().

13.47.2.26 #define I_SCROLL 4

Definition at line 21 of file structures.h.

Referenced by _cmd_get(), iselect(), nprintind(), and updateindicators_kb().

13.47.2.27 #define IFACE_MAX 4

Definition at line 177 of file structures.h.

13.47.2.28 #define KB_NAME_LEN 40

Definition at line 174 of file structures.h.

Referenced by _setupusb(), and os_setupusb().

13.47.2.29 #define LIFT_MAX 5

Definition at line 56 of file structures.h.

Referenced by cmd_lift(), and loaddpi().

13.47.2.30 #define LIFT_MIN 1

Definition at line 55 of file structures.h.

Referenced by cmd_lift(), and loaddpi().

13.47.2.31 #define MACRO_MAX 1024

Definition at line 51 of file structures.h.

Referenced by _cmd_macro().

13.47.2.32 #define MD_NAME_LEN 16

Definition at line 82 of file structures.h.

Referenced by cmd_hwsave_kb(), cmd_hwsave_mouse(), cmd_name(), gethwmodename(), gethwprofilename(), getmodename(), hwloadmode(), hwtonative(), and nativetohw().

13.47.2.33 #define MODE_COUNT 6

Definition at line 100 of file structures.h.

Referenced by _freeprofile(), allocprofile(), and readcmd().

13.47.2.34 #define MSG_SIZE 64

Definition at line 176 of file structures.h.

Referenced by _usbsend(), cmd_hwload_kb(), cmd_hwload_mouse(), cmd_hwsave_kb(), cmd_hwsave_mouse(), cmd_pollrate(), fwupdate(), getfwversion(), hwloadmode(), loaddpi(), loadrgb_kb(), loadrgb_mouse(), os_inputmain(), os_usbrecv(), os_usbsend(), savedpi(), savergb_kb(), savergb_mouse(), setactive_kb(), setactive_mouse(), updatedpi(), updatergb_kb(), and updatergb_mouse().

13.47.2.35 #define NEEDS_FW_UPDATE(*kb*) ((kb)->fwversion == 0 && HAS_FEATURES((kb), FEAT_FWUPDATE | FEAT_FWVERSION))

Definition at line 161 of file structures.h.

Referenced by _start_dev(), readcmd(), revertusb(), setactive_kb(), and setactive_mouse().

13.47.2.36 #define OUTFIFO_MAX 10

Definition at line 24 of file structures.h.

Referenced by _mknotifynode(), _rmnotifynode(), inputupdate_keys(), nprintf(), readcmd(), rmdevpath(), and updateindicators_kb().

13.47.2.37 #define PR_NAME_LEN 16

Definition at line 99 of file structures.h.

Referenced by cmd_hwload_kb(), cmd_hwload_mouse(), cmd_profilename(), getprofilename(), hwtonative(), and nativetohw().

13.47.2.38 #define SCROLL_ACCELERATED 0

Definition at line 164 of file structures.h.

Referenced by readcmd().

13.47.2.39 #define SCROLL_MAX 10

Definition at line 166 of file structures.h.

Referenced by readcmd().

13.47.2.40 #define SCROLL_MIN 1

Definition at line 165 of file structures.h.

Referenced by readcmd().

13.47.2.41 #define SERIAL_LEN 34

Definition at line 175 of file structures.h.

Referenced by _setupusb(), and os_setupusb().

13.47.2.42 #define SET_KEYBIT(*array, index*) do { (array)[(index) / 8] |= 1 << ((index) % 8); } while(0)

Definition at line 15 of file structures.h.

Referenced by _cmd_macro(), cmd_notify(), corsair_mousecopy(), hid_kb_translate(), and hid_mouse_translate().

13.47.3 Variable Documentation

13.47.3.1 const union devcmd vtable_keyboard

Definition at line 52 of file device_vtable.c.

Referenced by get_vtable().

13.47.3.2 const union devcmd vtable_keyboard_nonrgb

Definition at line 99 of file device_vtable.c.

Referenced by get_vtable().

13.47.3.3 const union devcmd vtable_mouse

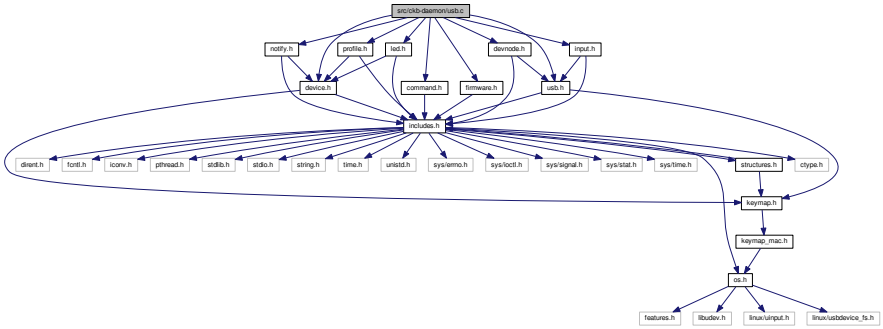
Definition at line 146 of file device_vtable.c.

Referenced by get_vtable().

13.48 src/ckb-daemon/usb.c File Reference

```
#include "command.h"
#include "device.h"
#include "devnode.h"
#include "firmware.h"
#include "input.h"
#include "led.h"
#include "notify.h"
#include "profile.h"
#include "usb.h"
```

Include dependency graph for usb.c:



Functions

- const char * vendor_str (short vendor)

  brief .
- const char * product_str (short product)

  brief .
- static const devcmd * get_vtable (short vendor, short product)

  brief .
- static void * devmain (usbdevice *kb)

  brief .
- static void * _setupusb (void *context)

  brief .
- void setupusb (usbdevice *kb)
- int revertusb (usbdevice *kb)
- int _resetusb (usbdevice *kb, const char *file, int line)
- int usb_tryreset (usbdevice *kb)
- int _usbsend (usbdevice *kb, const uchar *messages, int count, const char *file, int line)
- int _usbrecv (usbdevice *kb, const uchar *out_msg, uchar *in_msg, const char *file, int line)
- int closeusb (usbdevice *kb)

Variables

- pthread_mutex_t usbmutex = PTHREAD_MUTEX_INITIALIZER

  brief .
- volatile int reset_stop = 0

  brief .
- int features_mask = -1

  brief .
- int hwload_mode

  hwload_mode is defined in device.c

13.48.1 Function Documentation

13.48.1.1 int _resetusb (usbdevice * *kb*, const char * *file*, int *line*)

_resetusb Reset a USB device.

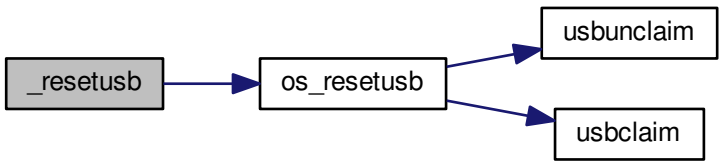
First reset the device via os_resetusb() after a long delay (it may send something to the host). If this worked (retval == 0), give the device another long delay Then perform the initialization via the device specific start() function entry in kb->vtable and if this is successful also, return the result of the device depenten updatergb() with force=true.

Definition at line 436 of file usb.c.

References usbdevice::active, DELAY_LONG, os_resetusb(), and usbdevice::vtable.

```
436                                                     {
437     // Perform a USB reset
438     DELAY_LONG(kb);
439     int res = os_resetusb(kb, file, line);
440     if(res)
441         return res;
442     DELAY_LONG(kb);
443     // Re-initialize the device
444     if(kb->vtable->start(kb, kb->active) != 0)
445         return -1;
446     if(kb->vtable->updatergb(kb, 1) != 0)
447         return -1;
448     return 0;
449 }
```

Here is the call graph for this function:



13.48.1.2 static void* _setupusb (void * *context*) [static]

_setupusb A horrible function for setting up an usb device

Parameters

context	As _setupusb() is called as a new thread, the kb∗ is transferred as void∗

Returns

a ptread_t∗ 0, here casted as void∗. Retval is always null

The basic structure of the function is somewhat habituated. It is more like an assembler routine than a structured program. This is not really bad, but just getting used to.

After every action, which can be practically fault-prone, the routine goes into the same error handling: It goes via goto to one of two exit labels. The difference is whether or not an unlock has to be performed on the imutex variable. In both cases, closeusb() is called, then an unlock is performed on the dmutex.

The only case where this error handling is not performed is the correct return of the call to devmain(). Here simply the return value of devmain() is passed to the caller.

In either case, the routine terminates with a void∗ 0 because either devmain() has returned constant null or the routine itself returns zero.

The basic idea of this routine is the following:

First some initialization of kb standard structured and local vars is done.

- **kb** is set to the pointer given from start environment
- local vars **vendor** and **product** are set to the values from the corresponding fields of kb
- local var **vt and** the **kb-**>**vtable** are both set to the retval of *get_vtable()*
- **kb-**>**features** are set depending on the type of hardware connected:
  - set either to standard non rgb (all common flags like binding, notify, FW, hardware-loading etc) or in case of RGB-device set to standard + RGB, pollrate-change and fw-update
  - exclude all features which are disabled via feature_mask (set by daemon CLI parameters)
  - if it is a mouse, add adjust-rate
  - if it is a monochrome device, set the flag for RGB-protocol, but single color
- the standard delay time is initialized in kb->usbdelay
- A fixed 100ms wait is the start. **Although the DELAY_LONG macro is given a parameter, it is ignored. Occasionally refactor it.**
- The first relevant point is the operating system-specific opening of the interface in os_setupusb(). As a result, some parameters should be set in kb (name, serial, fwversion, epcount = number of usb endpoints), and all endpoints should be claimed with usbclaim(). Claiming is the only point where os_setupusb() can produce an error (-1, otherwise 0).
- The following two statements deal with possible errors when setting the kb values in the current routine: If the version or the name was not read correctly, they are set to default values:
  - serial is set to "<vendor>: <product> -NoID"
  - the name is set to "<vendor> <product>".
- Then the user level input subsystem is activated via os_openinput(). There are two file descriptors, one for the mouse and one for the keyboard. **As mentioned in structures.h, not the just opened FD numbers are stored under kb-**>**uinput_kb or kb-**>**uinput_mouse, but the values increased by 1!** The reason is, if the open fails or not open has been done until now, that struct member is set to 0, not to -1 or other negative value. So all usage of this kb->handle must be something like "`kb->handle - 1`", as you can find it in the code.
- The next action is to create a separate thread, which gets as parameter kb and starts with os_inputmain(). The thread is immediately detached so that it can return its resource completely independently if it should terminate.

- The same happens with os_setupindicators(), which initially initializes all LED variables in kb to off and then starts the _ledthread() thread with kb as parameter and then detaches it. Here again only the generation of the thread can fail.

- Via an entry in the vable (allocprofile, identical for all three vtable types), allocprofile() is called in profile.c. With a valid parameter kb, a usbprofile structure is allocated and stored as a kb->profile. Then initmode() is called for each of the initializable modes (MODE_COUNT, currently 6). This procedure creates the memory space for the mode information, initializes the range to 0, and then sets the light.forceupdate and dpi.forceupdate to true. This forces an update later in the initialization of the device.

  The first mode is set as the current mode and two force flags are set (this seems to be mode-intersecting flags for light and update).

  Warning

  There is no error handling for the allocprofile() and initmode() procedures. However, since they allocate storage areas, the subsequent assignments and initializations can run in a SEGV.

- Not completely understandable is why now via the vtable the function updateindicators() is called. But this actually happens in the just started thread _ledthread(). Either the initialization is wrong und must done here with force or the overview is lost, what happens when...

  Regardless: For a mouse nothing happens here, for a keyboard updateindicators_kb() is called via the entry in kb->vtable. The first parameter is kb again, the second is constant 1 (means force = true). This causes the LED status to be sent after a 5ms delay via os_sendindicators() (ioctl with a `usbdevfs_ctrltransfer`).

  The notification is sent to all currently open notification channels then.

  Setupindicators() and with it updateindicators_kb() can fail.

- From this point - if an error is detected - the error label is addressed by goto statement, which first performs an unlock on the imutex. This is interesting because the next statement is exactly this: An unlock on the imutex.

- Via vtable the *kb->start()* function is called next. This is the same for a mouse and an RGB keyboard: start_dev(), for a non RGB keyboard it is start_kb_nrgb().

  First parameter is as always kb, second is 0 (makeactive = false).

  - In start_kb_nrgb() set the keyboard into a so-called software mode (NK95_HWOFF) via ioctl with `usbdevfs_ctrltransfer` in function _nk95cmd(), which will in turn is called via macro nk95cmd() via start_kb_nrgb().

    Then two dummy values (active and pollrate) are set in the kb structure and ready.
  - start_dev() does a bit more - because this function is for both mouse and keyboard. start_dev() calls - after setting an extended timeout parameter - _start_dev(). Both are located in device.c.
  - First, _start_dev() attempts to determine the firmware version of the device, but only if two conditions are met: hwload-mode is not null (then hw-loading is disabled) and the device has the FEAT_HWLOAD feature. Then the firmware and the poll rate are fetched via getfwversion().

    If hwload_mode is set to "load only once" (==1), then the HWLOAD feature is masked, so that no further reading can take place.
  - Now check if device needs a firmware update. If so, set it up and leave the function without error.
  - Else load the hardware profile from device if the hw-pointer is not set and hw-loading is possible and allowed.

    Return error if mode == 2 (load always) and loading got an error. Else mask the HWLOAD feature, because hwload must be 1 and the error couold be a repeated hw-reading.

    Puh, that is real Horror code. It seems to be not faulty, but completely unreadable.
  - Finally, the second parameter of _startdev() is used to check whether the device is to be activated. Depending on the parameter, the active or the idle-member in the correspondig vtable is called. These are device-dependent again:

Device	active	idle
RGB Keyboard	cmd_active_kb() means: start the device with a lot of kb-specific initializers (software controlled mode)	cmd_idle_kb() set the device with a lot of kb-specific initializers into the hardware controlled mode)
non RGB Keyboard	cmd_io_none() means: Do nothing	cmd_io_none() means: Do nothing
Mouse	cmd_active_mouse() similar to cmd_active_kb()	cmd_idle_mouse similar to cmd_idle_kb()

- If either *start()* succeeded or the next following usb_tryreset(), it goes on, otherwise again a hard abort occurs.
- Next, go to mkdevpath(). After securing the EUID (effective UID) especially for macOS, work starts really in _mkdevpath(). Create - no matter how many devices were registered - either the ckb0/ files **version**, **pid** and **connected** or the **cmd** command fifo, the first notification fifo **notify0**, **model** and **serial** as well as the **features** of the device and the **pollrate**.
- If all this is done and no error has occurred, a debug info is printed ("Setup finished for ckbx") updateconnected() writes the new device into the text file under ckb0/ and devmain() is called.

devmain()'s return value is returned by _setupusb() when we terminate.

- The remaining code lines are the two exit labels as described above

Definition at line 224 of file usb.c.

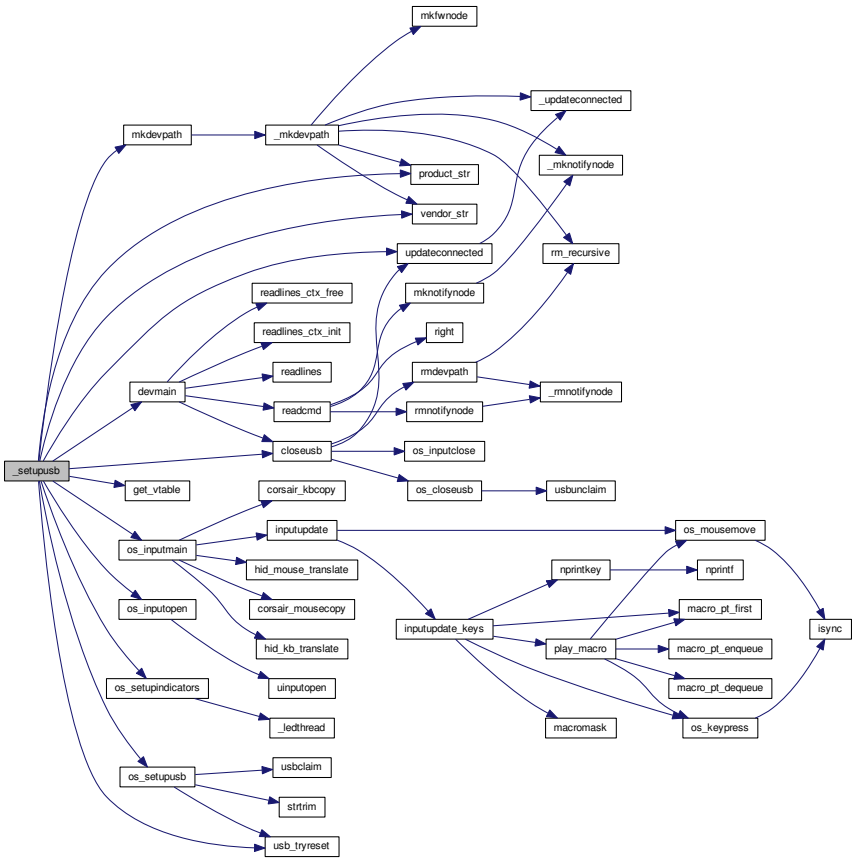
References ckb_info, closeusb(), DELAY_LONG, devmain(), devpath, dmutex, FEAT_ADJRATE, FEAT_MONOCHROME, FEAT_STD_NRGB, FEAT_STD_RGB, usbdevice::features, features_mask, get_vtable(), imutex, INDEX_OF, usbdevice::inputthread, IS_MONOCHROME, IS_MOUSE, IS_RGB, KB_NAME_LEN, keyboard, mkdevpath(), usbdevice::name, os_inputmain(), os_inputopen(), os_setupindicators(), os_setupusb(), usbdevice::product, product_str(), usbdevice::serial, SERIAL_LEN, updateconnected(), USB_DELAY_DEFAULT, usb_tryreset(), usbdevice::usbdelay, usbdevice::vendor, vendor_str(), and usbdevice::vtable.

Referenced by setupusb().

```
224                                     {
237     usbdevice* kb = context;
238     // Set standard fields
239     short vendor = kb->vendor, product = kb->product;
240     const devcmd* vt = kb->vtable = get_vtable(vendor, product);
241     kb->features = (IS_RGB(vendor, product) ? FEAT_STD_RGB :
    FEAT_STD_NRGB) & features_mask;
242     if(IS_MOUSE(vendor, product)) kb->features |= FEAT_ADJRATE;
243     if(IS_MONOCHROME(vendor, product)) kb->features |=
    FEAT_MONOCHROME;
244     kb->usbdelay = USB_DELAY_DEFAULT;
245
246     // Perform OS-specific setup
250     DELAY_LONG(kb);
251
257     if(os_setupusb(kb))
258         goto fail;
259
265     // Make up a device name and serial if they weren't assigned
266     if(!kb->serial[0])
267         snprintf(kb->serial, SERIAL_LEN, "%04x:%04x-NoID", kb->
    vendor, kb->product);
268     if(!kb->name[0])
269         snprintf(kb->name, KB_NAME_LEN, "%s %s", vendor_str(kb->
    vendor), product_str(kb->product));
270
271     // Set up an input device for key events
279     if(os_inputopen(kb))
280         goto fail;
284     if(pthread_create(&kb->inputthread, 0, os_inputmain, kb))
285         goto fail;
286     pthread_detach(kb->inputthread);
292     if(os_setupindicators(kb))
293         goto fail;
294
295     // Set up device
308     vt->allocprofile(kb);
```

```
319     vt->updateindicators(kb, 1);
324     pthread_mutex_unlock(imutex(kb));
358     if(vt->start(kb, 0) && usb_tryreset(kb))
359         goto fail_noinput;
365     // Make /dev path
366     if(mkdevpath(kb))
367         goto fail_noinput;
373     // Finished. Enter main loop
374     int index = INDEX_OF(kb, keyboard);
375     ckb_info("Setup finished for %s%d\n", devpath, index);
376     updateconnected();
379     return devmain(kb);
382     fail:
383     pthread_mutex_unlock(imutex(kb));
384     fail_noinput:
385     closeusb(kb);
386     pthread_mutex_unlock(dmutex(kb));
387     return 0;
388 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.48.1.3 int _usbrecv (usbdevice * *kb,* const uchar * *out_msg,* uchar * *in_msg,* const char * *file,* int *line*)

_usbrecv Request data from a USB device by first sending an output packet and then reading the response.

To fully understand this, you need to know about usb: All control is at the usb host (the CPU). If the device wants to communicate something to the host, it must wait for the host to ask. The usb protocol defines the cycles and periods in which actions are to be taken.

So in order to receive a data packet from the device, the host must first send a send request.

This is done by _usbrecv() in the first block by sending the MSG_SIZE large data block from **out_msg** via os-_usbsend() as it is a machine depending implementation. The usb target device is as always determined over kb.

For os_usbsend() to know that it is a receive request, the **is_recv** parameter is set to true (1). With this, os_usbsend () generates a control package for the hardware, not a data packet.

If sending of the control package is not successful, a maximum of 5 times the transmission is repeated (including the first attempt). If a non-cancelable error is signaled or the drive is stopped via reset_stop, _usbrecv() immediately returns 0.

After this, the function waits for the requested response from the device using os_usbrecv ().

os_usbrecv() returns 0, -1 or something else.

Zero signals a serious error which is not treatable and _usbrecv() also returns 0.

-1 means that it is a treatable error - a timeout for example - and therefore the next transfer attempt is started after a long pause (DELAY_LONG) if not reset_stop or the wrong hwload_mode require a termination with a return value of 0.

After 5 attempts, _usbrecv () returns and returns 0 as well as an error message.

When data is received, the number of received bytes is returned. This should always be MSG_SIZE, but os_-usbrecv() can also return less. It should not be more, because then there would be an unhandled buffer overflow, but it could be less. This would be signaled in os_usbrecv () with a message.

The buffers behind **out_msg** and **in_msg** are MSG_SIZE at least (currently 64 Bytes). More is ok but useless, less brings unpredictable behavior. < Synchonization between macro and color information

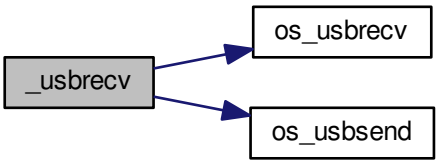
Definition at line 611 of file usb.c.

References ckb_err_fn, DELAY_LONG, DELAY_MEDIUM, DELAY_SHORT, hwload_mode, mmutex, os_usbrecv(), os_usbsend(), and reset_stop.

```
611                                                                                                 {
612     // Try a maximum of 5 times
613     for (int try = 0; try < 5; try++) {
614         // Send the output message
615         pthread_mutex_lock(mmutex(kb));
616         DELAY_SHORT(kb);
617         int res = os_usbsend(kb, out_msg, 1, file, line);
618         pthread_mutex_unlock(mmutex(kb));
619         if (res == 0)
620             return 0;
621         else if (res == -1) {
622             // Retry on temporary failure
623             if (reset_stop)
624                 return 0;
625             DELAY_LONG(kb);
626             continue;
627         }
628         // Wait for the response
629         DELAY_MEDIUM(kb);
630         res = os_usbrecv(kb, in_msg, file, line);
631         if(res == 0)
632             return 0;
633         else if(res != -1)
634             return res;
635         if(reset_stop || hwload_mode != 2)
636             return 0;
637         DELAY_LONG(kb);
638     }
639     // Give up
640     ckb_err_fn("Too many send/recv failures. Dropping.\n", file, line);
641     return 0;
```

```
642 }
```

Here is the call graph for this function:



13.48.1.4 int _usbsend (usbdevice * *kb,* const uchar * *messages,* int *count,* const char * *file,* int *line*)

_usbsend send a logical message completely to the given device

Todo A lot of different conditions are combined in this code. Don't think, it is good in every combination...

The main task of _usbsend () is to transfer the complete logical message from the buffer beginning with *messages* to **count** * **MSG_SIZE**.

According to usb 2.0 specification, a USB transmits a maximum of 64 byte user data packets. For the transmission of longer messages we need a segmentation. And that is exactly what happens here.

The message is given one by one to os_usbsend() in MSG_SIZE (= 64) byte large bites.

Attention

This means that the buffer given as argument must be n * MSG_SIZE Byte long.

An essential constant parameter which is relevant for os_usbsend() only is is_recv = 0, which means sending.

Now it gets a little complicated again:

- If os_usbsend() returns 0, only zero bytes could be sent in one of the packets, or it was an error (-1 from the systemcall), but not a timeout. How many Bytes were sent in total from earlier calls does not seem to matter, _usbsend() returns a total of 0.
- Returns os_usbsend() -1, first check if **reset_stop** is set globally or (incomprehensible) hwload_mode is not set to "always". In either case, _usbsend() returns 0, otherwise it is assumed to be a temporary transfer error and it simply retransmits the physical packet after a long delay.
- If the return value of os_usbsend() was neither 0 nor -1, it specifies the numer of bytes transferred.

  Here is an information hiding conflict with os_usbsend() (at least in the Linux version):

  If os_usbsend() can not transfer the entire packet, errors are thrown and the number of bytes sent is returned. _usbsend() interprets this as well and remembers the total number of bytes transferred in the local variable **total_sent**. Subsequently, however, transmission is continued with the next complete MSG_SIZE block and not with the first of the possibly missing bytes.

  Todo Check whether this is the same in the macOS variant. It is not dramatic, but if errors occur, it can certainly irritate the devices completely if they receive incomplete data streams. Do we have errors with the messages "Wrote YY bytes (expected 64)" in the system logs? If not, we do not need to look any further.

When the last packet is transferred, _usbsend() returns the effectively counted set of bytes (from **total_sent**). This at least gives the caller the opportunity to check whether something has been lost in the middle.

A bit strange is the structure of the program: Handling the **count** MSG_SIZE blocks to be transferred is done in the outer for (...) loop. Repeating the transfer with a treatable error is managed by the inner while(1) loop.

This must be considered when reading the code; The "break" on successful block transfer leaves the inner while, not the for (...). < Synchonization between macro and color information

Definition at line 542 of file usb.c.

References DELAY_LONG, DELAY_SHORT, hwload_mode, mmutex, MSG_SIZE, os_usbsend(), and reset_stop.

```
542                                                                                                    {
543     int total_sent = 0;
544     for(int i = 0; i < count; i++){
545         // Send each message via the OS function
546         while(1){
547             pthread_mutex_lock(mmutex(kb));
548             DELAY_SHORT(kb);
549             int res = os_usbsend(kb, messages + i * MSG_SIZE, 0, file, line);
550             pthread_mutex_unlock(mmutex(kb));
551             if(res == 0)
552                 return 0;
553             else if(res != -1){
554                 total_sent += res;
555                 break;
556             }
557             // Stop immediately if the program is shutting down or hardware load is set to tryonce
558             if(reset_stop || hwload_mode != 2)
559                 return 0;
560             // Retry as long as the result is temporary failure
561             DELAY_LONG(kb);
562         }
563     }
564     return total_sent;
565 }
```

Here is the call graph for this function:

13.48.1.5 int closeusb (usbdevice * *kb*)

closeusb Close a USB device and remove device entry.

An imutex lock ensures first of all, that no communication is currently running from the viewpoint of the driver to the user input device (ie the virtual driver with which characters or mouse movements are sent from the daemon to the operating system as inputs).

If the **kb** has an acceptable value != 0, the index of the device is looked for and with this index os_inputclose() is called. After this no more characters can be sent to the operating system.

Then the connection to the usb device is capped by os_closeusb().

Todo What is not yet comprehensible is the call to updateconnected() BEFORE os_closeusb(). Should that be in the other sequence? Or is updateconnected() not displaying the connected usb devices, but the representation which uinput devices are loaded? Questions about questions ...

If there is no valid **handle**, only updateconnected() is called. We are probably trying to disconnect a connection under construction. Not clear.

The cmd pipe as well as all open notify pipes are deleted via rmdevpath ().

This means that nothing can happen to the input path - so the device-specific imutex is unlocked again and remains unlocked.

Also the dmutex is unlocked now, but only to join the thread, which was originally taken under **kb->thread** (which started with _setupusb()) with pthread_join() again. Because of the closed devices that thread would have to quit sometime

See Also

the hack note with rmdevpath())

As soon as the thread is caught, the dmutex is locked again, which is what I do not understand yet: What other thread can do usb communication now?

If the vtabel exists for the given kb (why not? It seems to have race conditions here!!), via the vtable the actually device-specific, but still everywhere identical freeprofile() is called. This frees areas that are no longer needed. Then the **usbdevice** structure in its array is set to zero completely.

Error handling is rather unusual in closeusb(); Everything works (no matter what the called functions return), and closeusb() always returns zero (success).

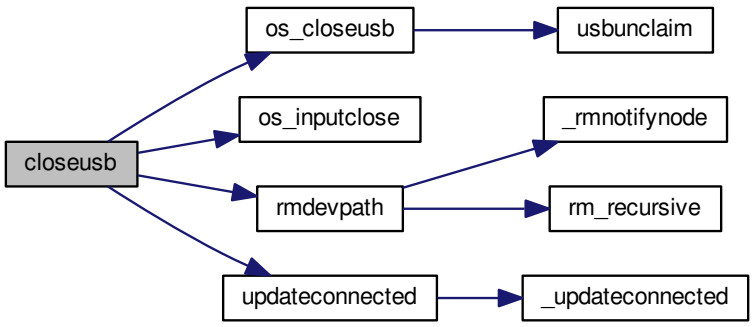
Definition at line 687 of file usb.c.

References ckb_info, devpath, dmutex, usbdevice::handle, imutex, INDEX_OF, keyboard, os_closeusb(), os_inputclose(), rmdevpath(), usbdevice::thread, updateconnected(), and usbdevice::vtable.

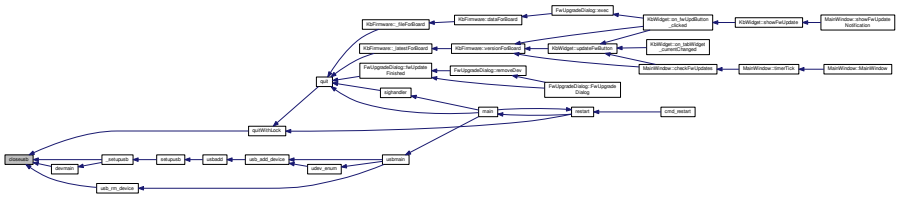
Referenced by _setupusb(), devmain(), quitWithLock(), and usb_rm_device().

```
687                             {
688     pthread_mutex_lock(imutex(kb));
689     if(kb->handle){
690         int index = INDEX_OF(kb, keyboard);
691         ckb_info("Disconnecting %s%d\n", devpath, index);
692         os_inputclose(kb);
693         updateconnected();
694         // Close USB device
695         os_closeusb(kb);
696     } else
697         updateconnected();
698     rmdevpath(kb);
699 
700     // Wait for thread to close
701     pthread_mutex_unlock(imutex(kb));
702     pthread_mutex_unlock(dmutex(kb));
703     pthread_join(kb->thread, 0);
704     pthread_mutex_lock(dmutex(kb));
705 
706     // Delete the profile and the control path
707     if(!kb->vtable)
708         return 0;
709     kb->vtable->freeprofile(kb);
710     memset(kb, 0, sizeof(usbdevice));
711     return 0;
712 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.48.1.6 static void* devmain (usbdevice * *kb*) [static]

devmain is called by _setupusb

Parameters

kb	the pointer to the device. Even if it has the name kb, it is valid also for a mouse (the whole driver seems to be implemented first for a keyboard).

Returns

always a nullptr

Synchronization

The syncing via mutexes is interesting:

1. *imutex* (the Input mutex)

   This one is locked in `setupusb()`. That function does only two things: Locking the mutex and trying to start a thread at `_setupusb()`. _setupusb() unlocks *imutex* after getting some buffers and initalizing internal structures from the indicators (this function often gets problems with error messages like "unable to read indicators" or "Timeout bla blubb").

Warning

have a look at *updateindicators()* later.
if creating the thread is not successful, the *imutex* remains blocked. Have a look at setupusb() later.

2. *dmutex* (the Device mutex)

   This one is very interesting, because it is handled in devmain(). It seems that it is locked only in *_ledthread()*, which is a thread created in *os_setupindicators()*. os_setupindicators() again is called in *_setupusb()* long before calling devmain(). So this mutex is locked when we start the function as the old comment says.

   Before reading from the FIFO and direct afterwards an unlock..lock sequence is implemented here. Even if only the function readlines() should be surrounded by the unlock..lock, the variable definition of the line pointer is also included here. Not nice, but does not bother either. Probably the Unlock..lock is needed so that now another process can change the control structure *linectx* while we wait in readlines().

   Todo Hope to find the need for dmutex usage later.
   Should this function be declared as pthread_t∗ function, because of the defintion of pthread-create? But void∗ works also...

Attention

dmutex should still be locked when this is called

First a *readlines_ctx* buffer structure is initialized by `readlines_ctx_init()`.

After some setup functions, beginning in _setupusb() which has called devmain(), we read the command input--Fifo designated to that device in an endless loop. This loop has two possible exits (plus reaction to signals, not mentioned here).

If the reading via readlines() is successful (we might have read multiple lines), the interpretation is done by readcmd() iff the connection to the device is still available (checked via IS_CONNECTED(kb)). This is true if the kb-structure has a handle and an event pointer both != Null). If not, the loop is left (the first exit point).

if nothing is in the line buffer (some magic interrupt?), continue in the endless while without any reaction.

Todo readcmd() gets a **line**, not **lines**. Have a look on that later.

Is the condition IS_CONNECTED valid? What functions change the condititon for the macro?

If interpretation and communication with the usb device got errors, they are signalled by readcmd() (non zero retcode). In this case the usb device is closed via closeusb() and the endless loop is left (the second exit point).

After leaving the endless loop the readlines-ctx structure and its buffers are freed by readlines_ctx_free().

Definition at line 145 of file usb.c.

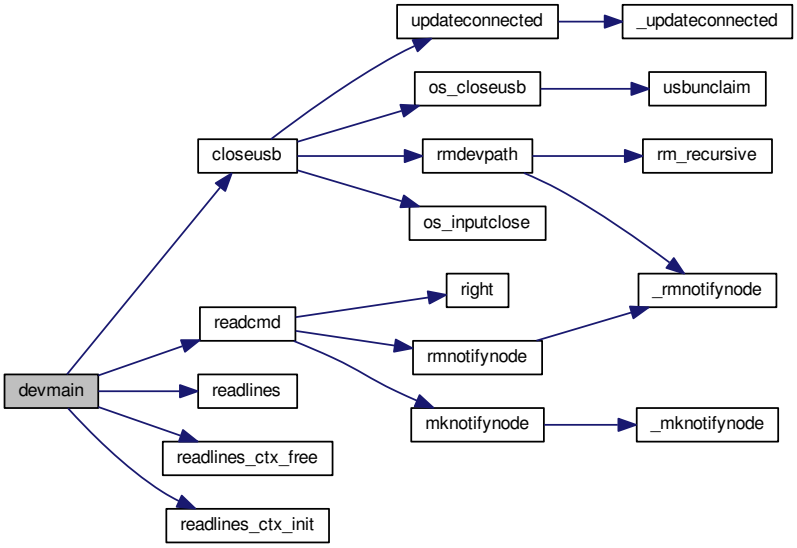
References closeusb(), dmutex, usbdevice::infifo, IS_CONNECTED, readcmd(), readlines(), readlines_ctx_free(), and readlines_ctx_init().

Referenced by _setupusb().

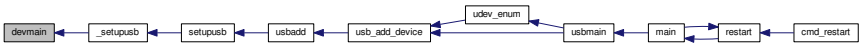
```
145                                     {
147     int kbfifo = kb->infifo - 1;
150     readlines_ctx linectx;
151     readlines_ctx_init(&linectx);
156     while(1){
163         pthread_mutex_unlock(dmutex(kb));
164         // Read from FIFO
165         const char* line;
166         int lines = readlines(kbfifo, linectx, &line);
167         pthread_mutex_lock(dmutex(kb));
168         // End thread when the handle is removed
169         if(!IS_CONNECTED(kb))
170             break;
174         if(lines){
177             if(readcmd(kb, line)){
183                 // USB transfer failed; destroy device
184                 closeusb(kb);
185                 break;
```

```
186             }
187         }
188     }
189     pthread_mutex_unlock(dmutex(kb));
192     readlines_ctx_free(linectx);
193     return 0;
194 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.48.1.7 static const devcmd* get_vtable (short *vendor*, short *product*) [static]

get_vtable returns the correct vtable pointer

Parameters

vendor	short usb vendor ID
product	short usb product ID

Returns

Depending on the type and model, the corresponding vtable pointer is returned (see below)

At present, we have three different vtables:

- `vtable_mouse` is used for all mouse types. This may be wrong with some newer mice?
- `vtable_keyboard` is used for all RGB Keyboards.

- `vtable_keyboard_nonrgb` for all the rest.

Todo Is the last point really a good decision and always correct?

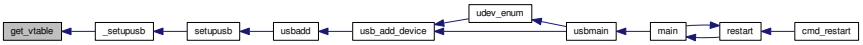
Definition at line 112 of file usb.c.

References IS_MOUSE, IS_RGB, vtable_keyboard, vtable_keyboard_nonrgb, and vtable_mouse.

Referenced by _setupusb().

```
112                                                                 {
113     return IS_MOUSE(vendor, product) ? &vtable_mouse :
      IS_RGB(vendor, product) ? &vtable_keyboard : &
      vtable_keyboard_nonrgb;
114 }
```

Here is the caller graph for this function:



13.48.1.8 const char* product_str (short *product*)

product_str returns a condensed view on what type of device we have.

At present, various models and their properties are known from corsair products. Some models differ in principle (mice and keyboards), others differ in the way they function (for example, RGB and non RGB), but they are very similar.

Here, only the first point is taken into consideration and we return a unified model string. If the model is not known with its number, *product_str* returns an empty string.

The model numbers and corresponding strings wwith the numbers in hex-string are defined in `usb.h`

At present, this function is used to initialize `kb->name` and to give information in debug strings.

Attention

The combinations below have to fit to the combinations in the macros mentioned above. So if you add a device with a new number, change both.

Todo There are macros defined in usb.h to detect all the combinations below. the only difference is the parameter: The macros need the *kb**, product_str() needs the *product ID*

Definition at line 70 of file usb.c.

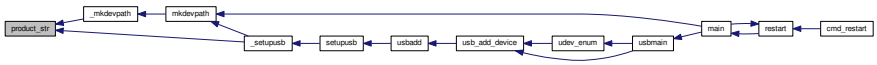
References P_GLAIVE, P_HARPOON, P_K63_NRGB, P_K65, P_K65_LUX, P_K65_NRGB, P_K65_RFIRE, P_K68, P_K70, P_K70_LUX, P_K70_LUX_NRGB, P_K70_NRGB, P_K70_RFIRE, P_K70_RFIRE_NRGB, P_K95, P_K95_NRGB, P_K95_PLATINUM, P_M65, P_M65_PRO, P_SABRE_L, P_SABRE_N, P_SABRE_O, P_SABRE_O2, P_SCIMITAR, P_SCIMITAR_PRO, P_STRAFE, P_STRAFE_NRGB, and P_STRAFE_NRGB_2.

Referenced by _mkdevpath(), and _setupusb().

```
70                                      {
71     if(product == P_K95 || product == P_K95_NRGB)
72         return "k95";
73     if(product == P_K95_PLATINUM)
74         return "k95p";
75     if(product == P_K70 || product == P_K70_NRGB || product ==
      P_K70_LUX || product == P_K70_LUX_NRGB || product ==
      P_K70_RFIRE || product == P_K70_RFIRE_NRGB)
76         return "k70";
```

```
77     if(product == P_K68)
78         return "k68";
79     if(product == P_K65 || product == P_K65_NRGB || product ==
       P_K65_LUX || product == P_K65_RFIRE)
80         return "k65";
81     if(product == P_K63_NRGB)
82         return "k63";
83     if(product == P_STRAFE || product == P_STRAFE_NRGB || product ==
       P_STRAFE_NRGB_2)
84         return "strafe";
85     if(product == P_M65 || product == P_M65_PRO)
86         return "m65";
87     if(product == P_SABRE_O || product == P_SABRE_L || product ==
       P_SABRE_N || product == P_SABRE_O2)
88         return "sabre";
89     if(product == P_SCIMITAR || product == P_SCIMITAR_PRO)
90         return "scimitar";
91     if(product == P_HARPOON)
92         return "harpoon";
93     if(product == P_GLAIVE)
94         return "glaive";
95     return "";
96 }
```

Here is the caller graph for this function:



13.48.1.9 int revertusb (usbdevice * *kb*)

revertusb sets a given device to inactive (hardware controlled) mode if not a fw-ugrade is indicated

First is checked, whether a firmware-upgrade is indicated for the device. If so, revertusb() returns 0.

Todo Why is this useful? Are there problems seen with deactivating a device with older fw-version??? Why isn't this an error indicating reason and we return success (0)?

Anyway, the following steps are similar to some other procs, dealing with low level usb handling:

- If we do not have an RGB device, a simple setting to Hardware-mode (NK95_HWON) is sent to the device via n95cmd().

  Todo The return value of nk95cmd() is ignored (but sending the ioctl may produce an error and _nk95_cmd will indicate this), instead revertusb() returns success in any case.

- If we have an RGB device, setactive() is called with second param active = false. That function will have a look on differences between keyboards and mice.

  More precisely setactive() is just a macro to call via the kb->vtable enties either the active() or the idle() function where the vtable points to. setactive() may return error indications. If so, revertusb() returns -1, otherwise 0 in any other case.

Definition at line 417 of file usb.c.

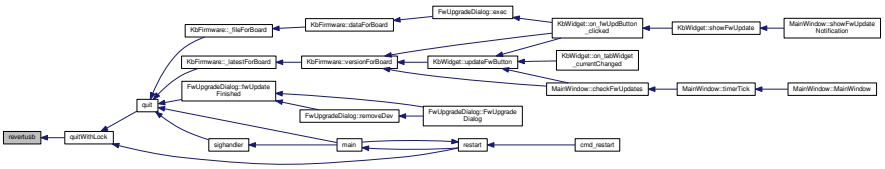
References FEAT_RGB, HAS_FEATURES, NEEDS_FW_UPDATE, NK95_HWON, nk95cmd, and setactive.

Referenced by quitWithLock().

```
417                             {
418     if(NEEDS_FW_UPDATE(kb))
419         return 0;
420     if(!HAS_FEATURES(kb, FEAT_RGB)){
421         nk95cmd(kb, NK95_HWON);
422         return 0;
423     }
```

```
424     if(setactive(kb, 0))
425         return -1;
426     return 0;
427 }
```

Here is the caller graph for this function:



13.48.1.10 void setupusb (usbdevice * *kb*)

setupusb starts a thread with kb as parameter and _setupusb() as entrypoint.

Set up a USB device after its handle is open. Spawns a new thread _setupusb() with standard parameter kb. dmutex must be locked prior to calling this function. The function will unlock it when finished. In kb->thread the thread id is mentioned, because closeusb() needs this info for joining that thread again.

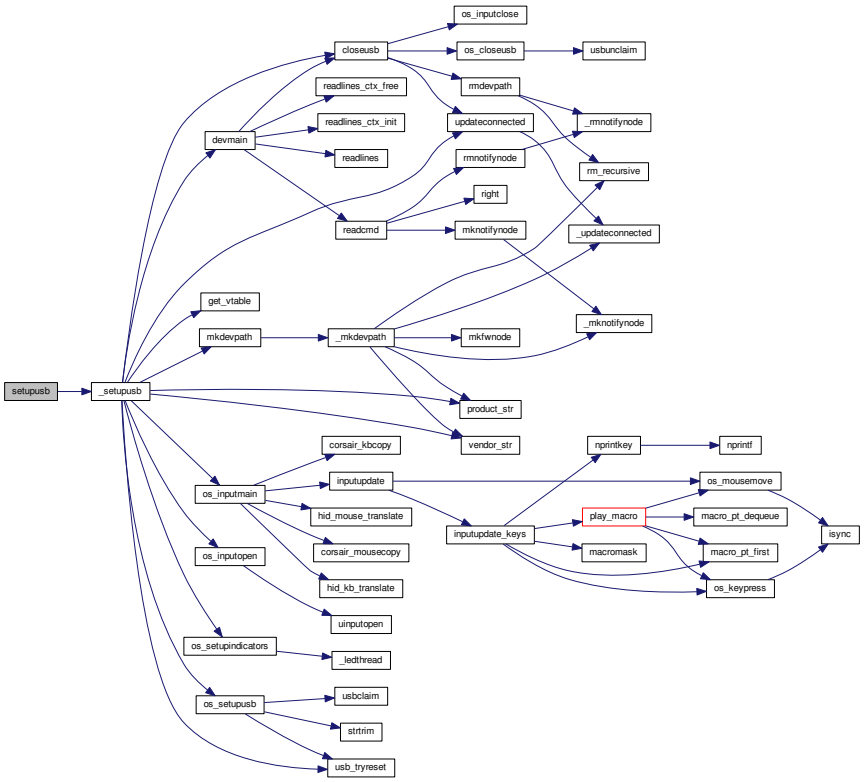
Definition at line 396 of file usb.c.

References _setupusb(), ckb_err, imutex, and usbdevice::thread.

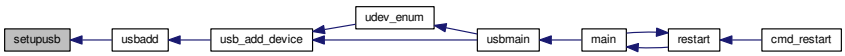
Referenced by usbadd().

```
396                         {
397     pthread_mutex_lock(imutex(kb));
398     if(pthread_create(&kb->thread, 0, _setupusb, kb))
399         ckb_err("Failed to create USB thread\n");
400 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.48.1.11 int usb_tryreset (usbdevice * *kb*)

usb_tryreset does what the name means: Try to reset the usb via resetusb()

This function is called if an usb command ran into an error in case of one of the following two situations:

- When setting up a new usb device and the start() function got an error (

  See Also

  _setupusb())

- If upgrading to a new firmware gets an error (

  See Also

  cmd_fwupdate()).

  The previous action which got the error will NOT be re-attempted.

In an endless loop usb_tryreset() tries to reset the given usb device via the macro resetusb().

This macro calls _resetusb() with debugging information.

_resetusb() sends a command via the operating system dependent function os_resetusb() and - if successful - reinitializes the device. os_resetusb() returns -2 to indicate a broken device and all structures should be removed for it.

In that case, the loop is terminated, an error message is produced and usb_tryreset() returns -1.

In case resetusb() has success, the endless loop is left via a return 0 (success).

If the return value from resetusb() is -1, the loop is continued with the next try.

If the global variable **reset_stop** is set directly when the function is called or after each try, usb_tryreset() stops working and returns -1.

Todo Why does usb_tryreset() hide the information returned from resetusb()? Isn't it needed by the callers?

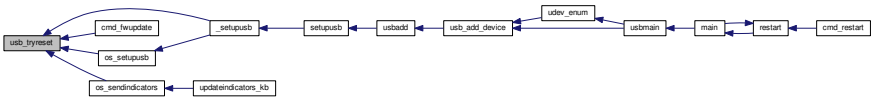
Definition at line 475 of file usb.c.

References ckb_err, ckb_info, reset_stop, and resetusb.

Referenced by _setupusb(), cmd_fwupdate(), os_sendindicators(), and os_setupusb().

```
475                                 {
476     if(reset_stop)
477         return -1;
478     ckb_info("Attempting reset...\n");
479     while(1){
480         int res = resetusb(kb);
481         if(!res){
482             ckb_info("Reset success\n");
483             return 0;
484         }
485         if(res == -2 || reset_stop)
486             break;
487     }
488     ckb_err("Reset failed. Disconnecting.\n");
489     return -1;
490 }
```

Here is the caller graph for this function:



13.48.1.12 const char* vendor_str (short *vendor*)

uncomment to see USB packets sent to the device

vendor_str returns "corsair" if the given *vendor* argument is equal to *V_CORSAIR* (0x1bc) else it returns ""

Attention

There is also a string defined V_CORSAIR_STR, which returns the device number as string in hex "1b1c".

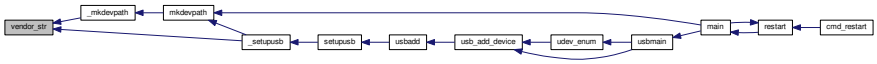
Definition at line 43 of file usb.c.

References V_CORSAIR.

Referenced by _mkdevpath(), and _setupusb().

```
43                                      {
44     if(vendor == V_CORSAIR)
45         return "corsair";
46     return "";
47 }
```

Here is the caller graph for this function:



13.48.2 Variable Documentation

13.48.2.1 int features_mask = -1

features_mask Mask of features to exclude from all devices

That bit mask ist set to enable all (-1). When interpreting the input parameters, some of these bits can be cleared.

At the moment binding, notifying and mouse-acceleration can be disabled via command line.

Have a look at *main()* in main.c for details.

Definition at line 35 of file usb.c.

Referenced by _setupusb(), and main().

13.48.2.2 int hwload_mode

hwload_mode is defined in device.c

Definition at line 7 of file device.c.

Referenced by _start_dev(), _usbrecv(), and _usbsend().

13.48.2.3 volatile int reset_stop = 0

reset_stop is boolean: Reset stopper for when the program shuts down.

Is set only by *quit()* to true (1) to inform several usb_∗ functions to end their loops and tries.

Definition at line 25 of file usb.c.

Referenced by _usbrecv(), _usbsend(), quitWithLock(), and usb_tryreset().

13.48.2.4 pthread_mutex_t usbmutex = PTHREAD_MUTEX_INITIALIZER

usbmutex is a never referenced mutex!

Todo We should have a look why this mutex is never used.

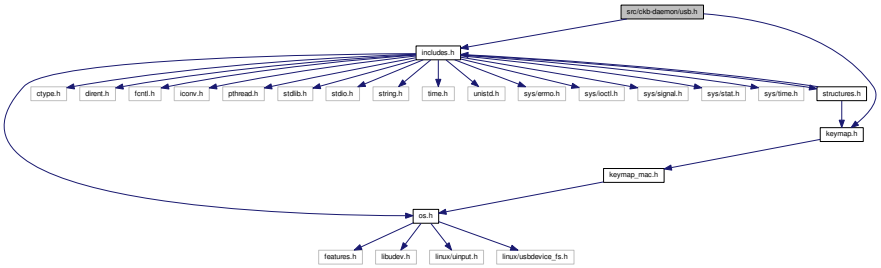
Definition at line 17 of file usb.c.

13.49 src/ckb-daemon/usb.h File Reference

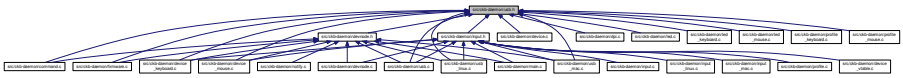
Definitions for using USB interface.

```
#include "includes.h"
#include "keymap.h"
```

Include dependency graph for usb.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define V_CORSAIR 0x1b1c

  For the following Defines please see "Detailed Description".
- #define V_CORSAIR_STR "1b1c"
- #define P_K63_NRGB 0x1b40
- #define P_K63_NRGB_STR "1b40"
- #define IS_K63(kb) ((kb)->vendor == V_CORSAIR && (kb)->product == P_K63_NRGB)
- #define P_K65 0x1b17
- #define P_K65_STR "1b17"
- #define P_K65_NRGB 0x1b07
- #define P_K65_NRGB_STR "1b07"
- #define P_K65_LUX 0x1b37
- #define P_K65_LUX_STR "1b37"
- #define P_K65_RFIRE 0x1b39
- #define P_K65_RFIRE_STR "1b39"
- #define IS_K65(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K65 || (kb)->product == P_K65-_NRGB || (kb)->product == P_K65_LUX || (kb)->product == P_K65_RFIRE))
- #define P_K68 0x1b3f
- #define P_K68_STR "1b3f"
- #define IS_K68(kb) ((kb)->vendor == V_CORSAIR && (kb)->product == P_K68)
- #define P_K70 0x1b13
- #define P_K70_STR "1b13"
- #define P_K70_NRGB 0x1b09
- #define P_K70_NRGB_STR "1b09"
- #define P_K70_LUX 0x1b33
- #define P_K70_LUX_STR "1b33"
- #define P_K70_LUX_NRGB 0x1b36
- #define P_K70_LUX_NRGB_STR "1b36"
- #define P_K70_RFIRE 0x1b38
- #define P_K70_RFIRE_STR "1b38"
- #define P_K70_RFIRE_NRGB 0x1b3a

- #define P_K70_RFIRE_NRGB_STR "1b3a"
- #define IS_K70(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K70 || (kb)->product == P_K70_NRGB || (kb)->product == P_K70_RFIRE || (kb)->product == P_K70_RFIRE_NRGB || (kb)->product == P_K70_LUX || (kb)->product == P_K70_LUX_NRGB))
- #define P_K95 0x1b11
- #define P_K95_STR "1b11"
- #define P_K95_NRGB 0x1b08
- #define P_K95_NRGB_STR "1b08"
- #define P_K95_PLATINUM 0x1b2d
- #define P_K95_PLATINUM_STR "1b2d"
- #define IS_K95(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K95 || (kb)->product == P_K95_NRGB || (kb)->product == P_K95_PLATINUM))
- #define P_STRAFE 0x1b20
- #define P_STRAFE_STR "1b20"
- #define P_STRAFE_NRGB 0x1b15
- #define P_STRAFE_NRGB_STR "1b15"
- #define P_STRAFE_NRGB_2 0x1b44
- #define P_STRAFE_NRGB_2_STR "1b44"
- #define IS_STRAFE(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_STRAFE || (kb)->product == P_STRAFE_NRGB || (kb)->product == P_STRAFE_NRGB_2))
- #define P_M65 0x1b12
- #define P_M65_STR "1b12"
- #define P_M65_PRO 0x1b2e
- #define P_M65_PRO_STR "1b2e"
- #define IS_M65(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_M65 || (kb)->product == P_M65_PRO))
- #define P_SABRE_O 0x1b14 /* optical */
- #define P_SABRE_O_STR "1b14"
- #define P_SABRE_L 0x1b19 /* laser */
- #define P_SABRE_L_STR "1b19"
- #define P_SABRE_N 0x1b2f /* new? */
- #define P_SABRE_N_STR "1b2f"
- #define P_SABRE_O2 0x1b32 /* Observed on a CH-9000111-EU model SABRE */
- #define P_SABRE_O2_STR "1b32"
- #define IS_SABRE(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_SABRE_O || (kb)->product == P_SABRE_L || (kb)->product == P_SABRE_N || (kb)->product == P_SABRE_O2))
- #define P_SCIMITAR 0x1b1e
- #define P_SCIMITAR_STR "1b1e"
- #define P_SCIMITAR_PRO 0x1b3e
- #define P_SCIMITAR_PRO_STR "1b3e"
- #define IS_SCIMITAR(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_SCIMITAR || (kb)->product == P_SCIMITAR_PRO))
- #define P_HARPOON 0x1b3c
- #define P_HARPOON_STR "1b3c"
- #define IS_HARPOON(kb) ((kb)->vendor == V_CORSAIR && (kb)->product == P_HARPOON)
- #define P_GLAIVE 0x1b34
- #define P_GLAIVE_STR "1b34"
- #define IS_GLAIVE(kb) ((kb)->vendor == V_CORSAIR && (kb)->product == P_GLAIVE)
- #define IS_RGB(vendor, product) ((vendor) == (V_CORSAIR) && (product) != (P_K65_NRGB) && (product) != (P_K70_NRGB) && (product) != (P_K95_NRGB))

  RGB vs non-RGB test (note: non-RGB Strafe is still considered "RGB" in that it shares the same protocol. The difference is denoted with the "monochrome" feature).
- #define IS_MONOCHROME(vendor, product) ((vendor) == (V_CORSAIR) && ((product) == (P_K68) || (product) == (P_STRAFE_NRGB) || (product) == (P_STRAFE_NRGB_2)))

*The difference between non RGB and monochrome is, that monochrome has lights, but just in one color. nonRGB has no lights. Change this if new **monochrome** devices are added.*

- #define IS_RGB_DEV(kb) IS_RGB((kb)->vendor, (kb)->product)

  For calling with a usbdevice∗, vendor and product are extracted and IS_RGB() is returned.
- #define IS_MONOCHROME_DEV(kb) IS_MONOCHROME((kb)->vendor, (kb)->product)

  For calling with a usbdevice∗, vendor and product are extracted and IS_MONOCHROME() is returned.
- #define IS_FULLRANGE(kb) (IS_RGB((kb)->vendor, (kb)->product) && (kb)->product != P_K65 && (kb)->product != P_K70 && (kb)->product != P_K95)

  Full color range (16.8M) vs partial color range (512)
- #define IS_MOUSE(vendor, product) ((vendor) == (V_CORSAIR) && ((product) == (P_M65) || (product) == (P_M65_PRO) || (product) == (P_SABRE_O) || (product) == (P_SABRE_L) || (product) == (P_SABRE_N) || (product) == (P_SCIMITAR) || (product) == (P_SCIMITAR_PRO) || (product) == (P_SABRE_O2) || (product) == (P_GLAIVE) || (product) == (P_HARPOON)))

  Mouse vs keyboard test.
- #define IS_MOUSE_DEV(kb) IS_MOUSE((kb)->vendor, (kb)->product)

  For calling with a usbdevice∗, vendor and product are extracted and IS_MOUSE() is returned.
- #define IS_PLATINUM(kb) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K95_PLATINUM))

  Used to apply quirks and features to the PLATINUM devices.
- #define IS_V2_OVERRIDE(kb) (IS_PLATINUM(kb) || IS_K63(kb) || IS_K68(kb) || IS_HARPOON(kb) || IS_GLAIVE(kb) || (kb)->product == P_STRAFE_NRGB_2)

  Used when a device has a firmware with a low version number that uses the new protocol.
- #define DELAY_SHORT(kb) clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = ((int) (kb->usbdelay)) ∗ 1000000}, NULL)

  USB delays for when the keyboards get picky about timing That was the original comment, but it is used anytime.
- #define DELAY_MEDIUM(kb) clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = ((int) (kb->usbdelay)) ∗ 10000000}, NULL)

  the medium delay is used after sending a command before waiting for the answer.
- #define DELAY_LONG(kb) clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = 100000000}, NULL)

  The longest delay takes place where something went wrong (eg when resetting the device)
- #define USB_DELAY_DEFAULT 5

  *This constant is used to initialize **kb->usbdelay**. It is used in many places (see macros above) but often also overwritten to the fixed value of 10. Pure Hacker code.*
- #define resetusb(kb) _resetusb(kb, __FILE_NOPATH__, __LINE__)

  resetusb() is just a macro to call _resetusb() with debuggin constants (file, lineno)
- #define usbsend(kb, messages, count) _usbsend(kb, messages, count, __FILE_NOPATH__, __LINE__)

  usbsend macro is used to wrap _usbsend() with debugging information (file and lineno)
- #define usbrecv(kb, out_msg, in_msg) _usbrecv(kb, out_msg, in_msg, __FILE_NOPATH__, __LINE__)

  usbrecv macro is used to wrap _usbrecv() with debugging information (file and lineno)
- #define nk95cmd(kb, command) _nk95cmd(kb, (command) >> 16 & 0xFF, (command) & 0xFFFF, __FILE_NOPATH__, __LINE__)

  nk95cmd() macro is used to wrap _nk95cmd() with debugging information (file and lineno). the command structure is different:
  Just the bits 23..16 are used as bits 7..0 for bRequest
  Bits 15..0 are used as wValue
- #define NK95_HWOFF 0x020030

  Hardware-specific commands for the K95 nonRGB,.
- #define NK95_HWON 0x020001

  Hardware playback on.
- #define NK95_M1 0x140001

  Switch to mode 1.
- #define NK95_M2 0x140002

  Switch to mode 2.
- #define NK95_M3 0x140003

  Switch to mode 3.

Functions

- const char * vendor_str (short vendor)

  uncomment to see USB packets sent to the device
- const char * product_str (short product)

  product_str returns a condensed view on what type of device we have.
- int usbmain ()

  Start the USB main loop. Returns program exit code when finished.
- void usbkill ()

  Stop the USB system.
- void setupusb (usbdevice *kb)

  setupusb starts a thread with kb as parameter and _setupusb() as entrypoint.
- int os_setupusb (usbdevice *kb)

  os_setupusb OS-specific setup for a specific usb device.
- void * os_inputmain (void *context)

  os_inputmain is run in a separate thread and will be detached from the main thread, so it needs to clean up its own resources.
- int revertusb (usbdevice *kb)

  revertusb sets a given device to inactive (hardware controlled) mode if not a fw-ugrade is indicated
- int closeusb (usbdevice *kb)

  closeusb Close a USB device and remove device entry.
- void os_closeusb (usbdevice *kb)

  os_closeusb unclaim it, destroy the udev device and clear data structures at kb
- int _resetusb (usbdevice *kb, const char *file, int line)

  _resetusb Reset a USB device.
- int os_resetusb (usbdevice *kb, const char *file, int line)

  os_resetusb is the os specific implementation for resetting usb
- int _usbsend (usbdevice *kb, const uchar *messages, int count, const char *file, int line)

  _usbsend send a logical message completely to the given device
- int _usbrecv (usbdevice *kb, const uchar *out_msg, uchar *in_msg, const char *file, int line)

  _usbrecv Request data from a USB device by first sending an output packet and then reading the response.
- int os_usbsend (usbdevice *kb, const uchar *out_msg, int is_recv, const char *file, int line)

  os_usbsend sends a data packet (MSG_SIZE = 64) Bytes long
- int os_usbrecv (usbdevice *kb, uchar *in_msg, const char *file, int line)

  os_usbrecv receives a max MSGSIZE long buffer from usb device
- void os_sendindicators (usbdevice *kb)

  os_sendindicators update the indicators for the special keys (Numlock, Capslock and what else?)
- int _nk95cmd (usbdevice *kb, uchar bRequest, ushort wValue, const char *file, int line)

  _nk95cmd If we control a non RGB keyboard, set the keyboard via ioctl with usbdevfs_ctrltransfer
- int usb_tryreset (usbdevice *kb)

  usb_tryreset does what the name means: Try to reset the usb via resetusb()

13.49.1 Detailed Description

Vendor/product codes

The list of defines in the first part of the file describes the various types of equipment from Corsair and summarizes them according to specific characteristics.

Each device type is described with two defines:

- On the one hand the device ID with which the device can be recognized on the USB as a short

- and on the other hand the same representation as a string, but without leading "0x".

First entry-pair is the Provider ID (vendorID) from Corsair.

Block No. | contains | Devices are bundled via --------— | -------— | ----------------------— 1 | The first block contains the K63 Non RGB Keyboard. No other K63 is known so far. 2 | the K65-like keyboards, regardless of their properties (RGB, ...). | In summary, they can be queried using the macro IS_K65(). 3 | K68 keyboard | IS_K68(). 4 | the K70-like Keyboards with all their configuration types | summarized by IS_K70(). 5 | the K95 series keyboards | collected with the macro IS_K95(). 6 | strafe keyboards | IS_STRAFE() 7 | M65 mice with and without RGB | IS_M65() 8 | Sabre mice | IS_SABRE() 9 | Scimitar mice | IS_SCIMITAR() 10| Harpoon mice | IS_HARPOON() 11| Glaive mice | IS_GLAIVE()

Definition in file usb.h.

13.49.2 Macro Definition Documentation

13.49.2.1 #define DELAY_LONG(*kb*) clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = 100000000}, NULL)

Definition at line 186 of file usb.h.

Referenced by _resetusb(), _setupusb(), _usbrecv(), _usbsend(), cmd_hwload_kb(), cmd_hwload_mouse(), cmd_hwsave_kb(), and cmd_hwsave_mouse().

13.49.2.2 #define DELAY_MEDIUM(*kb*) clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = ((int)(kb->usbdelay)) * 10000000}, NULL)

Definition at line 182 of file usb.h.

Referenced by _usbrecv(), and setactive_kb().

13.49.2.3 #define DELAY_SHORT(*kb*) clock_nanosleep(CLOCK_MONOTONIC, 0, &(struct timespec) {.tv_nsec = ((int)(kb->usbdelay)) * 1000000}, NULL)

The short delay is used before any send or receive

Definition at line 178 of file usb.h.

Referenced by _usbrecv(), _usbsend(), and updateindicators_kb().

13.49.2.4 #define IS_FULLRANGE(*kb*) (IS_RGB((kb)->vendor, (kb)->product) && (kb)->product != P_K65 && (kb)->product != P_K70 && (kb)->product != P_K95)

Definition at line 160 of file usb.h.

Referenced by readcmd(), and updatergb_kb().

13.49.2.5 #define IS_GLAIVE(*kb*) ((kb)->vendor == V_CORSAIR && (kb)->product == P_GLAIVE)

Definition at line 121 of file usb.h.

Referenced by updatergb_mouse().

13.49.2.6 #define IS_HARPOON(*kb*) ((kb)->vendor == V_CORSAIR && (kb)->product == P_HARPOON)

Definition at line 117 of file usb.h.

13.49.2.7 #define IS_K63(*kb*) ((kb)->vendor == V_CORSAIR && (kb)->product == P_K63_NRGB)

Definition at line 47 of file usb.h.

Referenced by has_key().

13.49.2.8 #define IS_K65(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K65 || (kb)->product == P_K65_NRGB || (kb)->product == P_K65_LUX || (kb)->product == P_K65_RFIRE))

Definition at line 57 of file usb.h.

Referenced by has_key().

13.49.2.9 #define IS_K68(*kb*) ((kb)->vendor == V_CORSAIR && (kb)->product == P_K68)

Definition at line 61 of file usb.h.

13.49.2.10 #define IS_K70(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K70 || (kb)->product == P_K70_NRGB || (kb)->product == P_K70_RFIRE || (kb)->product == P_K70_RFIRE_NRGB || (kb)->product == P_K70_LUX || (kb)->product == P_K70_LUX_NRGB))

Definition at line 75 of file usb.h.

13.49.2.11 #define IS_K95(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K95 || (kb)->product == P_K95_NRGB || (kb)->product == P_K95_PLATINUM))

Definition at line 83 of file usb.h.

Referenced by cmd_hwload_kb(), cmd_hwsave_kb(), and has_key().

13.49.2.12 #define IS_M65(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_M65 || (kb)->product == P_M65_PRO))

Definition at line 97 of file usb.h.

Referenced by isblack().

13.49.2.13 #define IS_MONOCHROME(*vendor, product*) ((vendor) == (V_CORSAIR) && ((product) == (P_K68) || (product) == (P_STRAFE_NRGB) || (product) == (P_STRAFE_NRGB_2)))

Definition at line 151 of file usb.h.

Referenced by _setupusb().

13.49.2.14 #define IS_MONOCHROME_DEV(*kb*) IS_MONOCHROME((kb)->vendor, (kb)->product)

Definition at line 157 of file usb.h.

13.49.2.15 #define IS_MOUSE(*vendor, product*) ((vendor) == (V_CORSAIR) && ((product) == (P_M65) || (product) == (P_M65_PRO) || (product) == (P_SABRE_O) || (product) == (P_SABRE_L) || (product) == (P_SABRE_N) || (product) == (P_SCIMITAR) || (product) == (P_SCIMITAR_PRO) || (product) == (P_SABRE_O2) || (product) == (P_GLAIVE) || (product) == (P_HARPOON)))

Definition at line 163 of file usb.h.

Referenced by _setupusb(), get_vtable(), has_key(), and os_inputmain().

13.49.2.16 #define IS_MOUSE_DEV(*kb*) IS_MOUSE((kb)->vendor, (kb)->product)

Definition at line 166 of file usb.h.

Referenced by readcmd().

13.49.2.17 #define IS_PLATINUM(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_K95_PLATINUM))

Definition at line 169 of file usb.h.

13.49.2.18 #define IS_RGB(*vendor, product*) ((vendor) == (V_CORSAIR) && (product) != (P_K65_NRGB) && (product) != (P_K70_NRGB) && (product) != (P_K95_NRGB))

Definition at line 146 of file usb.h.

Referenced by _setupusb(), get_vtable(), and os_inputmain().

13.49.2.19 #define IS_RGB_DEV(*kb*) IS_RGB((kb)->vendor, (kb)->product)

Definition at line 154 of file usb.h.

13.49.2.20 #define IS_SABRE(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_SABRE_O || (kb)->product == P_SABRE_L || (kb)->product == P_SABRE_N || (kb)->product == P_SABRE_O2))

Definition at line 107 of file usb.h.

Referenced by has_key(), loadrgb_mouse(), and savergb_mouse().

13.49.2.21 #define IS_SCIMITAR(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_SCIMITAR || (kb)->product == P_SCIMITAR_PRO))

Definition at line 113 of file usb.h.

Referenced by has_key(), loadrgb_mouse(), and savergb_mouse().

13.49.2.22 #define IS_STRAFE(*kb*) ((kb)->vendor == V_CORSAIR && ((kb)->product == P_STRAFE || (kb)->product == P_STRAFE_NRGB || (kb)->product == P_STRAFE_NRGB_2))

Definition at line 91 of file usb.h.

Referenced by savergb_kb().

13.49.2.23 #define IS_V2_OVERRIDE(*kb*) (IS_PLATINUM(kb) || IS_K63(kb) || IS_K68(kb) || IS_HARPOON(kb) || IS_GLAIVE(kb) || (kb)->product == P_STRAFE_NRGB_2)

Definition at line 172 of file usb.h.

Referenced by loadrgb_kb(), os_usbsend(), and savergb_kb().

13.49.2.24 #define NK95_HWOFF 0x020030

See Also

usb2.0 documentation for details. Set Hardware playback off

Definition at line 333 of file usb.h.

Referenced by start_kb_nrgb().

13.49.2.25 #define NK95_HWON 0x020001

Definition at line 336 of file usb.h.

Referenced by revertusb().

13.49.2.26 #define NK95_M1 0x140001

Definition at line 339 of file usb.h.

Referenced by setmodeindex_nrgb().

13.49.2.27 #define NK95_M2 0x140002

Definition at line 342 of file usb.h.

Referenced by setmodeindex_nrgb().

13.49.2.28 #define NK95_M3 0x140003

Definition at line 345 of file usb.h.

Referenced by setmodeindex_nrgb().

13.49.2.29 #define nk95cmd(*kb, command*) _nk95cmd(kb, (command) >> 16 & 0xFF, (command) & 0xFFFF, __FILE_NOPATH__, __LINE__)

Definition at line 328 of file usb.h.

Referenced by revertusb(), setmodeindex_nrgb(), and start_kb_nrgb().

13.49.2.30 #define P_GLAIVE 0x1b34

Definition at line 119 of file usb.h.

Referenced by product_str().

13.49.2.31 #define P_GLAIVE_STR "1b34"

Definition at line 120 of file usb.h.

13.49.2.32 #define P_HARPOON 0x1b3c

Definition at line 115 of file usb.h.

Referenced by product_str().

13.49.2.33 #define P_HARPOON_STR "1b3c"

Definition at line 116 of file usb.h.

13.49.2.34 #define P_K63_NRGB 0x1b40

Definition at line 45 of file usb.h.

Referenced by product_str().

13.49.2.35 #define P_K63_NRGB_STR "1b40"

Definition at line 46 of file usb.h.

13.49.2.36 #define P_K65 0x1b17

Definition at line 49 of file usb.h.

Referenced by product_str().

13.49.2.37 #define P_K65_LUX 0x1b37

Definition at line 53 of file usb.h.

Referenced by product_str().

13.49.2.38 #define P_K65_LUX_STR "1b37"

Definition at line 54 of file usb.h.

13.49.2.39 #define P_K65_NRGB 0x1b07

Definition at line 51 of file usb.h.

Referenced by product_str().

13.49.2.40 #define P_K65_NRGB_STR "1b07"

Definition at line 52 of file usb.h.

13.49.2.41 #define P_K65_RFIRE 0x1b39

Definition at line 55 of file usb.h.

Referenced by product_str().

13.49.2.42 #define P_K65_RFIRE_STR "1b39"

Definition at line 56 of file usb.h.

13.49.2.43 #define P_K65_STR "1b17"

Definition at line 50 of file usb.h.

13.49.2.44 #define P_K68 0x1b3f

Definition at line 59 of file usb.h.

Referenced by product_str().

13.49.2.45 #define P_K68_STR "1b3f"

Definition at line 60 of file usb.h.

13.49.2.46 #define P_K70 0x1b13

Definition at line 63 of file usb.h.

Referenced by product_str().

13.49.2.47 #define P_K70_LUX 0x1b33

Definition at line 67 of file usb.h.

Referenced by loadrgb_kb(), and product_str().

13.49.2.48 #define P_K70_LUX_NRGB 0x1b36

Definition at line 69 of file usb.h.

Referenced by loadrgb_kb(), and product_str().

13.49.2.49 #define P_K70_LUX_NRGB_STR "1b36"

Definition at line 70 of file usb.h.

13.49.2.50 #define P_K70_LUX_STR "1b33"

Definition at line 68 of file usb.h.

13.49.2.51 #define P_K70_NRGB 0x1b09

Definition at line 65 of file usb.h.

Referenced by product_str().

13.49.2.52 #define P_K70_NRGB_STR "1b09"

Definition at line 66 of file usb.h.

13.49.2.53 #define P_K70_RFIRE 0x1b38

Definition at line 71 of file usb.h.

Referenced by product_str().

13.49.2.54 #define P_K70_RFIRE_NRGB 0x1b3a

Definition at line 73 of file usb.h.

Referenced by product_str().

13.49.2.55 #define P_K70_RFIRE_NRGB_STR "1b3a"

Definition at line 74 of file usb.h.

13.49.2.56 #define P_K70_RFIRE_STR "1b38"

Definition at line 72 of file usb.h.

13.49.2.57 #define P_K70_STR "1b13"

Definition at line 64 of file usb.h.

13.49.2.58 #define P_K95 0x1b11

Definition at line 77 of file usb.h.

Referenced by product_str().

13.49.2.59 #define P_K95_NRGB 0x1b08

Definition at line 79 of file usb.h.

Referenced by _nk95cmd(), and product_str().

13.49.2.60 #define P_K95_NRGB_STR "1b08"

Definition at line 80 of file usb.h.

13.49.2.61 #define P_K95_PLATINUM 0x1b2d

Definition at line 81 of file usb.h.

Referenced by product_str(), and updatergb_kb().

13.49.2.62 #define P_K95_PLATINUM_STR "1b2d"

Definition at line 82 of file usb.h.

13.49.2.63 #define P_K95_STR "1b11"

Definition at line 78 of file usb.h.

13.49.2.64 #define P_M65 0x1b12

Definition at line 93 of file usb.h.

Referenced by product_str().

13.49.2.65 #define P_M65_PRO 0x1b2e

Definition at line 95 of file usb.h.

Referenced by product_str().

13.49.2.66 #define P_M65_PRO_STR "1b2e"

Definition at line 96 of file usb.h.

13.49.2.67 #define P_M65_STR "1b12"

Definition at line 94 of file usb.h.

13.49.2.68 #define P_SABRE_L 0x1b19 /* laser */

Definition at line 101 of file usb.h.

Referenced by product_str().

13.49.2.69 #define P_SABRE_L_STR "1b19"

Definition at line 102 of file usb.h.

13.49.2.70 #define P_SABRE_N 0x1b2f /* new? */

Definition at line 103 of file usb.h.

Referenced by product_str().

13.49.2.71 #define P_SABRE_N_STR "1b2f"

Definition at line 104 of file usb.h.

13.49.2.72 #define P_SABRE_O 0x1b14 /* optical */

Definition at line 99 of file usb.h.

Referenced by product_str().

13.49.2.73 #define P_SABRE_O2 0x1b32 /* Observed on a CH-9000111-EU model SABRE */

Definition at line 105 of file usb.h.

Referenced by product_str().

13.49.2.74 #define P_SABRE_O2_STR "1b32"

Definition at line 106 of file usb.h.

13.49.2.75 #define P_SABRE_O_STR "1b14"

Definition at line 100 of file usb.h.

13.49.2.76 #define P_SCIMITAR 0x1b1e

Definition at line 109 of file usb.h.

Referenced by product_str().

13.49.2.77 #define P_SCIMITAR_PRO 0x1b3e

Definition at line 111 of file usb.h.

Referenced by product_str().

13.49.2.78 #define P_SCIMITAR_PRO_STR "1b3e"

Definition at line 112 of file usb.h.

13.49.2.79 #define P_SCIMITAR_STR "1b1e"

Definition at line 110 of file usb.h.

13.49.2.80 #define P_STRAFE 0x1b20

Definition at line 85 of file usb.h.

Referenced by product_str().

13.49.2.81 #define P_STRAFE_NRGB 0x1b15

Definition at line 87 of file usb.h.

Referenced by product_str().

13.49.2.82 #define P_STRAFE_NRGB_2 0x1b44

Definition at line 89 of file usb.h.

Referenced by product_str().

13.49.2.83 #define P_STRAFE_NRGB_2_STR "1b44"

Definition at line 90 of file usb.h.

13.49.2.84 #define P_STRAFE_NRGB_STR "1b15"

Definition at line 88 of file usb.h.

13.49.2.85 #define P_STRAFE_STR "1b20"

Definition at line 86 of file usb.h.

13.49.2.86 #define resetusb(*kb*) _resetusb(kb, __FILE_NOPATH__, __LINE__)

Definition at line 246 of file usb.h.

Referenced by usb_tryreset().

13.49.2.87 #define USB_DELAY_DEFAULT 5

Definition at line 192 of file usb.h.

Referenced by _setupusb(), and start_dev().

13.49.2.88 #define usbrecv(*kb, out_msg, in_msg*) _usbrecv(kb, out_msg, in_msg, __FILE_NOPATH__, __LINE__)

Parameters

kb	THE usbdevice*
IN]	out_msg What information does the caller want from the device?
OUT]	in_msg Here comes the answer; The names represent the usb view, not the view of this function! So INput from usb is OUTput of this function.

Definition at line 288 of file usb.h.

Referenced by cmd_hwload_kb(), cmd_hwload_mouse(), getfwversion(), hwloadmode(), loaddpi(), loadrgb_kb(), and loadrgb_mouse().

13.49.2.89 #define usbsend(*kb, messages, count*) _usbsend(kb, messages, count, __FILE_NOPATH__, __LINE__)

Parameters

kb	THE usbdevice*
IN]	messages a Pointer to the first byte of the logical message
IN]	count how many MSG_SIZE buffers is the logical message long?

Definition at line 271 of file usb.h.

Referenced by cmd_hwsave_kb(), cmd_hwsave_mouse(), cmd_pollrate(), fwupdate(), loadrgb_kb(), savedpi(), savergb_kb(), savergb_mouse(), setactive_kb(), setactive_mouse(), updatedpi(), updatergb_kb(), and updatergb_mouse().

13.49.2.90 #define V_CORSAIR 0x1b1c

Warning

> When adding new devices please update src/ckb/fwupgradedialog.cpp as well.
> It should contain the same vendor/product IDs for any devices supporting firmware updates.
> In the same way, all other corresponding files have to be supplemented or modified: Currently known for this are **usb_linux.c** and **usb_mac.c**

Definition at line 42 of file usb.h.

Referenced by usb_add_device(), and vendor_str().

13.49.2.91 #define V_CORSAIR_STR "1b1c"

Definition at line 43 of file usb.h.

Referenced by udev_enum(), and usb_add_device().

13.49.3 Function Documentation

13.49.3.1 int _nk95cmd (usbdevice * kb, uchar bRequest, ushort wValue, const char * file, int line)

Parameters

kb	THE usbdevice*
bRequest	the byte array with the usb request
wValue	a usb wValue
file	for error message
line	for error message

Returns

1 (true) on failure, 0 (false) on success.

To send control packets to a non RGB non color K95 Keyboard, use this function. Normally it is called via the nk95cmd() macro.

If it is the wrong device for which the function is called, 0 is returned and nothing done. Otherwise a usbdevfs_ctrltransfer structure is filled and an USBDEVFS_CONTROL ioctl() called.

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0x40	see table below to switch hardware-modus at Keyboard	wValue	device	MSG_SIZE	5ms	the message buffer pointer
Host to Device, Type=Vendor, Recipient=Device	bRequest parameter	given wValue Parameter	device 0	0 data to write	5000	null

If a 0 or a negative error number is returned by the ioctl, an error message is shown depending on the errno or "No data written" if retval was 0. In either case 1 is returned to indicate the error. If the ioctl returned a value $>$ 0, 0 is returned to indicate no error.

Currently the following combinations for bRequest and wValue are used:

Device	what it might to do	constant	bRequest	wValue
non RGB Keyboard	set HW-modus on (leave the ckb driver)	HWON	0x0002	0x0030
non RGB Keyboard	set HW-modus off (initialize the ckb driver)	HWOFF	0x0002	0x0001
non RGB Keyboard	set light modus M1 in single-color keyboards	NK95_M1	0x0014	0x0001
non RGB Keyboard	set light modus M2 in single-color keyboards	NK95_M2	0x0014	0x0002

non RGB Keyboard	set light modus M3 in single-color keyboards	NK95_M3	0x0014	0x0003

See Also

usb.h

Definition at line 189 of file usb_linux.c.

References ckb_err_fn, usbdevice::handle, P_K95_NRGB, and usbdevice::product.

```
189                                                                                 {
190     if(kb->product != P_K95_NRGB)
191         return 0;
192     struct usbdevfs_ctrltransfer transfer = { 0x40, bRequest, wValue, 0, 0, 5000, 0 };
193     int res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
194     if(res <= 0){
195         ckb_err_fn("%s\n", file, line, res ? strerror(errno) : "No data written");
196         return 1;
197     }
198     return 0;
199 }
```

13.49.3.2 int _resetusb (usbdevice * kb, const char * file, int line)

Parameters

kb	THE usbdevice*
file	filename for error messages
line	line where it is called for error messages

Returns

Returns 0 on success, -1 if device should be removed

_resetusb Reset a USB device.

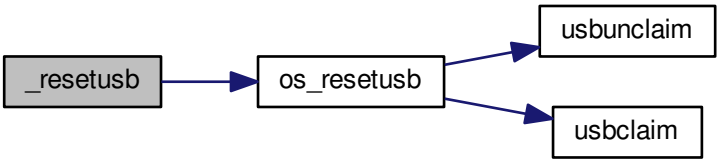
First reset the device via os_resetusb() after a long delay (it may send something to the host). If this worked (retval == 0), give the device another long delay Then perform the initialization via the device specific start() function entry in kb->vtable and if this is successful also, return the result of the device depenten updatergb() with force=true.

Definition at line 436 of file usb.c.

References usbdevice::active, DELAY_LONG, os_resetusb(), and usbdevice::vtable.

```
436                                                 {
437     // Perform a USB reset
438     DELAY_LONG(kb);
439     int res = os_resetusb(kb, file, line);
440     if(res)
441         return res;
442     DELAY_LONG(kb);
443     // Re-initialize the device
444     if(kb->vtable->start(kb, kb->active) != 0)
445         return -1;
446     if(kb->vtable->updatergb(kb, 1) != 0)
447         return -1;
448     return 0;
449 }
```

Here is the call graph for this function:



13.49.3.3 int _usbrecv (usbdevice * *kb,* const uchar * *out_msg,* uchar * *in_msg,* const char * *file,* int *line*)

Parameters

kb	THE usbdevice*
IN]	out_msg What information does the caller want from the device?
OUT]	in_msg Here comes the answer; The names represent the usb view, not the view of this function! So INput from usb is OUTput of this function.
IN]	file for debugging
IN]	line for debugging
IN]	reset_stop global variable is read

Returns

number of bytes read or zero on failure.

_usbrecv Request data from a USB device by first sending an output packet and then reading the response.

To fully understand this, you need to know about usb: All control is at the usb host (the CPU). If the device wants to communicate something to the host, it must wait for the host to ask. The usb protocol defines the cycles and periods in which actions are to be taken.

So in order to receive a data packet from the device, the host must first send a send request.

This is done by _usbrecv() in the first block by sending the MSG_SIZE large data block from **out_msg** via os-_usbsend() as it is a machine depending implementation. The usb target device is as always determined over kb.

For os_usbsend() to know that it is a receive request, the **is_recv** parameter is set to true (1). With this, os_usbsend () generates a control package for the hardware, not a data packet.

If sending of the control package is not successful, a maximum of 5 times the transmission is repeated (including the first attempt). If a non-cancelable error is signaled or the drive is stopped via reset_stop, _usbrecv() immediately returns 0.

After this, the function waits for the requested response from the device using os_usbrecv ().

os_usbrecv() returns 0, -1 or something else.

Zero signals a serious error which is not treatable and _usbrecv() also returns 0.

-1 means that it is a treatable error - a timeout for example - and therefore the next transfer attempt is started after a long pause (DELAY_LONG) if not reset_stop or the wrong hwload_mode require a termination with a return value of 0.

After 5 attempts, _usbrecv () returns and returns 0 as well as an error message.

When data is received, the number of received bytes is returned. This should always be MSG_SIZE, but os_usbrecv() can also return less. It should not be more, because then there would be an unhandled buffer overflow, but it could be less. This would be signaled in os_usbrecv () with a message.

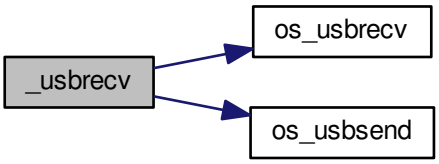
The buffers behind **out_msg** and **in_msg** are MSG_SIZE at least (currently 64 Bytes). More is ok but useless, less brings unpredictable behavior. < Synchonization between macro and color information

Definition at line 611 of file usb.c.

References ckb_err_fn, DELAY_LONG, DELAY_MEDIUM, DELAY_SHORT, hwload_mode, mmutex, os_usbrecv(), os_usbsend(), and reset_stop.

```
611                                                                                                    {
612     // Try a maximum of 5 times
613     for (int try = 0; try < 5; try++) {
614         // Send the output message
615         pthread_mutex_lock(mmutex(kb));
616         DELAY_SHORT(kb);
617         int res = os_usbsend(kb, out_msg, 1, file, line);
618         pthread_mutex_unlock(mmutex(kb));
619         if (res == 0)
620             return 0;
621         else if (res == -1) {
622             // Retry on temporary failure
623             if (reset_stop)
624                 return 0;
625             DELAY_LONG(kb);
626             continue;
627         }
628         // Wait for the response
629         DELAY_MEDIUM(kb);
630         res = os_usbrecv(kb, in_msg, file, line);
631         if(res == 0)
632             return 0;
633         else if(res != -1)
634             return res;
635         if(reset_stop || hwload_mode != 2)
636             return 0;
637         DELAY_LONG(kb);
638     }
639     // Give up
640     ckb_err_fn("Too many send/recv failures. Dropping.\n", file, line);
641     return 0;
642 }
```

Here is the call graph for this function:



13.49.3.4 int _usbsend (usbdevice * *kb*, const uchar * *messages*, int *count*, const char * *file*, int *line*)

Parameters

	kb	THE usbdevice∗
	IN]	messages a Pointer to the first byte of the logical message
	IN]	count how many MSG_SIZE buffers is the logical message long?
	IN]	file for debugging
	IN]	line for debugging
in	*reset_stop*	global variable is read

Returns

number of Bytes sent (ideal == count ∗ MSG_SIZE);
0 if a block could not be sent and it was not a timeout OR **reset_stop** was required or **hwload_mode** is not set to "always"

_usbsend send a logical message completely to the given device

Todo A lot of different conditions are combined in this code. Don't think, it is good in every combination...

The main task of _usbsend () is to transfer the complete logical message from the buffer beginning with *messages* to **count** ∗ **MSG_SIZE**.

According to usb 2.0 specification, a USB transmits a maximum of 64 byte user data packets. For the transmission of longer messages we need a segmentation. And that is exactly what happens here.

The message is given one by one to os_usbsend() in MSG_SIZE (= 64) byte large bites.

Attention

This means that the buffer given as argument must be n ∗ MSG_SIZE Byte long.

An essential constant parameter which is relevant for os_usbsend() only is is_recv = 0, which means sending.

Now it gets a little complicated again:

- If os_usbsend() returns 0, only zero bytes could be sent in one of the packets, or it was an error (-1 from the systemcall), but not a timeout. How many Bytes were sent in total from earlier calls does not seem to matter, _usbsend() returns a total of 0.
- Returns os_usbsend() -1, first check if **reset_stop** is set globally or (incomprehensible) hwload_mode is not set to "always". In either case, _usbsend() returns 0, otherwise it is assumed to be a temporary transfer error and it simply retransmits the physical packet after a long delay.
- If the return value of os_usbsend() was neither 0 nor -1, it specifies the numer of bytes transferred.

  Here is an information hiding conflict with os_usbsend() (at least in the Linux version):

  If os_usbsend() can not transfer the entire packet, errors are thrown and the number of bytes sent is returned. _usbsend() interprets this as well and remembers the total number of bytes transferred in the local variable **total_sent**. Subsequently, however, transmission is continued with the next complete MSG_SIZE block and not with the first of the possibly missing bytes.

  Todo Check whether this is the same in the macOS variant. It is not dramatic, but if errors occur, it can certainly irritate the devices completely if they receive incomplete data streams. Do we have errors with the messages "Wrote YY bytes (expected 64)" in the system logs? If not, we do not need to look any further.

When the last packet is transferred, _usbsend() returns the effectively counted set of bytes (from **total_sent**). This at least gives the caller the opportunity to check whether something has been lost in the middle.

A bit strange is the structure of the program: Handling the **count** MSG_SIZE blocks to be transferred is done in the outer for (...) loop. Repeating the transfer with a treatable error is managed by the inner while(1) loop.

This must be considered when reading the code; The "break" on successful block transfer leaves the inner while, not the for (...). < Synchonization between macro and color information

Definition at line 542 of file usb.c.

References DELAY_LONG, DELAY_SHORT, hwload_mode, mmutex, MSG_SIZE, os_usbsend(), and reset_stop.

```
542                                                                                                 {
543     int total_sent = 0;
544     for(int i = 0; i < count; i++){
545         // Send each message via the OS function
546         while(1){
547             pthread_mutex_lock(mmutex(kb));
548             DELAY_SHORT(kb);
549             int res = os_usbsend(kb, messages + i * MSG_SIZE, 0, file, line);
550             pthread_mutex_unlock(mmutex(kb));
551             if(res == 0)
552                 return 0;
553             else if(res != -1){
554                 total_sent += res;
555                 break;
556             }
557             // Stop immediately if the program is shutting down or hardware load is set to tryonce
558             if(reset_stop || hwload_mode != 2)
559                 return 0;
560             // Retry as long as the result is temporary failure
561             DELAY_LONG(kb);
562         }
563     }
564     return total_sent;
565 }
```

Here is the call graph for this function:

13.49.3.5 int closeusb (usbdevice * *kb*)

Parameters

IN,OUT]	kb

Returns

Returns 0 (everytime. No error handling is done!)

closeusb Close a USB device and remove device entry.

An imutex lock ensures first of all, that no communication is currently running from the viewpoint of the driver to the user input device (ie the virtual driver with which characters or mouse movements are sent from the daemon to the operating system as inputs).

If the **kb** has an acceptable value != 0, the index of the device is looked for and with this index os_inputclose() is called. After this no more characters can be sent to the operating system.

Then the connection to the usb device is capped by os_closeusb().

Todo What is not yet comprehensible is the call to updateconnected() BEFORE os_closeusb(). Should that be in the other sequence? Or is updateconnected() not displaying the connected usb devices, but the representation which uinput devices are loaded? Questions about questions ...

If there is no valid **handle**, only updateconnected() is called. We are probably trying to disconnect a connection under construction. Not clear.

The cmd pipe as well as all open notify pipes are deleted via rmdevpath ().

This means that nothing can happen to the input path - so the device-specific imutex is unlocked again and remains unlocked.

Also the dmutex is unlocked now, but only to join the thread, which was originally taken under **kb->thread** (which started with _setupusb()) with pthread_join() again. Because of the closed devices that thread would have to quit sometime

See Also

the hack note with rmdevpath())

As soon as the thread is caught, the dmutex is locked again, which is what I do not understand yet: What other thread can do usb communication now?

If the vtabel exists for the given kb (why not? It seems to have race conditions here!!), via the vtable the actually device-specific, but still everywhere identical freeprofile() is called. This frees areas that are no longer needed. Then the **usbdevice** structure in its array is set to zero completely.

Error handling is rather unusual in closeusb(); Everything works (no matter what the called functions return), and closeusb() always returns zero (success).

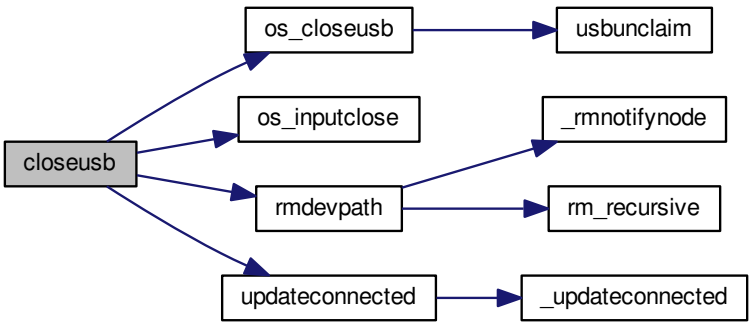
Definition at line 687 of file usb.c.

References ckb_info, devpath, dmutex, usbdevice::handle, imutex, INDEX_OF, keyboard, os_closeusb(), os_inputclose(), rmdevpath(), usbdevice::thread, updateconnected(), and usbdevice::vtable.

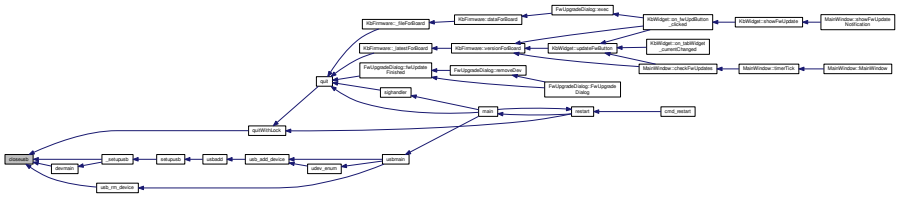
Referenced by _setupusb(), devmain(), quitWithLock(), and usb_rm_device().

```
687                             {
688     pthread_mutex_lock(imutex(kb));
689     if(kb->handle){
690         int index = INDEX_OF(kb, keyboard);
691         ckb_info("Disconnecting %s%d\n", devpath, index);
692         os_inputclose(kb);
693         updateconnected();
694         // Close USB device
695         os_closeusb(kb);
696     } else
697         updateconnected();
698     rmdevpath(kb);
699 
700     // Wait for thread to close
701     pthread_mutex_unlock(imutex(kb));
702     pthread_mutex_unlock(dmutex(kb));
703     pthread_join(kb->thread, 0);
704     pthread_mutex_lock(dmutex(kb));
705 
706     // Delete the profile and the control path
707     if(!kb->vtable)
708         return 0;
709     kb->vtable->freeprofile(kb);
710     memset(kb, 0, sizeof(usbdevice));
711     return 0;
712 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.49.3.6 void os_closeusb (usbdevice * *kb*)

Parameters

IN,OUT]	kb THE usbdevice*

os_closeusb unclaim it, destroy the udev device and clear data structures at kb

os_closeusb is the linux specific implementation for closing an active usb port.

If a valid handle is given in the kb structure, the usb port is unclaimed (usbunclaim()).

The device in unrefenced via library function udev_device_unref().

handle, udev and the first char of kbsyspath are cleared to 0 (empty string for kbsyspath).

Definition at line 448 of file usb_linux.c.

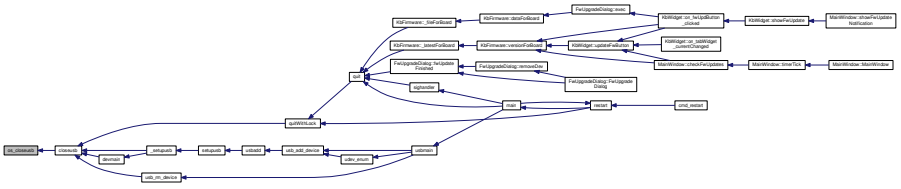
References usbdevice::handle, INDEX_OF, kbsyspath, keyboard, usbdevice::udev, and usbunclaim().

Referenced by closeusb().

```
448                                {
449     if(kb->handle){
450         usbunclaim(kb, 0);
451         close(kb->handle - 1);
452     }
453     if(kb->udev)
454         udev_device_unref(kb->udev);
455     kb->handle = 0;
456     kb->udev = 0;
457     kbsyspath[INDEX_OF(kb, keyboard)][0] = 0;
458 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.49.3.7 void* os_inputmain (void * *context*)

Parameters

context	THE usbdevice* ; Because os_inputmain() is started as a new thread, its formal parameter is named "context".

Returns

null

os_inputmain is run in a separate thread and will be detached from the main thread, so it needs to clean up its own resources.

Todo This function is a collection of many tasks. It should be divided into several sub-functions for the sake of greater convenience:

1. set up an URB (Userspace Ressource Buffer) to communicate with the USBDEVFS_* ioctl()s
2. perform the ioctl()
3. interpretate the information got into the URB buffer or handle error situations and retry operation or leave the endless loop
4. inform the os about the data
5. loop endless via 2.
6. if endless loop has gone, deinitalize the interface, free buffers etc.
7. return null

Here the actions in detail:

Monitor input transfers on all endpoints for non-RGB devices For RGB, monitor all but the last, as it's used for input/output

Get an usbdevfs_urb data structure and clear it via memset()

Hopefully the buffer lengths are equal for all devices with congruent types. You can find out the correctness for your device with lsusb –v or similar on macOS. Currently the following combinations are known and implemented:

device	detect with macro combination	endpoint #	buffer-length
each	none	0	8, 64 for FW v3
RGB Mouse	IS_RGB && IS_MOUSE	1	10
RGB Keyboard	IS_RGB && !IS_MOUSE	1	21
RGB Mouse or Keyboard	IS_RGB	2	MSG_SIZE (64)
non RGB Mouse or Keyboard	!IS_RGB	1	4
non RGB Mouse or Keyboard	!IS_RGB	2	15

Now submit all the URBs via ioctl(USBDEVFS_SUBMITURB) with type USBDEVFS_URB_TYPE_INTERRUPT (the endpoints are defined as type interrupt). Endpoint number is 0x80..0x82 or 0x83, depending on the model.

The userSpaceFS knows the URBs now, so start monitoring input

if the ioctl returns something != 0, let's have a deeper look what happened. Broken devices or shutting down the entire system leads to closing the device and finishing this thread.

If just an EPIPE ocurred, give the device a CLEAR_HALT and resubmit the URB.

A correct REAPURB returns a Pointer to the URB which we now have a closer look into. Lock all following actions with imutex.

Process the input depending on type of device. Interprete the actual size of the URB buffer

device	detect with macro combination	seems to be endpoint #	actual buffer-length	function called
mouse (RGB and non RGB)	IS_MOUSE	nA	8, 10 or 11	hid_mouse_-translate()
mouse (RGB and non RGB)	IS_MOUSE	nA	MSG_SIZE (64)	corsair_-mousecopy()
RGB Keyboard	IS_RGB && !IS_MOUSE	1	8 (BIOS Mode)	hid_kb_translate()
RGB Keyboard	IS_RGB && !IS_MOUSE	2	5 or 21, KB inactive!	hid_kb_translate()
RGB Keyboard	IS_RGB && !IS_MOUSE	3?	MSG_SIZE	corsair_kbcopy()
non RGB Keyboard	!IS_RGB && !IS_MOUSE	nA	nA	hid_kb_translate()

The input data is transformed and copied to the kb structure. Now give it to the OS and unlock the imutex afterwards.

Re-submit the URB for the next run.

If the endless loop is terminated, clean up by discarding the URBs via ioctl(USBDEVFS_DISCARDURB), free the URB buffers and return a null pointer as thread exit code.

Definition at line 248 of file usb_linux.c.

References usbdevice::active, ckb_err, ckb_info, corsair_kbcopy(), corsair_mousecopy(), devpath, usbdevice::epcount, usbdevice::fwversion, usbdevice::handle, hid_kb_translate(), hid_mouse_translate(), imutex, INDEX_OF, usbdevice::input, inputupdate(), IS_MOUSE, IS_RGB, keyboard, usbinput::keys, MSG_SIZE, usbdevice::product, usbinput::rel_x, usbinput::rel_y, and usbdevice::vendor.

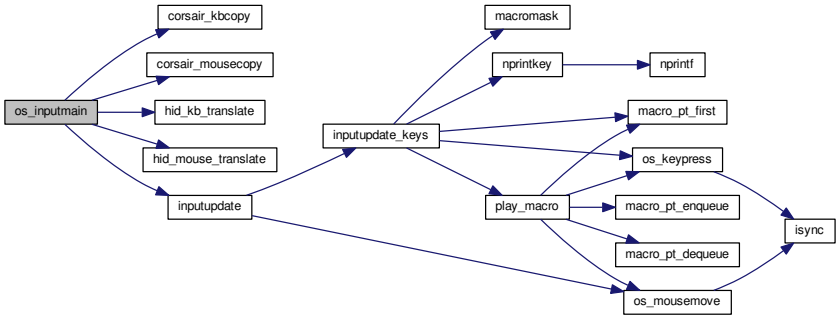
Referenced by _setupusb().

```
248                                {
249     usbdevice* kb = context;
250     int fd = kb->handle - 1;
251     short vendor = kb->vendor, product = kb->product;
252     int index = INDEX_OF(kb, keyboard);
253     ckb_info("Starting input thread for %s%d\n", devpath, index);
254
```

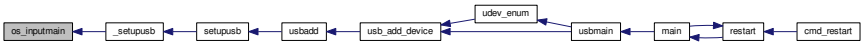
```
259     int urbcount = IS_RGB(vendor, product) ? (kb->epcount - 1) : kb->
    epcount;
260     if (urbcount == 0) {
261         ckb_err("urbcount = 0, so there is nothing to claim in os_inputmain()\n");
262         return 0;
263     }
264 
266     struct usbdevfs_urb urbs[urbcount + 1];
267     memset(urbs, 0, sizeof(urbs));
268 
282     urbs[0].buffer_length = (kb->fwversion >= 0x300 ? MSG_SIZE : 8);
283     if(urbcount > 1 && IS_RGB(vendor, product)) {
284         if(IS_MOUSE(vendor, product))
285             urbs[1].buffer_length = 10;
286         else
287             urbs[1].buffer_length = 21;
288         urbs[2].buffer_length = MSG_SIZE;
289         if(urbcount != 3)
290             urbs[urbcount - 1].buffer_length = MSG_SIZE;
291     } else if(kb->fwversion < 0x300) {
292             urbs[1].buffer_length = 4;
293             urbs[2].buffer_length = 15;
294     }
295 
298     for(int i = 0; i < urbcount; i++){
299         urbs[i].type = USBDEVFS_URB_TYPE_INTERRUPT;
300         urbs[i].endpoint = 0x80 | (i + 1);
301         urbs[i].buffer = malloc(urbs[i].buffer_length);
302         ioctl(fd, USBDEVFS_SUBMITURB, urbs + i);
303     }
304 
306     while (1) {
307         struct usbdevfs_urb* urb = 0;
308 
311         if (ioctl(fd, USBDEVFS_REAPURB, &urb)) {
312             if (errno == ENODEV || errno == ENOENT || errno == ESHUTDOWN)
313                 // Stop the thread if the handle closes
314                 break;
315             else if(errno == EPIPE && urb){
317                 ioctl(fd, USBDEVFS_CLEAR_HALT, &urb->endpoint);
318                 // Re-submit the URB
319                 if(urb)
320                     ioctl(fd, USBDEVFS_SUBMITURB, urb);
321                 urb = 0;
322             }
323             continue;
324         }
325 
329         if (urb) {
330 
342             pthread_mutex_lock(imutex(kb));
343             // EP workaround for FWv3
344             // Corsair input comes through 0x81, but case 1 in keymap.c is used for 6KRO
345             uchar urbendpoint = (kb->fwversion >= 0x300 ? 2 : (urb->endpoint & 0xF));
346             if(IS_MOUSE(vendor, product)){
347                 switch(urb->actual_length){
348                 case 8:
349                 case 10:
350                 case 11:
351                     // HID mouse input
352                     hid_mouse_translate(kb->input.keys, &kb->
    input.rel_x, &kb->input.rel_y, -urbendpoint, urb->actual_length, urb->buffer, kb->
    fwversion);
353                     break;
354                 case MSG_SIZE:
355                     // Corsair mouse input
356                     corsair_mousecopy(kb->input.keys, -urbendpoint, urb->buffer);
357                     break;
358                 }
359             } else if(IS_RGB(vendor, product)){
360                 switch(urb->actual_length){
361                 case 8:
362                     // RGB EP 1: 6KRO (BIOS mode) input
363                     hid_kb_translate(kb->input.keys, -1, urb->actual_length, urb->
    buffer);
364                     break;
365                 case 21:
366                 case 5:
367                     // RGB EP 2: NKRO (non-BIOS) input. Accept only if keyboard is inactive
368                     if(!kb->active)
369                         hid_kb_translate(kb->input.keys, -2, urb->actual_length,
    urb->buffer);
370                     break;
371                 case MSG_SIZE:
372                     // RGB EP 3: Corsair input
373                     corsair_kbcopy(kb->input.keys, -urbendpoint, urb->buffer);
374                     break;
```

```
375                 }
376             } else {
377                 // Non-RGB input
378                 hid_kb_translate(kb->input.keys, urb->endpoint & 0xF, urb->
    actual_length, urb->buffer);
379             }
382             inputupdate(kb);
383             pthread_mutex_unlock(imutex(kb));
384
386             ioctl(fd, USBDEVFS_SUBMITURB, urb);
387             urb = 0;
388         }
389     }
390
394     ckb_info("Stopping input thread for %s%d\n", devpath, index);
395     for(int i = 0; i < urbcount; i++){
396         ioctl(fd, USBDEVFS_DISCARDURB, urbs + i);
397         free(urbs[i].buffer);
398     }
399     return 0;
400 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.49.3.8 int os_resetusb (usbdevice * *kb*, const char * *file*, int *line*)

Parameters

kb	THE usbdevice*
file	filename for error messages
line	line where it is called for error messages

Returns

Returns 0 on success, -2 if device should be removed and -1 if reset should by tried again

os_resetusb is the os specific implementation for resetting usb

Try to reset an usb device in a linux user space driver.

1. unclaim the device, but do not reconnect the system driver (second param resetting = true)

2. reset the device via USBDEVFS_RESET command
3. claim the device again. Returns 0 on success, -2 if device should be removed and -1 if reset should by tried again

Todo it seems that no one wants to try the reset again. But I'v seen it somewhere...

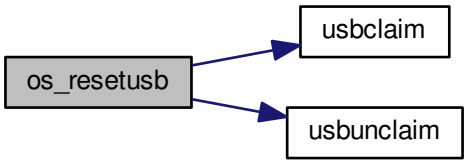
Definition at line 510 of file usb_linux.c.

References usbdevice::handle, TEST_RESET, usbclaim(), and usbunclaim().

Referenced by _resetusb().

```
510                                                                 {
511     TEST_RESET(usbunclaim(kb, 1));
512     TEST_RESET(ioctl(kb->handle - 1, USBDEVFS_RESET));
513     TEST_RESET(usbclaim(kb));
514     // Success!
515     return 0;
516 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.49.3.9 void os_sendindicators (usbdevice * *kb*)

Parameters

kb	THE usbdevice*

os_sendindicators update the indicators for the special keys (Numlock, Capslock and what else?)

os_sendindicators update the indicators for the special keys (Numlock, Capslock and what else?)

Read the data from kb->ileds ans send them via ioctl() to the keyboard.

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0x21	0x09	0x0200	Interface 0	MSG_SIZE 1 Byte	timeout 0,5ms	the message buffer pointer
Host to Device, Type=Class, Recipi-ent=Interface (why not endpoint?)	9 = SEND?	specific	0	1	500	struct∗ kb->ileds

The ioctl command is USBDEVFS_CONTROL.

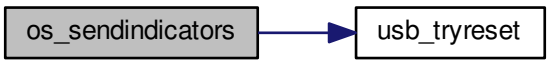
Definition at line 214 of file usb_linux.c.

References ckb_err, usbdevice::fwversion, usbdevice::handle, usbdevice::ileds, and usb_tryreset().

Referenced by updateindicators_kb().

```
214                                     {
215     static int countForReset = 0;
216     void *ileds;
217     ushort leds;
218     if(kb->fwversion >= 0x300) {
219         leds = (kb->ileds << 8) | 0x0001;
220         ileds = &leds;
221     }
222     else {
223         ileds = &kb->ileds;
224     }
225     struct usbdevfs_ctrltransfer transfer = { 0x21, 0x09, 0x0200, 0x00, (kb->
      fwversion >= 0x300 ? 2 : 1), 500, ileds };
226     int res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
227     if(res <= 0) {
228         ckb_err("%s\n", res ? strerror(errno) : "No data written");
229         if (usb_tryreset(kb) == 0 && countForReset++ < 3) {
230             os_sendindicators(kb);
231         }
232     }
233 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.49.3.10 int os_setupusb (usbdevice * *kb*)

Parameters

kb	THE usbdevice*

Returns

0 on success, -1 otherwise.

os_setupusb OS-specific setup for a specific usb device.

Perform the operating system-specific opening of the interface in os_setupusb(). As a result, some parameters should be set in kb (name, serial, fwversion, epcount = number of usb endpoints), and all endpoints should be claimed with usbclaim(). Claiming is the only point where os_setupusb() can produce an error (-1).

- Copy device description and serial
- Copy firmware version (needed to determine USB protocol)
- Do some output about connecting interfaces
- Claim the USB interfaces

Todo in these modules a pullrequest is outstanding

< Try to reset the device and recall the function

< Don't do this endless in recursion

< os_setupusb() has a return value (used as boolean)

Definition at line 548 of file usb_linux.c.

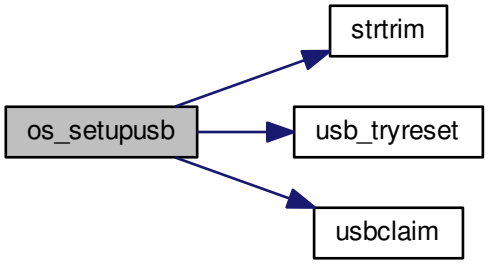
References ckb_err, ckb_info, devpath, usbdevice::epcount, usbdevice::fwversion, INDEX_OF, KB_NAME_LEN, keyboard, usbdevice::name, usbdevice::serial, SERIAL_LEN, strtrim(), usbdevice::udev, usb_tryreset(), and usbclaim().

Referenced by _setupusb().

```
548                                {
551     struct udev_device* dev = kb->udev;
552     const char* name = udev_device_get_sysattr_value(dev, "product");
553     if(name)
554         strncpy(kb->name, name, KB_NAME_LEN);
555     strtrim(kb->name);
556     const char* serial = udev_device_get_sysattr_value(dev, "serial");
557     if(serial)
558         strncpy(kb->serial, serial, SERIAL_LEN);
559     strtrim(kb->serial);
562     const char* firmware = udev_device_get_sysattr_value(dev, "bcdDevice");
563     if(firmware)
564         sscanf(firmware, "%hx", &kb->fwversion);
565     else
566         kb->fwversion = 0;
567     int index = INDEX_OF(kb, keyboard);
568
570     ckb_info("Connecting %s at %s%d\n", kb->name, devpath, index);
571
577     const char* ep_str = udev_device_get_sysattr_value(dev, "bNumInterfaces");
578 #ifdef DEBUG
579     ckb_info("claiming interfaces. name=%s, firmware=%s; ep_str=%s\n", name, firmware, ep_str);
580 #endif //DEBUG
581     kb->epcount = 0;
582     if(ep_str)
583         sscanf(ep_str, "%d", &kb->epcount);
584     if(kb->epcount < 2){
585         // IF we have an RGB KB with 0 or 1 endpoints, it will be in BIOS mode.
586         ckb_err("Unable to read endpoint count from udev, assuming %d and reading >>%s<< or device
       is in BIOS mode\n", kb->epcount, ep_str);
587         if (usb_tryreset(kb) == 0) {
588             static int retryCount = 0;
589             if (retryCount++ < 5) {
590                 return os_setupusb(kb);
591             }
592         }
```

```
593         return -1;
594         // ToDo are there special versions we have to detect? If there are, that was the old code to handle
    it:
595         // This shouldn't happen, but if it does, assume EP count based onckb_warn what the device is
    supposed to have
596         // kb->epcount = (HAS_FEATURES(kb, FEAT_RGB) ? 4 : 3);
597         // ckb_warn("Unable to read endpoint count from udev, assuming %d and reading >>%s<<...\n",
    kb->epcount, ep_str);
598      }
599      if(usbclaim(kb)){
600         ckb_err("Failed to claim interfaces: %s\n", strerror(errno));
601         return -1;
602      }
603      return 0;
604 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.49.3.11 int os_usbrecv (usbdevice * *kb*, uchar * *in_msg*, const char * *file*, int *line*)

Parameters

kb	THE usbdevice*
in_msg	the buffer to fill with the message received
file	for debugging
line	for debugging

Returns

-1 on timeout, 0 on hard error, numer of bytes received otherwise

os_usbrecv does what its name says:

The comment at the beginning of the procedure causes the suspicion that the firmware versionspecific distinction is missing for receiving from usb endpoint 3 or 4. The commented code contains only the reception from EP4, but this may be wrong for a software version 2.0 or higher (see the code for os-usbsend ()).

So all the receiving is done via an ioctl() like in os_usbsend. The ioctl() is given a struct usbdevfs_ctrltransfer, in which the relevant parameters are entered:

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0xA1	0x01	0x0200	endpoint to be addressed from epcount - 1	MSG_SIZE	5ms	the message buffer pointer
Device to Host, Type=Class, Recipi-ent=Interface	1 = RECEIVE?	specific	Interface #	64	5000	in_msg

The ioctl() returns the number of bytes received. Here is the usual check again:

- If the return value is -1 AND the error is a timeout (ETIMEOUT), os_usbrecv() will return -1 to indicate that it is probably a recoverable problem and a retry is recommended.

- For another negative value or other error identifier OR 0 bytes are received, 0 is returned as an identifier for a heavy error.

- In all other cases, the function returns the number of bytes received.

If this is not the entire blocksize (MSG_SIZE bytes), an error message is issued on the standard error channel [warning "Read YY bytes (expected 64)"].

Definition at line 130 of file usb_linux.c.

References ckb_err_fn, ckb_warn_fn, usbdevice::epcount, usbdevice::handle, and MSG_SIZE.

Referenced by _usbrecv().

```
130                                                                         {
131     int res;
132     // This is what CUE does, but it doesn't seem to work on linux.
133     /*if(kb->fwversion >= 0x130){
134         struct usbdevfs_bulktransfer transfer = {0};
135         transfer.ep = 0x84;
136         transfer.len = MSG_SIZE;
137         transfer.timeout = 5000;
138         transfer.data = in_msg;
139         res = ioctl(kb->handle - 1, USBDEVFS_BULK, &transfer);
140     } else {*/
141         struct usbdevfs_ctrltransfer transfer = { 0xa1, 0x01, 0x0300, kb->
    epcount - 1, MSG_SIZE, 5000, in_msg };
142         res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
143     //}
144     if(res <= 0){
145         ckb_err_fn("%s\n", file, line, res ? strerror(errno) : "No data read");
146         if(res == -1 && errno == ETIMEDOUT)
147             return -1;
148         else
149             return 0;
150     } else if(res != MSG_SIZE)
151         ckb_warn_fn("Read %d bytes (expected %d)\n", file, line, res,
    MSG_SIZE);
152 #ifdef DEBUG_USB_RECV
153     char converted[MSG_SIZE*3 + 1];
154     for(int i=0;i<MSG_SIZE;i++)
155         sprintf(&converted[i*3], "%02x ", in_msg[i]);
156     ckb_warn_fn("Recv %s\n", file, line, converted);
157 #endif
158     return res;
159 }
```

Here is the caller graph for this function:

13.49.3.12 int os_usbsend (usbdevice * *kb,* const uchar * *out_msg,* int *is_recv,* const char * *file,* int *line*)

Parameters

kb	THE usbdevice*
out_msg	the MSGSIZE char long buffer to send
is_recv	if true, just send an ioctl for further reading packets. If false, send the data at **out_msg**.
file	for debugging
line	for debugging

Returns

-1 on timeout (try again), 0 on hard error, numer of bytes sent otherwise

os_usbsend has two functions:

- if is_recv == false, it tries to send a given MSG_SIZE buffer via the usb interface given with kb.
- otherwise a request is sent via the usb device to initiate the receiving of a message from the remote device.

The functionality for sending distinguishes two cases, depending on the version number of the firmware of the connected device:

If the firmware is less or equal 1.2, the transmission is done via an ioctl(). The ioctl() is given a struct usbdevfs_ctrltransfer, in which the relevant parameters are entered:

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0x21	0x09	0x0200	endpoint / IF to be addressed from epcount-1	MSG_SIZE	5000 (=5ms)	the message buffer pointer
Host to Device, Type=Class, Recipient=Interface	9 = Send data?	specific	last or pre-last device #	64	5000	out_msg

The ioctl command is USBDEVFS_CONTROL.

The same constellation is used if the device is requested to send its data (is_recv = true).

For a more recent firmware and is_recv = false, the ioctl command USBDEVFS_CONTROL is not used (this tells the bus to enter the control mode), but the bulk method is used: USBDEVFS_BULK. This is astonishing, because all of the endpoints are type Interrupt, not bulk.

Anyhow, forthis purpose a different structure is used for the ioctl() (struct **usbdevfs_bulktransfer**) and this is also initialized differently:

The length and timeout parameters are given the same values as above. The formal parameter out_msg is also passed as a buffer pointer. For the endpoints, the firmware version is differentiated again:

For a firmware version between 1.3 and <2.0 endpoint 4 is used, otherwise (it can only be >=2.0) endpoint 3 is used.

Todo Since the handling of endpoints has already led to problems elsewhere, this implementation is extremely hardware-dependent and critical!

Eg. the new keyboard K95PLATINUMRGB has a version number significantly less than 2.0 - will it run with this implementation?

The ioctl() - no matter what type - returns the number of bytes sent. Now comes the usual check:

- If the return value is -1 AND the error is a timeout (ETIMEOUT), os_usbsend() will return -1 to indicate that it is probably a recoverable problem and a retry is recommended.
- For another negative value or other error identifier OR 0 bytes sent, 0 is returned as a heavy error identifier.
- In all other cases, the function returns the number of bytes sent.

If this is not the entire blocksize (MSG_SIZE bytes), an error message is issued on the standard error channel [warning "Wrote YY bytes (expected 64)"].

If DEBUG_USB_SEND is set during compilation, the number of bytes sent and their representation are logged to the error channel.

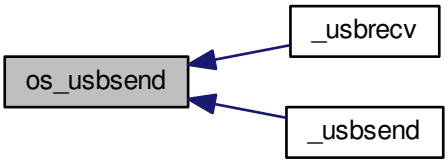
Definition at line 68 of file usb_linux.c.

References ckb_err_fn, ckb_warn_fn, usbdevice::epcount, usbdevice::fwversion, usbdevice::handle, IS_V2_OVERRIDE, and MSG_SIZE.

Referenced by _usbrecv(), and _usbsend().

```
68                                                                                                  {
69     int res;
70     if ((kb->fwversion >= 0x120 || IS_V2_OVERRIDE(kb)) && !is_recv){
71         struct usbdevfs_bulktransfer transfer = {0};
72         // FW 2.XX uses 0x03, FW 3.XX uses 0x02
73         transfer.ep = (kb->fwversion >= 0x130 && kb->fwversion < 0x200) ? 4 : (kb->
     fwversion >= 0x300 ? 2 : 3);
74         transfer.len = MSG_SIZE;
75         transfer.timeout = 5000;
76         transfer.data = (void*)out_msg;
77         res = ioctl(kb->handle - 1, USBDEVFS_BULK, &transfer);
78     } else {
79         struct usbdevfs_ctrltransfer transfer = { 0x21, 0x09, 0x0200, kb->
     epcount - 1, MSG_SIZE, 5000, (void*)out_msg };
80         res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
81     }
82
83     if (res <= 0){
84         ckb_err_fn(" %s, res = 0x%x\n", file, line, res ? strerror(errno) : "No data written",
     res);
85         if(res == -1 && errno == ETIMEDOUT)
86             return -1;
87         else
88             return 0;
89     } else if (res != MSG_SIZE)
90         ckb_warn_fn("Wrote %d bytes (expected %d)\n", file, line, res,
     MSG_SIZE);
91 #ifdef DEBUG_USB_SEND
92     char converted[MSG_SIZE*3 + 1];
93     for(int i=0;i<MSG_SIZE;i++)
94         sprintf(&converted[i*3], "%02x ", out_msg[i]);
95     ckb_warn_fn("Sent %s\n", file, line, converted);
96 #endif
97     return res;
98 }
```

Here is the caller graph for this function:



13.49.3.13 const char* product_str (short *product*)

Parameters

product	is the *short* USB device product ID

Returns

string to identify a type of device (see below)

product_str returns a condensed view on what type of device we have.

At present, various models and their properties are known from corsair products. Some models differ in principle (mice and keyboards), others differ in the way they function (for example, RGB and non RGB), but they are very similar.

Here, only the first point is taken into consideration and we return a unified model string. If the model is not known with its number, *product_str* returns an empty string.

The model numbers and corresponding strings wwith the numbers in hex-string are defined in `usb.h`

At present, this function is used to initialize `kb->name` and to give information in debug strings.

Attention

The combinations below have to fit to the combinations in the macros mentioned above. So if you add a device with a new number, change both.

Todo There are macros defined in usb.h to detect all the combinations below. the only difference is the parameter: The macros need the *kb**, product_str() needs the *product ID*

Definition at line 70 of file usb.c.

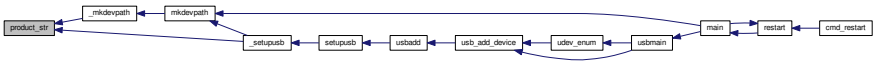
References P_GLAIVE, P_HARPOON, P_K63_NRGB, P_K65, P_K65_LUX, P_K65_NRGB, P_K65_RFIRE, P_K68, P_K70, P_K70_LUX, P_K70_LUX_NRGB, P_K70_NRGB, P_K70_RFIRE, P_K70_RFIRE_NRGB, P_K95, P_K95_NRGB, P_K95_PLATINUM, P_M65, P_M65_PRO, P_SABRE_L, P_SABRE_N, P_SABRE_O, P_SABRE_O2, P_SCIMITAR, P_SCIMITAR_PRO, P_STRAFE, P_STRAFE_NRGB, and P_STRAFE_NRGB_2.

Referenced by _mkdevpath(), and _setupusb().

```
70                                      {
71     if(product == P_K95 || product == P_K95_NRGB)
72         return "k95";
73     if(product == P_K95_PLATINUM)
74         return "k95p";
75     if(product == P_K70 || product == P_K70_NRGB || product ==
```

```
      P_K70_LUX || product == P_K70_LUX_NRGB || product ==
      P_K70_RFIRE || product == P_K70_RFIRE_NRGB)
76         return "k70";
77      if(product == P_K68)
78         return "k68";
79      if(product == P_K65 || product == P_K65_NRGB || product ==
      P_K65_LUX || product == P_K65_RFIRE)
80         return "k65";
81      if(product == P_K63_NRGB)
82         return "k63";
83      if(product == P_STRAFE || product == P_STRAFE_NRGB || product ==
      P_STRAFE_NRGB_2)
84         return "strafe";
85      if(product == P_M65 || product == P_M65_PRO)
86         return "m65";
87      if(product == P_SABRE_O || product == P_SABRE_L || product ==
      P_SABRE_N || product == P_SABRE_O2)
88         return "sabre";
89      if(product == P_SCIMITAR || product == P_SCIMITAR_PRO)
90         return "scimitar";
91      if(product == P_HARPOON)
92         return "harpoon";
93      if(product == P_GLAIVE)
94         return "glaive";
95      return "";
96  }
```

Here is the caller graph for this function:



13.49.3.14 int revertusb (usbdevice * *kb*)

Parameters

kb	THE usbdevice*

Returns

0 on success or if device needs firmware upgrade, -1 otherwise

revertusb sets a given device to inactive (hardware controlled) mode if not a fw-ugrade is indicated

First is checked, whether a firmware-upgrade is indicated for the device. If so, revertusb() returns 0.

Todo Why is this useful? Are there problems seen with deactivating a device with older fw-version??? Why isn't this an error indicating reason and we return success (0)?

Anyway, the following steps are similar to some other procs, dealing with low level usb handling:

- If we do not have an RGB device, a simple setting to Hardware-mode (NK95_HWON) is sent to the device via n95cmd().

  Todo The return value of nk95cmd() is ignored (but sending the ioctl may produce an error and _nk95_cmd will indicate this), instead revertusb() returns success in any case.

- If we have an RGB device, setactive() is called with second param active = false. That function will have a look on differences between keyboards and mice.

  More precisely setactive() is just a macro to call via the kb->vtable enties either the active() or the idle() function where the vtable points to. setactive() may return error indications. If so, revertusb() returns -1, otherwise 0 in any other case.

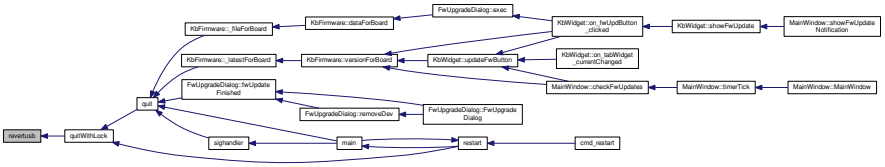
Definition at line 417 of file usb.c.

References FEAT_RGB, HAS_FEATURES, NEEDS_FW_UPDATE, NK95_HWON, nk95cmd, and setactive.

Referenced by quitWithLock().

```
417                             {
418     if(NEEDS_FW_UPDATE(kb))
419         return 0;
420     if(!HAS_FEATURES(kb, FEAT_RGB)){
421         nk95cmd(kb, NK95_HWON);
422         return 0;
423     }
424     if(setactive(kb, 0))
425         return -1;
426     return 0;
427 }
```

Here is the caller graph for this function:



13.49.3.15 void setupusb (usbdevice * *kb*)

Attention

Lock a device's dmutex (see device.h) before accessing the USB interface.

Parameters

kb	THE usbdevice* used everywhere
OUT]	kb->thread is used to store the thread ID of the fresh created thread.

setupusb starts a thread with kb as parameter and _setupusb() as entrypoint.

Set up a USB device after its handle is open. Spawns a new thread _setupusb() with standard parameter kb. dmutex must be locked prior to calling this function. The function will unlock it when finished. In kb->thread the thread id is mentioned, because closeusb() needs this info for joining that thread again.

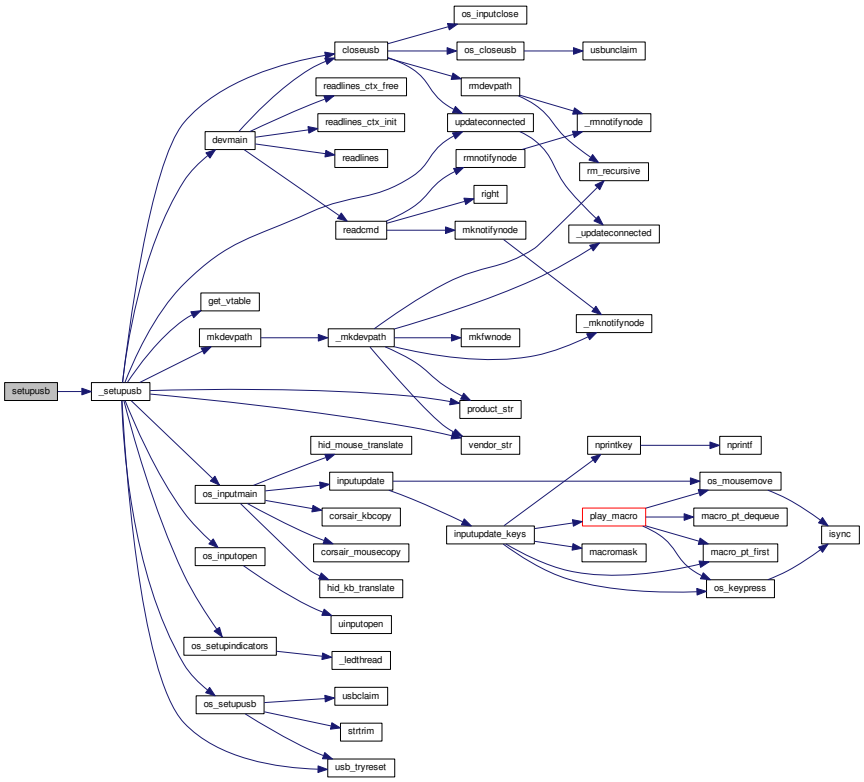
Definition at line 396 of file usb.c.

References _setupusb(), ckb_err, imutex, and usbdevice::thread.

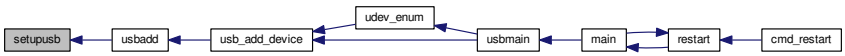
Referenced by usbadd().

```
396                             {
397     pthread_mutex_lock(imutex(kb));
398     if(pthread_create(&kb->thread, 0, _setupusb, kb))
399         ckb_err("Failed to create USB thread\n");
400 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.49.3.16 int usb_tryreset (usbdevice * *kb*)

Parameters

in,out	*kb*	THE usbdevice*
in	*reset_stop*	global variable is read

Returns

0 on success, -1 otherwise

usb_tryreset does what the name means: Try to reset the usb via resetusb()

This function is called if an usb command ran into an error in case of one of the following two situations:

- When setting up a new usb device and the start() function got an error (

See Also

_setupusb())

- If upgrading to a new firmware gets an error (

See Also

cmd_fwupdate()).

The previous action which got the error will NOT be re-attempted.

In an endless loop usb_tryreset() tries to reset the given usb device via the macro resetusb().

This macro calls _resetusb() with debugging information.

_resetusb() sends a command via the operating system dependent function os_resetusb() and - if successful - reinitializes the device. os_resetusb() returns -2 to indicate a broken device and all structures should be removed for it.

In that case, the loop is terminated, an error message is produced and usb_tryreset() returns -1.

In case resetusb() has success, the endless loop is left via a return 0 (success).

If the return value from resetusb() is -1, the loop is continued with the next try.

If the global variable **reset_stop** is set directly when the function is called or after each try, usb_tryreset() stops working and returns -1.

Todo Why does usb_tryreset() hide the information returned from resetusb()? Isn't it needed by the callers?

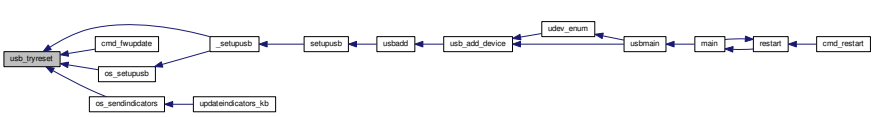
Definition at line 475 of file usb.c.

References ckb_err, ckb_info, reset_stop, and resetusb.

Referenced by _setupusb(), cmd_fwupdate(), os_sendindicators(), and os_setupusb().

```
475                                 {
476     if(reset_stop)
477         return -1;
478     ckb_info("Attempting reset...\n");
479     while(1){
480         int res = resetusb(kb);
481         if(!res){
482             ckb_info("Reset success\n");
483             return 0;
484         }
485         if(res == -2 || reset_stop)
486             break;
487     }
488     ckb_err("Reset failed. Disconnecting.\n");
489     return -1;
490 }
```

Here is the caller graph for this function:



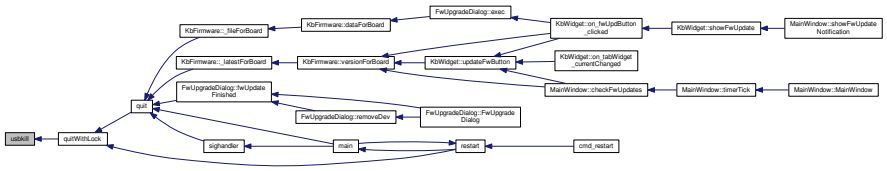
13.49.3.17 void usbkill ( )

Definition at line 853 of file usb_linux.c.

Referenced by quitWithLock().

```
853             {
854     udev_unref(udev);
855     udev = 0;
856 }
```

Here is the caller graph for this function:



13.49.3.18 int usbmain ( )

Start the USB main loop. Returns program exit code when finished.

usbmain is called by main() after setting up all other stuff.

Returns

0 normally or -1 if fatal error occurs (up to now only if no new devices are available)

First check whether the uinput module is loaded by the kernel.

Todo Why isn't missing of uinput a fatal error?

Create the udev object with udev_new() (is a function from libudev.h) terminate -1 if error

Enumerate all currently connected devices

Todo lae. here the work has to go on...

Definition at line 793 of file usb_linux.c.

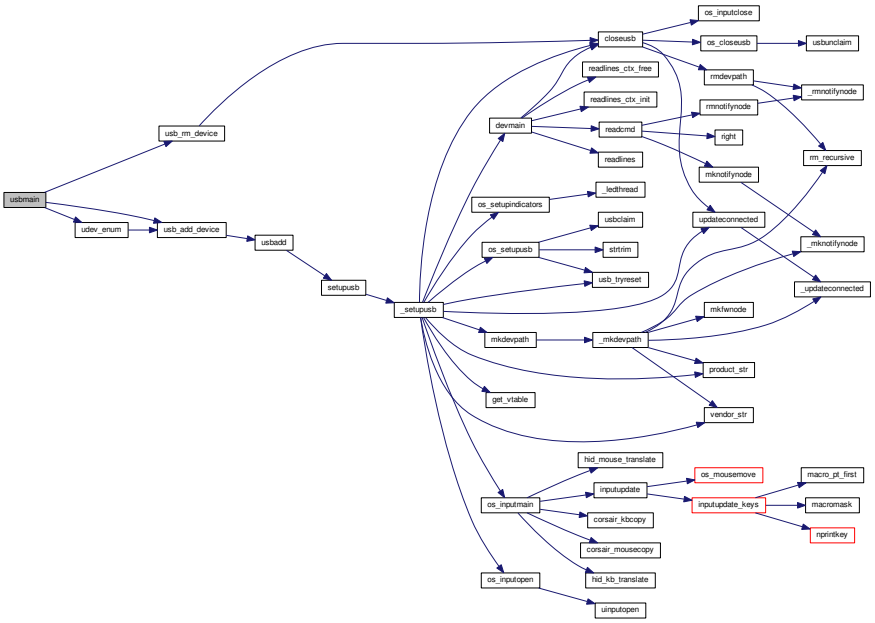
References ckb_fatal, ckb_warn, udev_enum(), usb_add_device(), and usb_rm_device().

Referenced by main().

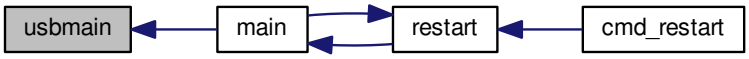
```
793             {
798     // Load the uinput module (if it's not loaded already)
799     if(system("modprobe uinput") != 0)
800         ckb_warn("Failed to load uinput module\n");
801
805     if(!(udev = udev_new())) {
806         ckb_fatal("Failed to initialize udev in usbmain(), usb_linux.c\n");
807         return -1;
808     }
809
812     udev_enum();
813
816     // Done scanning. Enter a loop to poll for device updates
817     struct udev_monitor* monitor = udev_monitor_new_from_netlink(udev, "udev");
818     udev_monitor_filter_add_match_subsystem_devtype(monitor, "usb", 0);
819     udev_monitor_enable_receiving(monitor);
820     // Get an fd for the monitor
821     int fd = udev_monitor_get_fd(monitor);
822     fd_set fds;
823     while(udev){
824         FD_ZERO(&fds);
825         FD_SET(fd, &fds);
826         // Block until an event is read
827         if(select(fd + 1, &fds, 0, 0, 0) > 0 && FD_ISSET(fd, &fds)){
828             struct udev_device* dev = udev_monitor_receive_device(monitor);
829             if(!dev)
```

```
830                 continue;
831             const char* action = udev_device_get_action(dev);
832             if(!action){
833                 udev_device_unref(dev);
834                 continue;
835             }
836             // Add/remove device
837             if(!strcmp(action, "add")){
838                 int res = usb_add_device(dev);
839                 if(res == 0)
840                     continue;
841                 // If the device matched but the handle wasn't opened correctly, re-enumerate (this
    sometimes solves the problem)
842                 if(res == -1)
843                     udev_enum();
844             } else if(!strcmp(action, "remove"))
845                 usb_rm_device(dev);
846             udev_device_unref(dev);
847         }
848     }
849     udev_monitor_unref(monitor);
850     return 0;
851 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.49.3.19 const char* vendor_str (short *vendor*)

uncomment to see USB packets received from the device vendor_str Vendor/product string representations

Parameters

vendor	*short* vendor ID

Returns

a string: either "" or "corsair"

uncomment to see USB packets sent to the device

vendor_str returns "corsair" if the given *vendor* argument is equal to *V_CORSAIR* (0x1bc) else it returns ""

Attention

There is also a string defined V_CORSAIR_STR, which returns the device number as string in hex "1b1c".

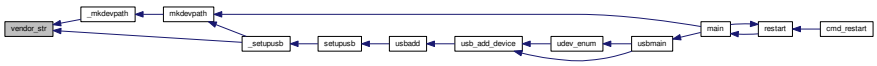
Definition at line 43 of file usb.c.

References V_CORSAIR.

Referenced by _mkdevpath(), and _setupusb().

```
43                                      {
44     if(vendor == V_CORSAIR)
45         return "corsair";
46     return "";
47 }
```

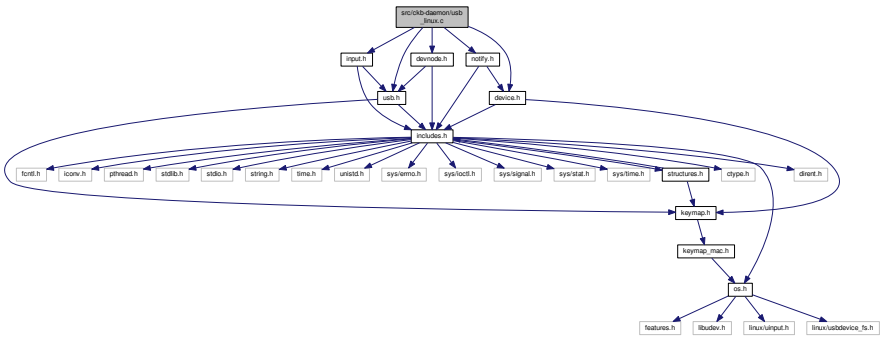
Here is the caller graph for this function:



13.50 src/ckb-daemon/usb_linux.c File Reference

```
#include "device.h"
#include "devnode.h"
#include "input.h"
#include "notify.h"
#include "usb.h"
```

Include dependency graph for usb_linux.c:



Data Structures

- struct _model

Macros

- #define DEBUG

  all open usb devices have their system path names here in this array.
- #define TEST_RESET(op)

  TEST_RESET doesa "try / catch" for resetting the usb interface.
- #define N_MODELS (sizeof(models) / sizeof(_model))

Functions

- int os_usbsend (usbdevice *kb, const uchar *out_msg, int is_recv, const char *file, int line)

  os_usbsend sends a data packet (MSG_SIZE = 64) Bytes long
- int os_usbrecv (usbdevice *kb, uchar *in_msg, const char *file, int line)

  os_usbrecv receives a max MSGSIZE long buffer from usb device
- int _nk95cmd (usbdevice *kb, uchar bRequest, ushort wValue, const char *file, int line)

  _nk95cmd If we control a non RGB keyboard, set the keyboard via ioctl with usbdevfs_ctrltransfer
- void os_sendindicators (usbdevice *kb)
- void * os_inputmain (void *context)

  os_inputmain This function is run in a separate thread and will be detached from the main thread, so it needs to clean up its own resources.
- static int usbunclaim (usbdevice *kb, int resetting)
- void os_closeusb (usbdevice *kb)
- static int usbclaim (usbdevice *kb)
- int os_resetusb (usbdevice *kb, const char *file, int line)
- void strtrim (char *string)
- int os_setupusb (usbdevice *kb)
- int usbadd (struct udev_device *dev, short vendor, short product)
- static int usb_add_device (struct udev_device *dev)

  Add a udev device. Returns 0 if device was recognized/added.
- static void usb_rm_device (struct udev_device *dev)

  usb_rm_device find the usb port to remove and close it via closeusb().
- static void udev_enum ()

  udev_enum use the udev_enumerate_add_match_subsystem() to get all you need but only that.
- int usbmain ()
- void usbkill ()

  Stop the USB system.

Variables

- static char kbsyspath [9][FILENAME_MAX]
- static struct udev * udev

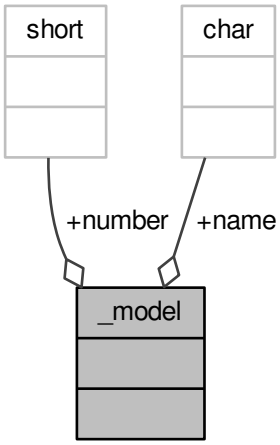
  struct udef is defined in /usr/include/libudev.h
- pthread_t usbthread
- pthread_t udevthread
- static _model models []

13.50.1 Data Structure Documentation

13.50.1.1 struct _model

Definition at line 658 of file usb_linux.c.

Collaboration diagram for _model:



Data Fields

const char *	name	
short	number	

13.50.2 Macro Definition Documentation

13.50.2.1 #define DEBUG

Definition at line 11 of file usb_linux.c.

13.50.2.2 #define N_MODELS (sizeof(models) / sizeof(_model))

Definition at line 700 of file usb_linux.c.

Referenced by usb_add_device().

13.50.2.3 #define TEST_RESET(*op*)

Value:

```
if(op){                                                                 \
        ckb_err_fn("resetusb failed: %s\n", file, line, strerror(errno));   \
        if(errno == EINTR || errno == EAGAIN)                           \
            return -1;              /* try again if status code says so */  \
        return -2;                  /* else, remove device */               \
    }
```

Definition at line 492 of file usb_linux.c.

Referenced by os_resetusb().

13.50.3 Function Documentation

13.50.3.1 int _nk95cmd (usbdevice * *kb,* uchar *bRequest,* ushort *wValue,* const char * *file,* int *line*)

To send control packets to a non RGB non color K95 Keyboard, use this function. Normally it is called via the nk95cmd() macro.

If it is the wrong device for which the function is called, 0 is returned and nothing done. Otherwise a usbdevfs_ctrltransfer structure is filled and an USBDEVFS_CONTROL ioctl() called.

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0x40	see table below to switch hardware-modus at Keyboard	wValue	device	MSG_SIZE	5ms	the message buffer pointer
Host to Device, Type=Vendor, Recipient=Device	bRequest parameter	given wValue Parameter	device 0	0 data to write	5000	null

If a 0 or a negative error number is returned by the ioctl, an error message is shown depending on the errno or "No data written" if retval was 0. In either case 1 is returned to indicate the error. If the ioctl returned a value $>$ 0, 0 is returned to indicate no error.

Currently the following combinations for bRequest and wValue are used:

Device	what it might to do	constant	bRequest	wValue
non RGB Keyboard	set HW-modus on (leave the ckb driver)	HWON	0x0002	0x0030
non RGB Keyboard	set HW-modus off (initialize the ckb driver)	HWOFF	0x0002	0x0001
non RGB Keyboard	set light modus M1 in single-color keyboards	NK95_M1	0x0014	0x0001
non RGB Keyboard	set light modus M2 in single-color keyboards	NK95_M2	0x0014	0x0002
non RGB Keyboard	set light modus M3 in single-color keyboards	NK95_M3	0x0014	0x0003

See Also

usb.h

Definition at line 189 of file usb_linux.c.

References ckb_err_fn, usbdevice::handle, P_K95_NRGB, and usbdevice::product.

```
189                                                                     {
190     if(kb->product != P_K95_NRGB)
```

```
191         return 0;
192     struct usbdevfs_ctrltransfer transfer = { 0x40, bRequest, wValue, 0, 0, 5000, 0 };
193     int res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
194     if(res <= 0){
195         ckb_err_fn("%s\n", file, line, res ? strerror(errno) : "No data written");
196         return 1;
197     }
198     return 0;
199 }
```

13.50.3.2 void os_closeusb (usbdevice * *kb*)

os_closeusb unclaim it, destroy the udev device and clear data structures at kb

os_closeusb is the linux specific implementation for closing an active usb port.

If a valid handle is given in the kb structure, the usb port is unclaimed (usbunclaim()).

The device in unrefenced via library function udev_device_unref().

handle, udev and the first char of kbsyspath are cleared to 0 (empty string for kbsyspath).

Definition at line 448 of file usb_linux.c.

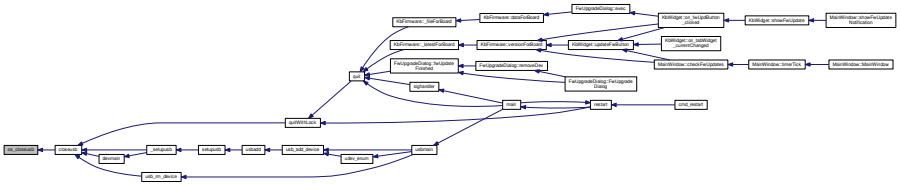
References usbdevice::handle, INDEX_OF, kbsyspath, keyboard, usbdevice::udev, and usbunclaim().

Referenced by closeusb().

```
448                                 {
449     if(kb->handle){
450         usbunclaim(kb, 0);
451         close(kb->handle - 1);
452     }
453     if(kb->udev)
454         udev_device_unref(kb->udev);
455     kb->handle = 0;
456     kb->udev = 0;
457     kbsyspath[INDEX_OF(kb, keyboard)][0] = 0;
458 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.50.3.3 void* os_inputmain (void * *context*)

os_inputmain is run in a separate thread and will be detached from the main thread, so it needs to clean up its own resources.

Todo This function is a collection of many tasks. It should be divided into several sub-functions for the sake of greater convenience:

1. set up an URB (Userspace Ressource Buffer) to communicate with the USBDEVFS_* ioctl()s
2. perform the ioctl()
3. interpretate the information got into the URB buffer or handle error situations and retry operation or leave the endless loop
4. inform the os about the data
5. loop endless via 2.
6. if endless loop has gone, deinitalize the interface, free buffers etc.
7. return null

Here the actions in detail:

Monitor input transfers on all endpoints for non-RGB devices For RGB, monitor all but the last, as it's used for input/output

Get an usbdevfs_urb data structure and clear it via memset()

Hopefully the buffer lengths are equal for all devices with congruent types. You can find out the correctness for your device with lsusb –v or similar on macOS. Currently the following combinations are known and implemented:

device	detect with macro combination	endpoint #	buffer-length
each	none	0	8, 64 for FW v3
RGB Mouse	IS_RGB && IS_MOUSE	1	10
RGB Keyboard	IS_RGB && !IS_MOUSE	1	21
RGB Mouse or Keyboard	IS_RGB	2	MSG_SIZE (64)
non RGB Mouse or Keyboard	!IS_RGB	1	4
non RGB Mouse or Keyboard	!IS_RGB	2	15

Now submit all the URBs via ioctl(USBDEVFS_SUBMITURB) with type USBDEVFS_URB_TYPE_INTERRUPT (the endpoints are defined as type interrupt). Endpoint number is 0x80..0x82 or 0x83, depending on the model.

The userSpaceFS knows the URBs now, so start monitoring input

if the ioctl returns something != 0, let's have a deeper look what happened. Broken devices or shutting down the entire system leads to closing the device and finishing this thread.

If just an EPIPE ocurred, give the device a CLEAR_HALT and resubmit the URB.

A correct REAPURB returns a Pointer to the URB which we now have a closer look into. Lock all following actions with imutex.

Process the input depending on type of device. Interprete the actual size of the URB buffer

device	detect with macro combination	seems to be endpoint #	actual buffer-length	function called

mouse (RGB and non RGB)	IS_MOUSE	nA	8, 10 or 11	hid_mouse_-translate()
mouse (RGB and non RGB)	IS_MOUSE	nA	MSG_SIZE (64)	corsair_-mousecopy()
RGB Keyboard	IS_RGB && !IS_MOUSE	1	8 (BIOS Mode)	hid_kb_translate()
RGB Keyboard	IS_RGB && !IS_MOUSE	2	5 or 21, KB inactive!	hid_kb_translate()
RGB Keyboard	IS_RGB && !IS_MOUSE	3?	MSG_SIZE	corsair_kbcopy()
non RGB Keyboard	!IS_RGB && !IS_MOUSE	nA	nA	hid_kb_translate()

The input data is transformed and copied to the kb structure. Now give it to the OS and unlock the imutex afterwards.

Re-submit the URB for the next run.

If the endless loop is terminated, clean up by discarding the URBs via ioctl(USBDEVFS_DISCARDURB), free the URB buffers and return a null pointer as thread exit code.

Definition at line 248 of file usb_linux.c.

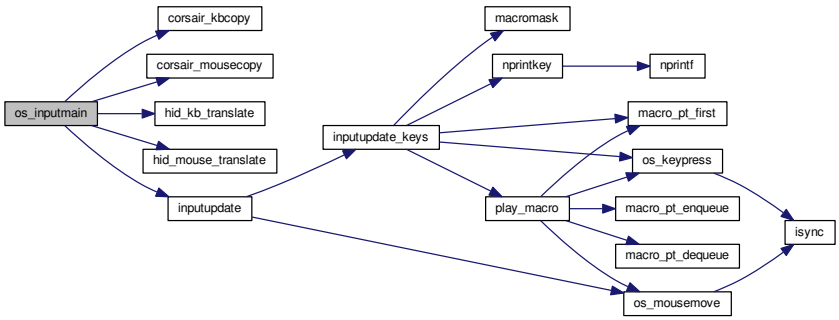
References usbdevice::active, ckb_err, ckb_info, corsair_kbcopy(), corsair_mousecopy(), devpath, usbdevice::epcount, usbdevice::fwversion, usbdevice::handle, hid_kb_translate(), hid_mouse_translate(), imutex, INDEX_OF, usbdevice::input, inputupdate(), IS_MOUSE, IS_RGB, keyboard, usbinput::keys, MSG_SIZE, usbdevice::product, usbinput::rel_x, usbinput::rel_y, and usbdevice::vendor.

Referenced by _setupusb().

```
248                                 {
249     usbdevice* kb = context;
250     int fd = kb->handle - 1;
251     short vendor = kb->vendor, product = kb->product;
252     int index = INDEX_OF(kb, keyboard);
253     ckb_info("Starting input thread for %s%d\n", devpath, index);
254 
259     int urbcount = IS_RGB(vendor, product) ? (kb->epcount - 1) : kb->
    epcount;
260     if (urbcount == 0) {
261         ckb_err("urbcount = 0, so there is nothing to claim in os_inputmain()\n");
262         return 0;
263     }
264 
266     struct usbdevfs_urb urbs[urbcount + 1];
267     memset(urbs, 0, sizeof(urbs));
268 
282     urbs[0].buffer_length = (kb->fwversion >= 0x300 ? MSG_SIZE : 8);
283     if(urbcount > 1 && IS_RGB(vendor, product)) {
284         if(IS_MOUSE(vendor, product))
285             urbs[1].buffer_length = 10;
286         else
287             urbs[1].buffer_length = 21;
288         urbs[2].buffer_length = MSG_SIZE;
289         if(urbcount != 3)
290             urbs[urbcount - 1].buffer_length = MSG_SIZE;
291     } else if(kb->fwversion < 0x300) {
292             urbs[1].buffer_length = 4;
293             urbs[2].buffer_length = 15;
294     }
295 
298     for(int i = 0; i < urbcount; i++){
299         urbs[i].type = USBDEVFS_URB_TYPE_INTERRUPT;
300         urbs[i].endpoint = 0x80 | (i + 1);
301         urbs[i].buffer = malloc(urbs[i].buffer_length);
302         ioctl(fd, USBDEVFS_SUBMITURB, urbs + i);
303     }
304 
306     while (1) {
307         struct usbdevfs_urb* urb = 0;
308 
311         if (ioctl(fd, USBDEVFS_REAPURB, &urb)) {
312             if (errno == ENODEV || errno == ENOENT || errno == ESHUTDOWN)
313                 // Stop the thread if the handle closes
314                 break;
315             else if(errno == EPIPE && urb){
317                 ioctl(fd, USBDEVFS_CLEAR_HALT, &urb->endpoint);
318                 // Re-submit the URB
319                 if(urb)
```

```
320                     ioctl(fd, USBDEVFS_SUBMITURB, urb);
321                 urb = 0;
322             }
323             continue;
324         }
325 
329         if (urb) {
330 
342             pthread_mutex_lock(imutex(kb));
343             // EP workaround for FWv3
344             // Corsair input comes through 0x81, but case 1 in keymap.c is used for 6KRO
345             uchar urbendpoint = (kb->fwversion >= 0x300 ? 2 : (urb->endpoint & 0xF));
346             if(IS_MOUSE(vendor, product)){
347                 switch(urb->actual_length){
348                 case 8:
349                 case 10:
350                 case 11:
351                     // HID mouse input
352                     hid_mouse_translate(kb->input.keys, &kb->
    input.rel_x, &kb->input.rel_y, -urbendpoint, urb->actual_length, urb->buffer, kb->
    fwversion);
353                     break;
354                 case MSG_SIZE:
355                     // Corsair mouse input
356                     corsair_mousecopy(kb->input.keys, -urbendpoint, urb->buffer);
357                     break;
358                 }
359             } else if(IS_RGB(vendor, product)){
360                 switch(urb->actual_length){
361                 case 8:
362                     // RGB EP 1: 6KRO (BIOS mode) input
363                     hid_kb_translate(kb->input.keys, -1, urb->actual_length, urb->
    buffer);
364                     break;
365                 case 21:
366                 case 5:
367                     // RGB EP 2: NKRO (non-BIOS) input. Accept only if keyboard is inactive
368                     if(!kb->active)
369                         hid_kb_translate(kb->input.keys, -2, urb->actual_length,
    urb->buffer);
370                     break;
371                 case MSG_SIZE:
372                     // RGB EP 3: Corsair input
373                     corsair_kbcopy(kb->input.keys, -urbendpoint, urb->buffer);
374                     break;
375                 }
376             } else {
377                 // Non-RGB input
378                 hid_kb_translate(kb->input.keys, urb->endpoint & 0xF, urb->
    actual_length, urb->buffer);
379             }
382             inputupdate(kb);
383             pthread_mutex_unlock(imutex(kb));
384 
386             ioctl(fd, USBDEVFS_SUBMITURB, urb);
387             urb = 0;
388         }
389     }
390 
394     ckb_info("Stopping input thread for %s%d\n", devpath, index);
395     for(int i = 0; i < urbcount; i++){
396         ioctl(fd, USBDEVFS_DISCARDURB, urbs + i);
397         free(urbs[i].buffer);
398     }
399     return 0;
400 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.50.3.4 int os_resetusb (usbdevice * *kb,* const char * *file,* int *line*)

os_resetusb is the os specific implementation for resetting usb

Try to reset an usb device in a linux user space driver.

1. unclaim the device, but do not reconnect the system driver (second param resetting = true)
2. reset the device via USBDEVFS_RESET command
3. claim the device again. Returns 0 on success, -2 if device should be removed and -1 if reset should by tried again

Todo it seems that no one wants to try the reset again. But I'v seen it somewhere...

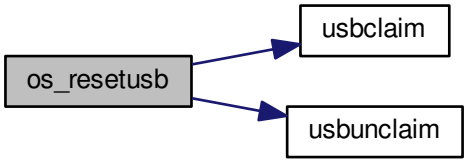
Definition at line 510 of file usb_linux.c.

References usbdevice::handle, TEST_RESET, usbclaim(), and usbunclaim().

Referenced by _resetusb().

```
510                                                     {
511     TEST_RESET(usbunclaim(kb, 1));
512     TEST_RESET(ioctl(kb->handle - 1, USBDEVFS_RESET));
513     TEST_RESET(usbclaim(kb));
514     // Success!
515     return 0;
516 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.50.3.5 void os_sendindicators (usbdevice * *kb*)

os_sendindicators update the indicators for the special keys (Numlock, Capslock and what else?)

os_sendindicators update the indicators for the special keys (Numlock, Capslock and what else?)

Read the data from kb->ileds ans send them via ioctl() to the keyboard.

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0x21	0x09	0x0200	Interface 0	MSG_SIZE 1 Byte	timeout 0,5ms	the message buffer pointer
Host to Device, Type=Class, Recipi-ent=Interface (why not endpoint?)	9 = SEND?	specific	0	1	500	struct* kb->ileds

The ioctl command is USBDEVFS_CONTROL.

Definition at line 214 of file usb_linux.c.

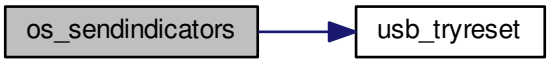
References ckb_err, usbdevice::fwversion, usbdevice::handle, usbdevice::ileds, and usb_tryreset().

Referenced by updateindicators_kb().

```
214                                     {
215     static int countForReset = 0;
216     void *ileds;
217     ushort leds;
218     if(kb->fwversion >= 0x300) {
219         leds = (kb->ileds << 8) | 0x0001;
```

```
220         ileds = &leds;
221     }
222     else {
223         ileds = &kb->ileds;
224     }
225     struct usbdevfs_ctrltransfer transfer = { 0x21, 0x09, 0x0200, 0x00, (kb->
    fwversion >= 0x300 ? 2 : 1), 500, ileds };
226     int res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
227     if(res <= 0) {
228         ckb_err("%s\n", res ? strerror(errno) : "No data written");
229         if (usb_tryreset(kb) == 0 && countForReset++ < 3) {
230             os_sendindicators(kb);
231         }
232     }
233 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.50.3.6 int os_setupusb (usbdevice * *kb*)

os_setupusb OS-specific setup for a specific usb device.

Perform the operating system-specific opening of the interface in os_setupusb(). As a result, some parameters should be set in kb (name, serial, fwversion, epcount = number of usb endpoints), and all endpoints should be claimed with usbclaim(). Claiming is the only point where os_setupusb() can produce an error (-1).

- Copy device description and serial
- Copy firmware version (needed to determine USB protocol)
- Do some output about connecting interfaces
- Claim the USB interfaces

Todo in these modules a pullrequest is outstanding

< Try to reset the device and recall the function

< Don't do this endless in recursion

< os_setupusb() has a return value (used as boolean)

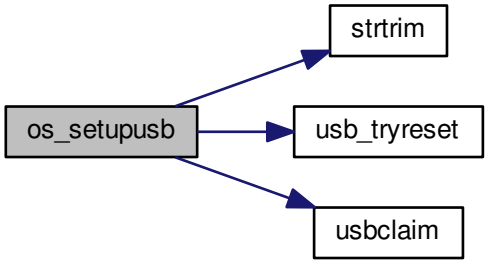
Definition at line 548 of file usb_linux.c.

References ckb_err, ckb_info, devpath, usbdevice::epcount, usbdevice::fwversion, INDEX_OF, KB_NAME_LEN, keyboard, usbdevice::name, usbdevice::serial, SERIAL_LEN, strtrim(), usbdevice::udev, usb_tryreset(), and usb-claim().

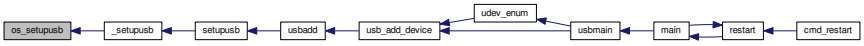
Referenced by _setupusb().

```
548                                {
551     struct udev_device* dev = kb->udev;
552     const char* name = udev_device_get_sysattr_value(dev, "product");
553     if(name)
554         strncpy(kb->name, name, KB_NAME_LEN);
555     strtrim(kb->name);
556     const char* serial = udev_device_get_sysattr_value(dev, "serial");
557     if(serial)
558         strncpy(kb->serial, serial, SERIAL_LEN);
559     strtrim(kb->serial);
562     const char* firmware = udev_device_get_sysattr_value(dev, "bcdDevice");
563     if(firmware)
564         sscanf(firmware, "%hx", &kb->fwversion);
565     else
566         kb->fwversion = 0;
567     int index = INDEX_OF(kb, keyboard);
568
570     ckb_info("Connecting %s at %s%d\n", kb->name, devpath, index);
571
577     const char* ep_str = udev_device_get_sysattr_value(dev, "bNumInterfaces");
578 #ifdef DEBUG
579     ckb_info("claiming interfaces. name=%s, firmware=%s; ep_str=%s\n", name, firmware, ep_str);
580 #endif //DEBUG
581     kb->epcount = 0;
582     if(ep_str)
583         sscanf(ep_str, "%d", &kb->epcount);
584     if(kb->epcount < 2){
585         // IF we have an RGB KB with 0 or 1 endpoints, it will be in BIOS mode.
586         ckb_err("Unable to read endpoint count from udev, assuming %d and reading >>%s<< or device
    is in BIOS mode\n", kb->epcount, ep_str);
587         if (usb_tryreset(kb) == 0) {
588             static int retryCount = 0;
589             if (retryCount++ < 5) {
590                 return os_setupusb(kb);
591             }
592         }
593         return -1;
594         // ToDo are there special versions we have to detect? If there are, that was the old code to handle
    it:
595         // This shouldn't happen, but if it does, assume EP count based onckb_warn what the device is
    supposed to have
596         // kb->epcount = (HAS_FEATURES(kb, FEAT_RGB) ? 4 : 3);
597         // ckb_warn("Unable to read endpoint count from udev, assuming %d and reading >>%s<<...\n",
    kb->epcount, ep_str);
598     }
599     if(usbclaim(kb)){
600         ckb_err("Failed to claim interfaces: %s\n", strerror(errno));
601         return -1;
602     }
603     return 0;
604 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.50.3.7 int os_usbrecv (usbdevice * *kb*, uchar * *in_msg*, const char * *file*, int *line*)

os_usbrecv does what its name says:

The comment at the beginning of the procedure causes the suspicion that the firmware versionspecific distinction is missing for receiving from usb endpoint 3 or 4. The commented code contains only the reception from EP4, but this may be wrong for a software version 2.0 or higher (see the code for os-usbsend ()).

So all the receiving is done via an ioctl() like in os_usbsend. The ioctl() is given a struct usbdevfs_ctrltransfer, in which the relevant parameters are entered:

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0xA1	0x01	0x0200	endpoint to be addressed from epcount - 1	MSG_SIZE	5ms	the message buffer pointer
Device to Host, Type=Class, Recipi-ent=Interface	1 = RECEIVE?	specific	Interface #	64	5000	in_msg

The ioctl() returns the number of bytes received. Here is the usual check again:

- If the return value is -1 AND the error is a timeout (ETIMEOUT), os_usbrecv() will return -1 to indicate that it is probably a recoverable problem and a retry is recommended.
- For another negative value or other error identifier OR 0 bytes are received, 0 is returned as an identifier for a heavy error.
- In all other cases, the function returns the number of bytes received.

If this is not the entire blocksize (MSG_SIZE bytes), an error message is issued on the standard error channel [warning "Read YY bytes (expected 64)"].

Definition at line 130 of file usb_linux.c.

References ckb_err_fn, ckb_warn_fn, usbdevice::epcount, usbdevice::handle, and MSG_SIZE.

Referenced by _usbrecv().

```
130                                                                         {
131     int res;
132     // This is what CUE does, but it doesn't seem to work on linux.
133     /*if(kb->fwversion >= 0x130){
134         struct usbdevfs_bulktransfer transfer = {0};
135         transfer.ep = 0x84;
136         transfer.len = MSG_SIZE;
137         transfer.timeout = 5000;
138         transfer.data = in_msg;
139         res = ioctl(kb->handle - 1, USBDEVFS_BULK, &transfer);
140     } else {*/
141         struct usbdevfs_ctrltransfer transfer = { 0xa1, 0x01, 0x0300, kb->
    epcount - 1, MSG_SIZE, 5000, in_msg };
142         res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
143     //}
144     if(res <= 0){
145         ckb_err_fn("%s\n", file, line, res ? strerror(errno) : "No data read");
146         if(res == -1 && errno == ETIMEDOUT)
147             return -1;
148         else
149             return 0;
150     } else if(res != MSG_SIZE)
151         ckb_warn_fn("Read %d bytes (expected %d)\n", file, line, res,
    MSG_SIZE);
152 #ifdef DEBUG_USB_RECV
153     char converted[MSG_SIZE*3 + 1];
154     for(int i=0;i<MSG_SIZE;i++)
155         sprintf(&converted[i*3], "%02x ", in_msg[i]);
156     ckb_warn_fn("Recv %s\n", file, line, converted);
157 #endif
158     return res;
159 }
```

Here is the caller graph for this function:

13.50.3.8 int os_usbsend (usbdevice * *kb*, const uchar * *out_msg*, int *is_recv*, const char * *file*, int *line*)

os_usbsend has two functions:

- if is_recv == false, it tries to send a given MSG_SIZE buffer via the usb interface given with kb.
- otherwise a request is sent via the usb device to initiate the receiving of a message from the remote device.

The functionality for sending distinguishes two cases, depending on the version number of the firmware of the connected device:

If the firmware is less or equal 1.2, the transmission is done via an ioctl(). The ioctl() is given a struct usbdevfs_ctrltransfer, in which the relevant parameters are entered:

bRequest-Type	bRequest	wValue	EP	size	Timeout	data
0x21	0x09	0x0200	endpoint / IF to be addressed from epcount-1	MSG_SIZE	5000 (=5ms)	the message buffer pointer
Host to Device, Type=Class, Recipi-ent=Interface	9 = Send data?	specific	last or pre-last device #	64	5000	out_msg

The ioctl command is USBDEVFS_CONTROL.

The same constellation is used if the device is requested to send its data (is_recv = true).

For a more recent firmware and is_recv = false, the ioctl command USBDEVFS_CONTROL is not used (this tells the bus to enter the control mode), but the bulk method is used: USBDEVFS_BULK. This is astonishing, because all of the endpoints are type Interrupt, not bulk.

Anyhow, forthis purpose a different structure is used for the ioctl() (struct **usbdevfs_bulktransfer**) and this is also initialized differently:

The length and timeout parameters are given the same values as above. The formal parameter out_msg is also passed as a buffer pointer. For the endpoints, the firmware version is differentiated again:

For a firmware version between 1.3 and <2.0 endpoint 4 is used, otherwise (it can only be >=2.0) endpoint 3 is used.

Todo Since the handling of endpoints has already led to problems elsewhere, this implementation is extremely hardware-dependent and critical!

Eg. the new keyboard K95PLATINUMRGB has a version number significantly less than 2.0 - will it run with this implementation?

The ioctl() - no matter what type - returns the number of bytes sent. Now comes the usual check:

- If the return value is -1 AND the error is a timeout (ETIMEOUT), os_usbsend() will return -1 to indicate that it is probably a recoverable problem and a retry is recommended.
- For another negative value or other error identifier OR 0 bytes sent, 0 is returned as a heavy error identifier.
- In all other cases, the function returns the number of bytes sent.

If this is not the entire blocksize (MSG_SIZE bytes), an error message is issued on the standard error channel [warning "Wrote YY bytes (expected 64)"].

If DEBUG_USB_SEND is set during compilation, the number of bytes sent and their representation are logged to the error channel.

Definition at line 68 of file usb_linux.c.

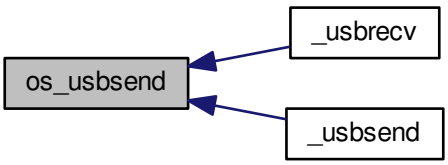
References ckb_err_fn, ckb_warn_fn, usbdevice::epcount, usbdevice::fwversion, usbdevice::handle, IS_V2_OVERRIDE, and MSG_SIZE.

Referenced by _usbrecv(), and _usbsend().

```
68                                                                                  {
69     int res;
70     if ((kb->fwversion >= 0x120 || IS_V2_OVERRIDE(kb)) && !is_recv){
71         struct usbdevfs_bulktransfer transfer = {0};
72         // FW 2.XX uses 0x03, FW 3.XX uses 0x02
73         transfer.ep = (kb->fwversion >= 0x130 && kb->fwversion < 0x200) ? 4 : (kb->
       fwversion >= 0x300 ? 2 : 3);
74         transfer.len = MSG_SIZE;
75         transfer.timeout = 5000;
```

```
76          transfer.data = (void*)out_msg;
77          res = ioctl(kb->handle - 1, USBDEVFS_BULK, &transfer);
78      } else {
79          struct usbdevfs_ctrltransfer transfer = { 0x21, 0x09, 0x0200, kb->
    epcount - 1, MSG_SIZE, 5000, (void*)out_msg };
80          res = ioctl(kb->handle - 1, USBDEVFS_CONTROL, &transfer);
81      }
82
83      if (res <= 0){
84          ckb_err_fn(" %s, res = 0x%x\n", file, line, res ? strerror(errno) : "No data written",
    res);
85          if(res == -1 && errno == ETIMEDOUT)
86              return -1;
87          else
88              return 0;
89      } else if (res != MSG_SIZE)
90          ckb_warn_fn("Wrote %d bytes (expected %d)\n", file, line, res,
    MSG_SIZE);
91 #ifdef DEBUG_USB_SEND
92      char converted[MSG_SIZE*3 + 1];
93      for(int i=0;i<MSG_SIZE;i++)
94          sprintf(&converted[i*3], "%02x ", out_msg[i]);
95      ckb_warn_fn("Sent %s\n", file, line, converted);
96 #endif
97      return res;
98 }
```

Here is the caller graph for this function:



13.50.3.9 void strtrim (char * *string*)

strtrim trims a string by removing leading and trailing spaces.

Parameters

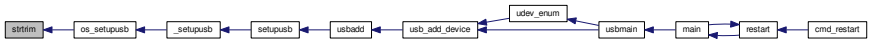
string	

Definition at line 523 of file usb_linux.c.

Referenced by os_setupusb().

```
523                         {
524     // Find last non-space
525     char* last = string;
526     for(char* c = string; *c != 0; c++){
527         if(!isspace(*c))
528             last = c;
529     }
530     last[1] = 0;
531     // Find first non-space
532     char* first = string;
533     for(; *first != 0; first++){
534         if(!isspace(*first))
535             break;
536     }
537     if(first != string)
538         memmove(string, first, last - first);
539 }
```

Here is the caller graph for this function:



13.50.3.10 static void udev_enum ( ) [static]

Reduce the hits of the enumeration by limiting to usb as technology and corsair as idVendor. Then filter with udev-_enumerate_scan_devices () all hits.

The following call to udev_enumerate_get_list_entry() fetches the entire hitlist as udev_list_entry ∗.

Use udev_list_entry_foreach() to iterate through the hit set.

If both the device name exists (udev_list_entry_get_name) and the subsequent creation of a new udev_device (udev_device_new_from_syspath) is ok, the new device is added to the list with usb_add_device().

If the latter does not work, the new device is released again (udev_device_unref ()).

After the last iteration, the enumerator is released with udev_enumerate_unref ().

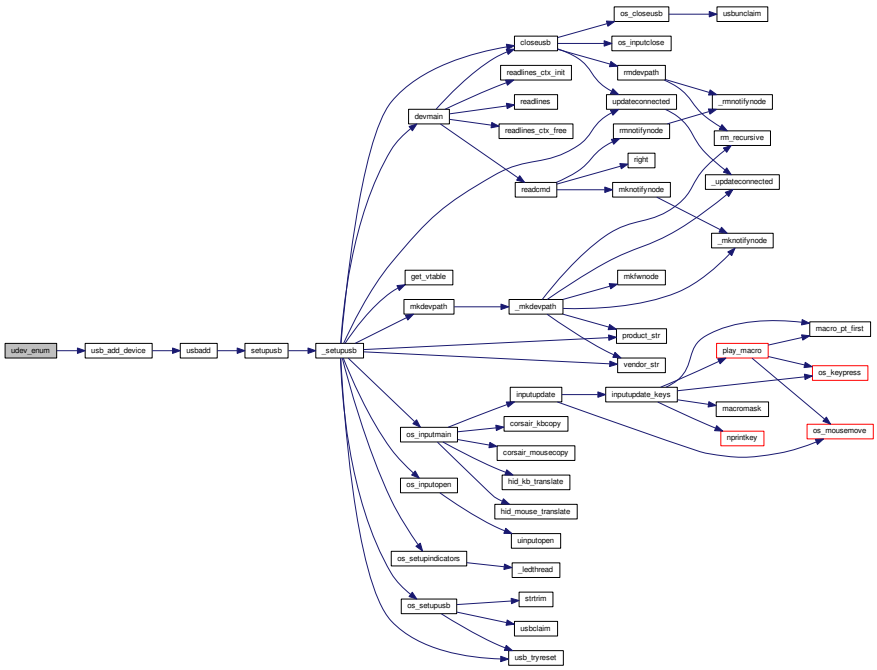
Definition at line 765 of file usb_linux.c.

References usb_add_device(), and V_CORSAIR_STR.

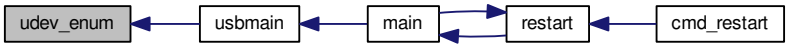
Referenced by usbmain().

```
765                         {
766     struct udev_enumerate* enumerator = udev_enumerate_new(udev);
767     udev_enumerate_add_match_subsystem(enumerator, "usb");
768     udev_enumerate_add_match_sysattr(enumerator, "idVendor", V_CORSAIR_STR);
769     udev_enumerate_scan_devices(enumerator);
770     struct udev_list_entry* devices, *dev_list_entry;
771     devices = udev_enumerate_get_list_entry(enumerator);
772
773     udev_list_entry_foreach(dev_list_entry, devices){
774         const char* path = udev_list_entry_get_name(dev_list_entry);
775         if(!path)
776             continue;
777         struct udev_device* dev = udev_device_new_from_syspath(udev, path);
778         if(!dev)
779             continue;
780         // If the device matches a recognized device ID, open it
781         if(usb_add_device(dev))
782             // Release device if not
783             udev_device_unref(dev);
784     }
785     udev_enumerate_unref(enumerator);
786 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.50.3.11 static int usb_add_device (struct udev_device * *dev*) [static]

If the device id can be found, call usbadd() with the appropriate parameters.

Parameters

dev	the functions usb_*_device get a struct udev* with the neccessary hardware-related information.

Returns

the retval of usbadd() or 1 if either vendor is not corsair or product is not mentioned in model[].

First get the idVendor via udev_device_get_sysattr_value(). If this is equal to the ID-string of corsair ("1b1c"), get the idProduct on the same way.

If we can find the model name in the model array, call usbadd() with the model number.

Todo So why the hell not a transformation between the string and the short presentation? Lets check if the string representation is used elsewhere.

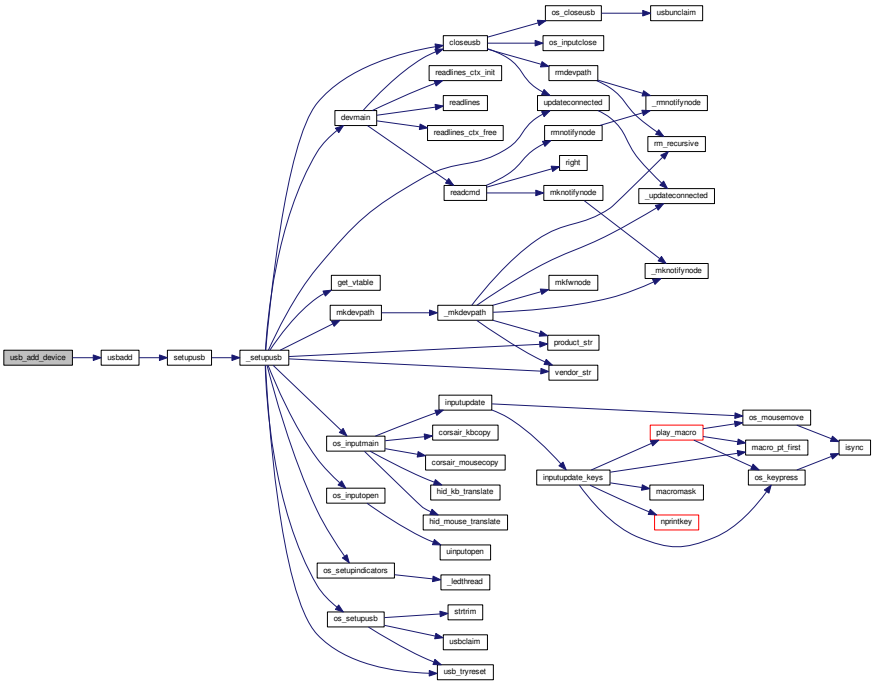
Definition at line 713 of file usb_linux.c.

References N_MODELS, usbadd(), V_CORSAIR, and V_CORSAIR_STR.

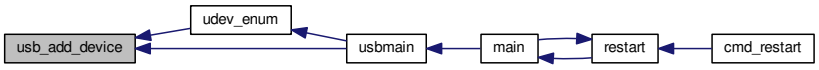
Referenced by udev_enum(), and usbmain().

```
713                                                 {
714     const char* vendor = udev_device_get_sysattr_value(dev, "idVendor");
715     if(vendor && !strcmp(vendor, V_CORSAIR_STR)){
716         const char* product = udev_device_get_sysattr_value(dev, "idProduct");
717         if(product){
718             for(_model* model = models; model < models +
      N_MODELS; model++){
719                 if(!strcmp(product, model->name)){
720                     return usbadd(dev, V_CORSAIR, model->number);
721                 }
722             }
723         }
724     }
725     return 1;
726 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.50.3.12 static void usb_rm_device (struct udev_device * *dev*) [static]

Parameters

dev	the functions usb_*_device get a struct udev* with the neccessary hardware-related information.

First try to find the system path of the device given in parameter dev. The index where the name is found is the same index we need to address the global keyboard array. That array holds all usbdevices.

Searching for the correct name in kbsyspath-array and closing the usb via closeusb() are protected by lock..unlock of the corresponding devmutex arraymember.

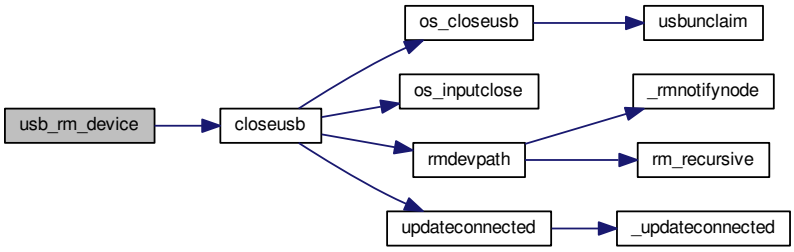
Definition at line 738 of file usb_linux.c.

References closeusb(), DEV_MAX, devmutex, kbsyspath, and keyboard.

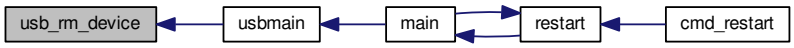
Referenced by usbmain().

```
738                                                 {
739     // Device removed. Look for it in our list of keyboards
740     const char* syspath = udev_device_get_syspath(dev);
741     if(!syspath || syspath[0] == 0)
742         return;
743     for(int i = 1; i < DEV_MAX; i++){
744         pthread_mutex_lock(devmutex + i);
745         if(!strcmp(syspath, kbsyspath[i]))
746             closeusb(keyboard + i);
747         pthread_mutex_unlock(devmutex + i);
748     }
749 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.50.3.13 int usbadd (struct udev_device * *dev*, short *vendor*, short *product*)

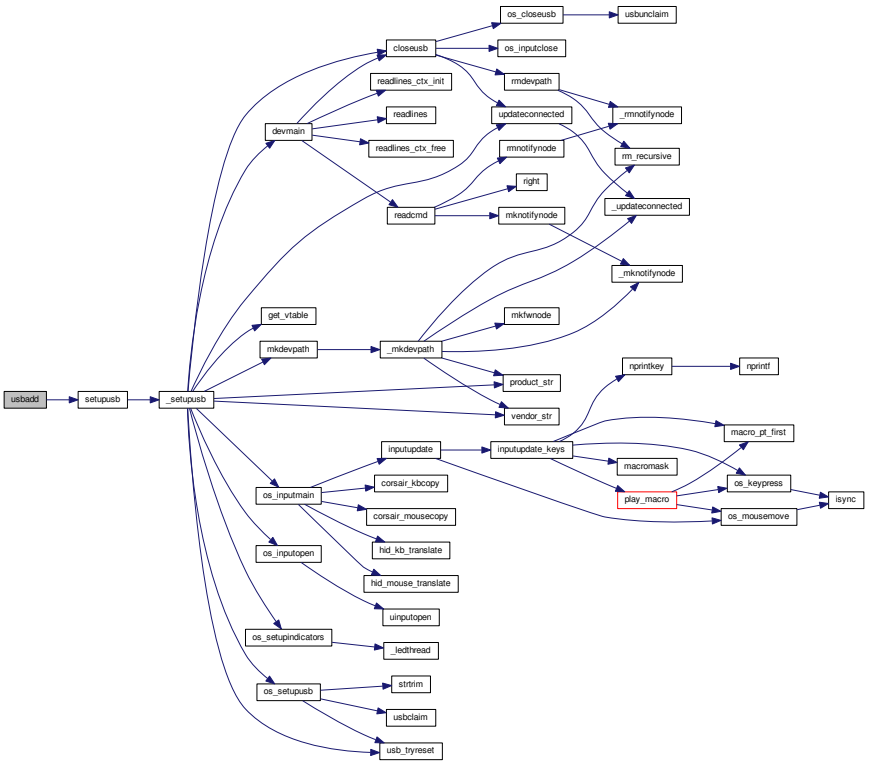
Definition at line 606 of file usb_linux.c.

References ckb_err, ckb_info, DEV_MAX, dmutex, usbdevice::handle, IS_CONNECTED, kbsyspath, keyboard, usbdevice::product, setupusb(), usbdevice::udev, and usbdevice::vendor.

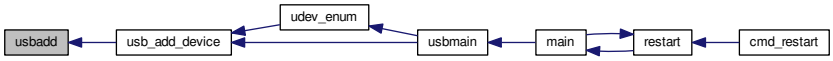
Referenced by usb_add_device().

```
606                                                                     {
607     const char* path = udev_device_get_devnode(dev);
608     const char* syspath = udev_device_get_syspath(dev);
609     if(!path || !syspath || path[0] == 0 || syspath[0] == 0){
610         ckb_err("Failed to get device path\n");
611         return -1;
612     }
613 #ifdef DEBUG
614     ckb_info(">>>vendor = 0x%x, product = 0x%x, path = %s, syspath = %s\n", vendor, product, path,
    syspath);
615 #endif // DEDBUG
616     // Find a free USB slot
617     for(int index = 1; index < DEV_MAX; index++){
618         usbdevice* kb = keyboard + index;
619         if(pthread_mutex_trylock(dmutex(kb))){
620             // If the mutex is locked then the device is obviously in use, so keep going
621             if(!strcmp(syspath, kbsyspath[index])){
622                 // Make sure this existing keyboard doesn't have the same syspath (this shouldn't happen)
623                 return 0;
624             }
625             continue;
626         }
627         if(!IS_CONNECTED(kb)){
628             // Open the sysfs device
629             kb->handle = open(path, O_RDWR) + 1;
630             if(kb->handle <= 0){
631                 ckb_err("Failed to open USB device: %s\n", strerror(errno));
632                 kb->handle = 0;
633                 pthread_mutex_unlock(dmutex(kb));
634                 return -1;
635             } else {
636                 // Set up device
637                 kb->udev = dev;
638                 kb->vendor = vendor;
639                 kb->product = product;
640                 strncpy(kbsyspath[index], syspath, FILENAME_MAX);
641                 // Mutex remains locked
642                 setupusb(kb);
643                 return 0;
644             }
645         }
646         pthread_mutex_unlock(dmutex(kb));
647     }
648     ckb_err("No free devices\n");
649     return -1;
650 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.50.3.14 static int usbclaim (usbdevice * kb) [static]

usbclaim does claiming all EPs for the usb device gicen by kb.

Parameters

kb	THE usbdevice*

Returns

0 on success, -1 otherwise.

Claim all endpoints for a given device (remeber the decrementing of the file descriptor) via ioctl(USBDEVFS_DISCONNECT) and ioctl(USBDEVFS_CLAIMINTERFACE).

Error handling is done for the ioctl(USBDEVFS_CLAIMINTERFACE) only. If this fails, now an error message is thrown and -1 is returned. Function is called in usb_linux.c only, so it is declared as static now.

Definition at line 472 of file usb_linux.c.

References ckb_err, ckb_info, usbdevice::epcount, and usbdevice::handle.

Referenced by os_resetusb(), and os_setupusb().

```
472                                 {
473     int count = kb->epcount;
474 #ifdef DEBUG
475     ckb_info("claiming %d endpoints\n", count);
476 #endif // DEBUG
477
478     for(int i = 0; i < count; i++){
479         struct usbdevfs_ioctl ctl = { i, USBDEVFS_DISCONNECT, 0 };
480         ioctl(kb->handle - 1, USBDEVFS_IOCTL, &ctl);
481         if(ioctl(kb->handle - 1, USBDEVFS_CLAIMINTERFACE, &i)) {
482             ckb_err("Failed to claim interface %d: %s\n", i, strerror(errno));
483             return -1;
484         }
485     }
486     return 0;
487 }
```

Here is the caller graph for this function:

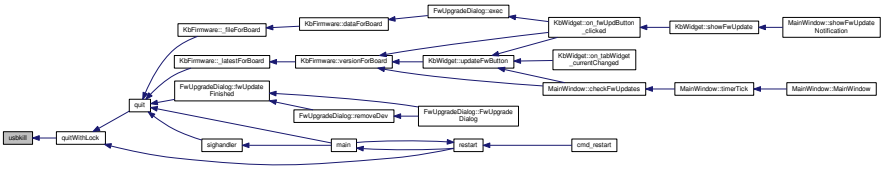
13.50.3.15 void usbkill ( )

Definition at line 853 of file usb_linux.c.

Referenced by quitWithLock().

```
853                 {
854     udev_unref(udev);
855     udev = 0;
856 }
```

Here is the caller graph for this function:



13.50.3.16 int usbmain ( )

Start the USB main loop. Returns program exit code when finished.

usbmain is called by main() after setting up all other stuff.

Returns

0 normally or -1 if fatal error occurs (up to now only if no new devices are available)

First check whether the uinput module is loaded by the kernel.

Todo Why isn't missing of uinput a fatal error?

Create the udev object with udev_new() (is a function from libudev.h) terminate -1 if error

Enumerate all currently connected devices

Todo lae. here the work has to go on...

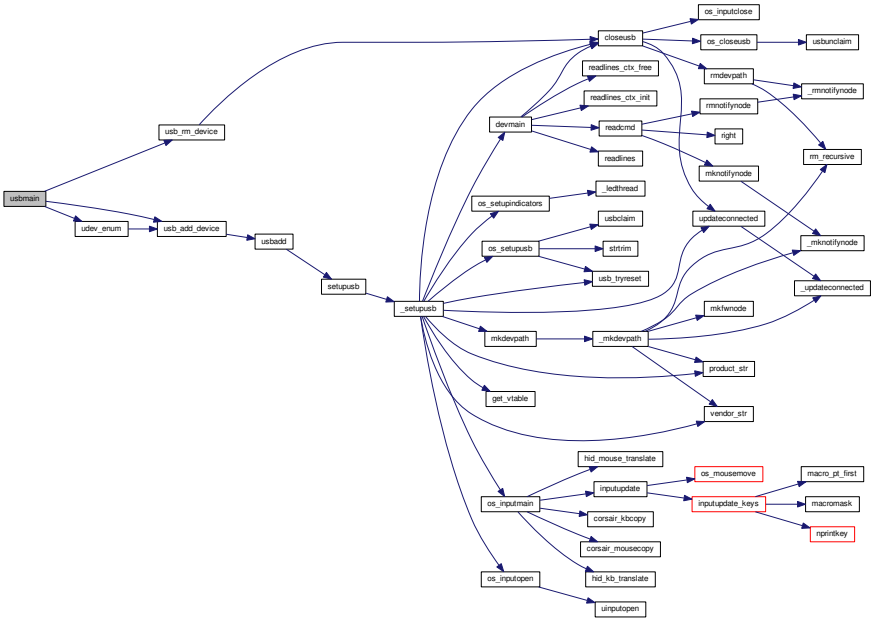
Definition at line 793 of file usb_linux.c.

References ckb_fatal, ckb_warn, udev_enum(), usb_add_device(), and usb_rm_device().

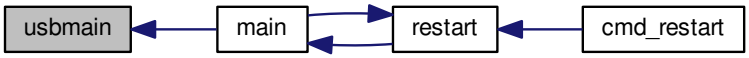
Referenced by main().

```
793             {
798     // Load the uinput module (if it's not loaded already)
799     if(system("modprobe uinput") != 0)
800         ckb_warn("Failed to load uinput module\n");
801 
805     if(!(udev = udev_new())) {
806         ckb_fatal("Failed to initialize udev in usbmain(), usb_linux.c\n");
807         return -1;
808     }
809 
812     udev_enum();
813 
816     // Done scanning. Enter a loop to poll for device updates
817     struct udev_monitor* monitor = udev_monitor_new_from_netlink(udev, "udev");
818     udev_monitor_filter_add_match_subsystem_devtype(monitor, "usb", 0);
819     udev_monitor_enable_receiving(monitor);
820     // Get an fd for the monitor
821     int fd = udev_monitor_get_fd(monitor);
822     fd_set fds;
823     while(udev){
824         FD_ZERO(&fds);
825         FD_SET(fd, &fds);
826         // Block until an event is read
827         if(select(fd + 1, &fds, 0, 0, 0) > 0 && FD_ISSET(fd, &fds)){
828             struct udev_device* dev = udev_monitor_receive_device(monitor);
829             if(!dev)
830                 continue;
831             const char* action = udev_device_get_action(dev);
832             if(!action){
833                 udev_device_unref(dev);
834                 continue;
835             }
836             // Add/remove device
837             if(!strcmp(action, "add")){
838                 int res = usb_add_device(dev);
839                 if(res == 0)
840                     continue;
841                 // If the device matched but the handle wasn't opened correctly, re-enumerate (this
        sometimes solves the problem)
842                 if(res == -1)
843                     udev_enum();
844             } else if(!strcmp(action, "remove"))
845                 usb_rm_device(dev);
846             udev_device_unref(dev);
847         }
848     }
849     udev_monitor_unref(monitor);
850     return 0;
851 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.50.3.17 static int usbunclaim (usbdevice * kb, int resetting) [static]

usbunclaim do an unclaiming of the usb device gicen by kb.

Parameters

kb	THE usbdevice*
resetting	boolean flag: If resseting is true, the caller will perform a bus reset command after unclaiming the device.

Returns

always 0.

Unclaim all endpoints for a given device (remeber the decrementing of the file descriptor) via ioctl(USBDEVFS_DISCARDURB).

Afterwards - if ressetting is false - do a USBDEVFS_CONNECT for EP 0 and 1. If it is a non RGB device, connect EP 2 also. The comment mentions RGB keyboards only, but as I understand the code, this is valid also for RGB mice.

There is no error handling yet. Function is called in usb_linux.c only, so it is declared as static now.

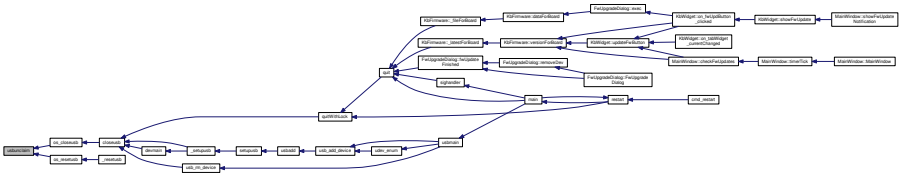
Definition at line 419 of file usb_linux.c.

References usbdevice::epcount, FEAT_RGB, usbdevice::handle, and HAS_FEATURES.

Referenced by os_closeusb(), and os_resetusb().

```
419                                                 {
420     int handle = kb->handle - 1;
421     int count = kb->epcount;
422     for (int i = 0; i < count; i++) {
423         ioctl(handle, USBDEVFS_RELEASEINTERFACE, &i);
424     }
425     // For RGB keyboards, the kernel driver should only be reconnected to interfaces 0 and 1 (HID), and
      only if we're not about to do a USB reset.
426     // Reconnecting any of the others causes trouble.
427     if (!resetting) {
428         struct usbdevfs_ioctl ctl = { 0, USBDEVFS_CONNECT, 0 };
429         ioctl(handle, USBDEVFS_IOCTL, &ctl);
430         ctl.ifno = 1;
431         ioctl(handle, USBDEVFS_IOCTL, &ctl);
432         // Also reconnect iface #2 (HID) for non-RGB keyboards
433         if(!HAS_FEATURES(kb, FEAT_RGB)){
434             ctl.ifno = 2;
435             ioctl(handle, USBDEVFS_IOCTL, &ctl);
436         }
437     }
438     return 0;
439 }
```

Here is the caller graph for this function:



13.50.4 Variable Documentation

13.50.4.1 char kbsyspath[9][FILENAME_MAX] [static]

Definition at line 13 of file usb_linux.c.

Referenced by os_closeusb(), usb_rm_device(), and usbadd().

13.50.4.2 _model models[] [static]

Attention

when adding new hardware this file hat to be changed too.

In this structure array *models*[] for each device the name (the device id as string in hex without leading 0x) and its usb device id as short must be entered in this array.

Definition at line 668 of file usb_linux.c.

13.50.4.3 struct udev* udev [static]

Definition at line 652 of file usb_linux.c.

13.50.4.4 pthread_t udevthread

Definition at line 655 of file usb_linux.c.

13.50.4.5 pthread_t usbthread

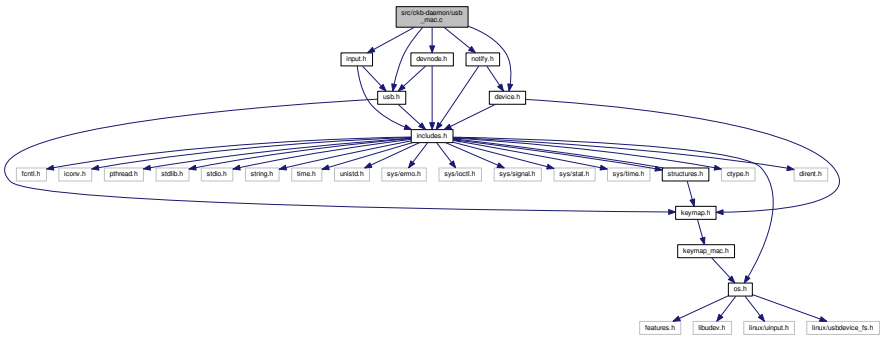
Todo These two thread vasriables seem to be unused: usbtread, udevthread

Definition at line 655 of file usb_linux.c.

13.51 src/ckb-daemon/usb_mac.c File Reference

```
#include "device.h"
#include "devnode.h"
#include "input.h"
#include "notify.h"
#include "usb.h"
```

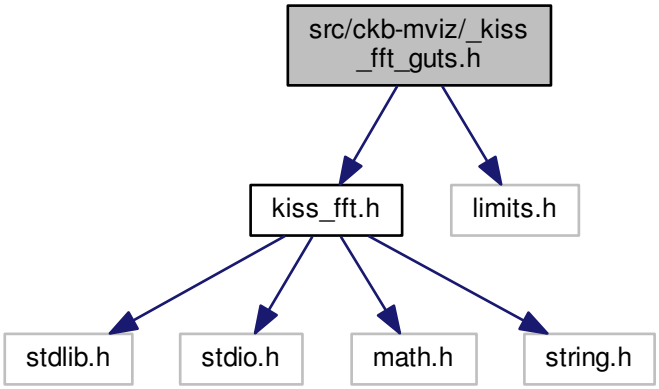
Include dependency graph for usb_mac.c:



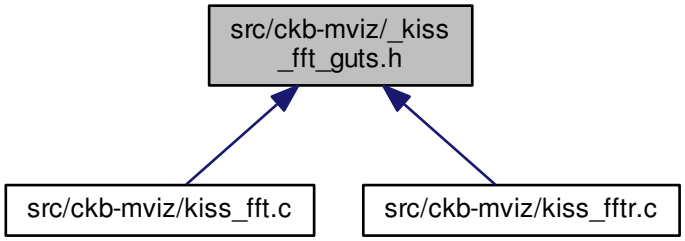
13.52 src/ckb-mviz/_kiss_fft_guts.h File Reference

```
#include "kiss_fft.h"
#include <limits.h>
```

Include dependency graph for _kiss_fft_guts.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct kiss_fft_state

Macros

- #define MAXFACTORS 32
- #define S_MUL(a, b) ((a)∗(b))
- #define C_MUL(m, a, b)
- #define C_FIXDIV(c, div) /∗ NOOP ∗/
- #define C_MULBYSCALAR(c, s)
- #define CHECK_OVERFLOW_OP(a, op, b) /∗ noop ∗/
- #define C_ADD(res, a, b)
- #define C_SUB(res, a, b)
- #define C_ADDTO(res, a)

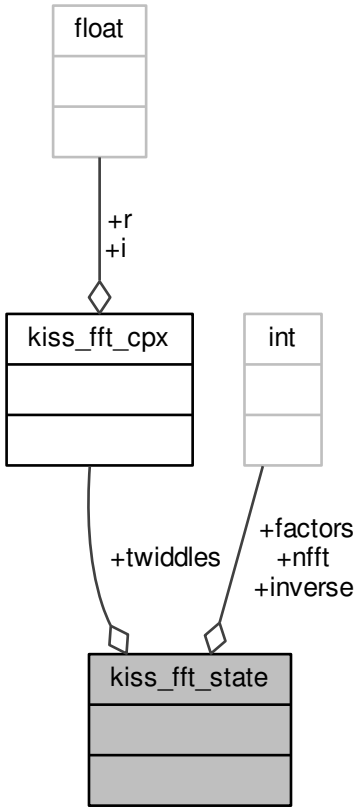
- #define C_SUBFROM(res, a)
- #define KISS_FFT_COS(phase) (kiss_fft_scalar) cos(phase)
- #define KISS_FFT_SIN(phase) (kiss_fft_scalar) sin(phase)
- #define HALF_OF(x) ((x)∗.5)
- #define kf_cexp(x, phase)
- #define pcpx(c) fprintf(stderr,"%g + %gi\n",(double)((c)->r),(double)((c)->i))
- #define KISS_FFT_TMP_ALLOC(nbytes) KISS_FFT_MALLOC(nbytes)
- #define KISS_FFT_TMP_FREE(ptr) KISS_FFT_FREE(ptr)

13.52.1 Data Structure Documentation

13.52.1.1 struct kiss_fft_state

Definition at line 28 of file _kiss_fft_guts.h.

Collaboration diagram for kiss_fft_state:



Data Fields

int	factors[2 ∗32]	
int	inverse	
int	nfft	
kiss_fft_cpx	twiddles[1]	

13.52.2 Macro Definition Documentation

13.52.2.1 #define C_ADD(*res, a, b*)

Value:

```
do { \
        CHECK_OVERFLOW_OP((a).r,+,(b).r)\
        CHECK_OVERFLOW_OP((a).i,+,(b).i)\
        (res).r=(a).r+(b).r;  (res).i=(a).i+(b).i; \
    }while(0)
```

Definition at line 100 of file _kiss_fft_guts.h.

Referenced by kf_bfly3(), kf_bfly4(), kf_bfly5(), kiss_fftr(), and kiss_fftri().

13.52.2.2 #define C_ADDTO(*res, a*)

Value:

```
do { \
        CHECK_OVERFLOW_OP((res).r,+,(a).r)\
        CHECK_OVERFLOW_OP((res).i,+,(a).i)\
        (res).r += (a).r;  (res).i += (a).i;\
    }while(0)
```

Definition at line 112 of file _kiss_fft_guts.h.

Referenced by kf_bfly2(), kf_bfly3(), kf_bfly4(), and kf_bfly_generic().

13.52.2.3 #define C_FIXDIV(*c, div*) /∗ NOOP ∗/

Definition at line 90 of file _kiss_fft_guts.h.

Referenced by kf_bfly2(), kf_bfly3(), kf_bfly4(), kf_bfly5(), kf_bfly_generic(), kiss_fftr(), and kiss_fftri().

13.52.2.4 #define C_MUL(*m, a, b*)

Value:

```
do{ (m).r = (a).r*(b).r - (a).i*(b).i;\
        (m).i = (a).r*(b).i + (a).i*(b).r; }while(0)
```

Definition at line 87 of file _kiss_fft_guts.h.

Referenced by kf_bfly2(), kf_bfly3(), kf_bfly4(), kf_bfly5(), kf_bfly_generic(), kiss_fftr(), and kiss_fftri().

13.52.2.5 #define C_MULBYSCALAR(*c, s*)

Value:

```
do{ (c).r *= (s);\
        (c).i *= (s); }while(0)
```

Definition at line 91 of file _kiss_fft_guts.h.

Referenced by kf_bfly3().

13.52.2.6 #define C_SUB(*res, a, b*)

Value:

```
do { \
        CHECK_OVERFLOW_OP((a).r,-,(b).r)\
        CHECK_OVERFLOW_OP((a).i,-,(b).i)\
        (res).r=(a).r-(b).r;  (res).i=(a).i-(b).i; \
    }while(0)
```

Definition at line 106 of file _kiss_fft_guts.h.

Referenced by kf_bfly2(), kf_bfly3(), kf_bfly4(), kf_bfly5(), kiss_fftr(), and kiss_fftri().

13.52.2.7 #define C_SUBFROM(*res, a*)

Value:

```
do {\
        CHECK_OVERFLOW_OP((res).r,-,(a).r)\
        CHECK_OVERFLOW_OP((res).i,-,(a).i)\
        (res).r -= (a).r;  (res).i -= (a).i; \
    }while(0)
```

Definition at line 119 of file _kiss_fft_guts.h.

13.52.2.8 #define CHECK_OVERFLOW_OP(*a, op, b*) /∗ noop ∗/

Definition at line 97 of file _kiss_fft_guts.h.

Referenced by kiss_fftr().

13.52.2.9 #define HALF_OF(x) ((x)∗.5)

Definition at line 138 of file _kiss_fft_guts.h.

Referenced by kf_bfly3(), and kiss_fftr().

13.52.2.10 #define kf_cexp(x, phase)

Value:

```
do{ \
        (x)->r = KISS_FFT_COS(phase);\
        (x)->i = KISS_FFT_SIN(phase);\
    }while(0)
```

Definition at line 141 of file _kiss_fft_guts.h.

Referenced by kiss_fft_alloc(), and kiss_fftr_alloc().

13.52.2.11 #define KISS_FFT_COS(phase) (kiss_fft_scalar) cos(phase)

Definition at line 136 of file _kiss_fft_guts.h.

13.52.2.12 #define KISS_FFT_SIN(phase) (kiss_fft_scalar) sin(phase)

Definition at line 137 of file _kiss_fft_guts.h.

13.52.2.13 #define KISS_FFT_TMP_ALLOC(*nbytes*) KISS_FFT_MALLOC(nbytes)

Definition at line 162 of file _kiss_fft_guts.h.

Referenced by kf_bfly_generic(), and kiss_fft_stride().

13.52.2.14 #define KISS_FFT_TMP_FREE(*ptr*) KISS_FFT_FREE(ptr)

Definition at line 163 of file _kiss_fft_guts.h.

Referenced by kf_bfly_generic(), and kiss_fft_stride().

13.52.2.15 #define MAXFACTORS 32

Definition at line 22 of file _kiss_fft_guts.h.

13.52.2.16 #define pcpx(*c*) fprintf(stderr,"%g + %gi\n",(double)((c)->r),(double)((c)->i))

Definition at line 149 of file _kiss_fft_guts.h.

13.52.2.17 #define S_MUL(*a*, *b*) ((a)*(b))

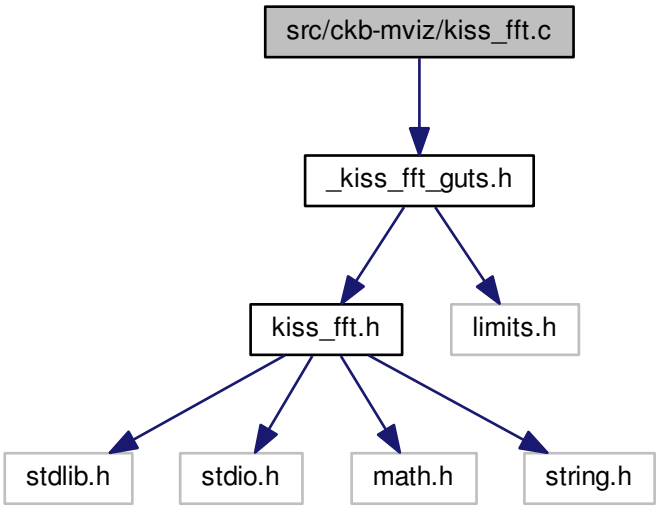
Definition at line 86 of file _kiss_fft_guts.h.

Referenced by kf_bfly5().

13.53 src/ckb-mviz/kiss_fft.c File Reference

```
#include "_kiss_fft_guts.h"
```

Include dependency graph for kiss_fft.c:



Functions

- static void kf_bfly2 (kiss_fft_cpx *Fout, const size_t fstride, const kiss_fft_cfg st, int m)
- static void kf_bfly4 (kiss_fft_cpx *Fout, const size_t fstride, const kiss_fft_cfg st, const size_t m)
- static void kf_bfly3 (kiss_fft_cpx *Fout, const size_t fstride, const kiss_fft_cfg st, size_t m)
- static void kf_bfly5 (kiss_fft_cpx *Fout, const size_t fstride, const kiss_fft_cfg st, int m)
- static void kf_bfly_generic (kiss_fft_cpx *Fout, const size_t fstride, const kiss_fft_cfg st, int m, int p)
- static void kf_work (kiss_fft_cpx *Fout, const kiss_fft_cpx *f, const size_t fstride, int in_stride, int *factors, const kiss_fft_cfg st)
- static void kf_factor (int n, int *facbuf)
- kiss_fft_cfg kiss_fft_alloc (int nfft, int inverse_fft, void *mem, size_t *lenmem)
- void kiss_fft_stride (kiss_fft_cfg st, const kiss_fft_cpx *fin, kiss_fft_cpx *fout, int in_stride)
- void kiss_fft (kiss_fft_cfg cfg, const kiss_fft_cpx *fin, kiss_fft_cpx *fout)
- void kiss_fft_cleanup (void)
- int kiss_fft_next_fast_size (int n)

13.53.1 Function Documentation

13.53.1.1 static void kf_bfly2 (kiss_fft_cpx * *Fout*, const size_t *fstride*, const kiss_fft_cfg *st*, int *m*) [static]

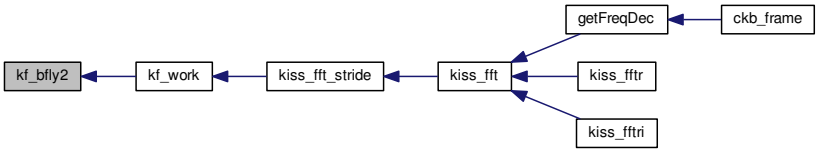
Definition at line 21 of file kiss_fft.c.

References C_ADDTO, C_FIXDIV, C_MUL, C_SUB, and kiss_fft_state::twiddles.

Referenced by kf_work().

```
27 {
28     kiss_fft_cpx * Fout2;
29     kiss_fft_cpx * tw1 = st->twiddles;
30     kiss_fft_cpx t;
31     Fout2 = Fout + m;
32     do{
33         C_FIXDIV(*Fout,2); C_FIXDIV(*Fout2,2);
34
35         C_MUL (t,  *Fout2 , *tw1);
36         tw1 += fstride;
37         C_SUB( *Fout2 ,  *Fout , t );
38         C_ADDTO( *Fout ,  t );
39         ++Fout2;
40         ++Fout;
41     }while (--m);
42 }
```

Here is the caller graph for this function:



13.53.1.2 static void kf_bfly3 (kiss_fft_cpx * *Fout*, const size_t *fstride*, const kiss_fft_cfg *st*, size_t *m*) [static]

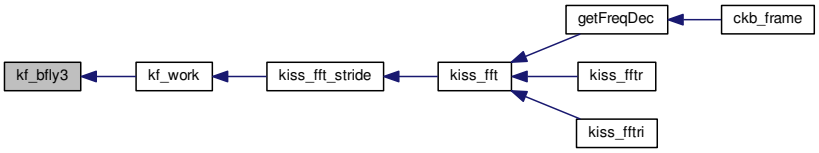
Definition at line 92 of file kiss_fft.c.

References C_ADD, C_ADDTO, C_FIXDIV, C_MUL, C_MULBYSCALAR, C_SUB, HALF_OF, kiss_fft_cpx::i, kiss_fft_cpx::r, and kiss_fft_state::twiddles.

Referenced by kf_work().

```
98 {
99      size_t k=m;
100      const size_t m2 = 2*m;
101      kiss_fft_cpx *tw1,*tw2;
102      kiss_fft_cpx scratch[5];
103      kiss_fft_cpx epi3;
104      epi3 = st->twiddles[fstride*m];
105
106      tw1=tw2=st->twiddles;
107
108      do{
109          C_FIXDIV(*Fout,3); C_FIXDIV(Fout[m],3); C_FIXDIV(Fout[m2],3);
110
111          C_MUL(scratch[1],Fout[m] , *tw1);
112          C_MUL(scratch[2],Fout[m2] , *tw2);
113
114          C_ADD(scratch[3],scratch[1],scratch[2]);
115          C_SUB(scratch[0],scratch[1],scratch[2]);
116          tw1 += fstride;
117          tw2 += fstride*2;
118
119          Fout[m].r = Fout->r - HALF_OF(scratch[3].r);
120          Fout[m].i = Fout->i - HALF_OF(scratch[3].i);
121
122          C_MULBYSCALAR( scratch[0] , epi3.i );
123
124          C_ADDTO(*Fout,scratch[3]);
125
126          Fout[m2].r = Fout[m].r + scratch[0].i;
127          Fout[m2].i = Fout[m].i - scratch[0].r;
128
129          Fout[m].r -= scratch[0].i;
130          Fout[m].i += scratch[0].r;
131
132          ++Fout;
133      }while(--k);
134 }
```

Here is the caller graph for this function:



13.53.1.3 static void kf_bfly4 (kiss_fft_cpx * *Fout*, const size_t *fstride*, const kiss_fft_cfg *st*, const size_t *m*) [static]

Definition at line 44 of file kiss_fft.c.

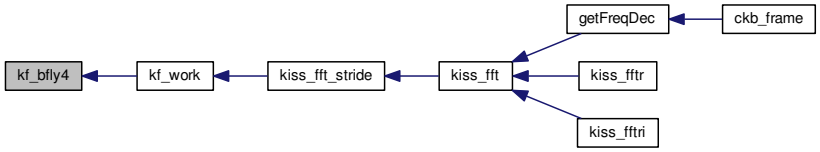
References C_ADD, C_ADDTO, C_FIXDIV, C_MUL, C_SUB, kiss_fft_cpx::i, kiss_fft_state::inverse, kiss_fft_cpx::r, and kiss_fft_state::twiddles.

Referenced by kf_work().

```
50 {
51     kiss_fft_cpx *tw1,*tw2,*tw3;
52     kiss_fft_cpx scratch[6];
53     size_t k=m;
54     const size_t m2=2*m;
55     const size_t m3=3*m;
56
57
58     tw3 = tw2 = tw1 = st->twiddles;
59
60     do {
61         C_FIXDIV(*Fout,4); C_FIXDIV(Fout[m],4); C_FIXDIV(Fout[m2],4);
      C_FIXDIV(Fout[m3],4);
62
```

```
63          C_MUL(scratch[0],Fout[m] , *tw1 );
64          C_MUL(scratch[1],Fout[m2] , *tw2 );
65          C_MUL(scratch[2],Fout[m3] , *tw3 );
66
67          C_SUB( scratch[5] , *Fout, scratch[1] );
68          C_ADDTO(*Fout, scratch[1]);
69          C_ADD( scratch[3] , scratch[0] , scratch[2] );
70          C_SUB( scratch[4] , scratch[0] , scratch[2] );
71          C_SUB( Fout[m2], *Fout, scratch[3] );
72          tw1 += fstride;
73          tw2 += fstride*2;
74          tw3 += fstride*3;
75          C_ADDTO( *Fout , scratch[3] );
76
77          if(st->inverse) {
78              Fout[m].r = scratch[5].r - scratch[4].i;
79              Fout[m].i = scratch[5].i + scratch[4].r;
80              Fout[m3].r = scratch[5].r + scratch[4].i;
81              Fout[m3].i = scratch[5].i - scratch[4].r;
82          }else{
83              Fout[m].r = scratch[5].r + scratch[4].i;
84              Fout[m].i = scratch[5].i - scratch[4].r;
85              Fout[m3].r = scratch[5].r - scratch[4].i;
86              Fout[m3].i = scratch[5].i + scratch[4].r;
87          }
88          ++Fout;
89      }while(--k);
90  }
```

Here is the caller graph for this function:



13.53.1.4 static void kf_bfly5 (kiss_fft_cpx * *Fout,* const size_t *fstride,* const kiss_fft_cfg *st,* int *m*) [static]

Definition at line 136 of file kiss_fft.c.

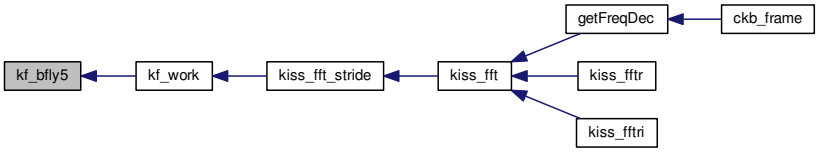
References C_ADD, C_FIXDIV, C_MUL, C_SUB, kiss_fft_cpx::i, kiss_fft_cpx::r, S_MUL, and kiss_fft_state::twiddles.

Referenced by kf_work().

```
142 {
143     kiss_fft_cpx *Fout0,*Fout1,*Fout2,*Fout3,*Fout4;
144     int u;
145     kiss_fft_cpx scratch[13];
146     kiss_fft_cpx * twiddles = st->twiddles;
147     kiss_fft_cpx *tw;
148     kiss_fft_cpx ya,yb;
149     ya = twiddles[fstride*m];
150     yb = twiddles[fstride*2*m];
151
152     Fout0=Fout;
153     Fout1=Fout0+m;
154     Fout2=Fout0+2*m;
155     Fout3=Fout0+3*m;
156     Fout4=Fout0+4*m;
157
158     tw=st->twiddles;
159     for ( u=0; u<m; ++u ) {
160         C_FIXDIV( *Fout0,5); C_FIXDIV( *Fout1,5); C_FIXDIV( *Fout2,5);
      C_FIXDIV( *Fout3,5); C_FIXDIV( *Fout4,5);
161         scratch[0] = *Fout0;
162
163         C_MUL(scratch[1] ,*Fout1, tw[u*fstride]);
164         C_MUL(scratch[2] ,*Fout2, tw[2*u*fstride]);
```

```
165         C_MUL(scratch[3] ,*Fout3, tw[3*u*fstride]);
166         C_MUL(scratch[4] ,*Fout4, tw[4*u*fstride]);
167
168         C_ADD( scratch[7],scratch[1],scratch[4]);
169         C_SUB( scratch[10],scratch[1],scratch[4]);
170         C_ADD( scratch[8],scratch[2],scratch[3]);
171         C_SUB( scratch[9],scratch[2],scratch[3]);
172
173         Fout0->r += scratch[7].r + scratch[8].r;
174         Fout0->i += scratch[7].i + scratch[8].i;
175
176         scratch[5].r = scratch[0].r + S_MUL(scratch[7].r,ya.r) + S_MUL(scratch[8].r,yb.
    r);
177         scratch[5].i = scratch[0].i + S_MUL(scratch[7].i,ya.r) + S_MUL(scratch[8].i,yb.
    r);
178
179         scratch[6].r =  S_MUL(scratch[10].i,ya.i) + S_MUL(scratch[9].i,yb.
    i);
180         scratch[6].i = -S_MUL(scratch[10].r,ya.i) - S_MUL(scratch[9].r,yb.
    i);
181
182         C_SUB(*Fout1,scratch[5],scratch[6]);
183         C_ADD(*Fout4,scratch[5],scratch[6]);
184
185         scratch[11].r = scratch[0].r + S_MUL(scratch[7].r,yb.r) + S_MUL(scratch[8].r,ya.
    r);
186         scratch[11].i = scratch[0].i + S_MUL(scratch[7].i,yb.r) + S_MUL(scratch[8].i,ya.
    r);
187         scratch[12].r = - S_MUL(scratch[10].i,yb.i) + S_MUL(scratch[9].i,ya.
    i);
188         scratch[12].i = S_MUL(scratch[10].r,yb.i) - S_MUL(scratch[9].r,ya.
    i);
189
190         C_ADD(*Fout2,scratch[11],scratch[12]);
191         C_SUB(*Fout3,scratch[11],scratch[12]);
192
193         ++Fout0;++Fout1;++Fout2;++Fout3;++Fout4;
194     }
195 }
```

Here is the caller graph for this function:



13.53.1.5 static void kf_bfly_generic (kiss_fft_cpx * *Fout*, const size_t *fstride*, const kiss_fft_cfg *st*, int *m*, int *p*) [static]

Definition at line 198 of file kiss_fft.c.

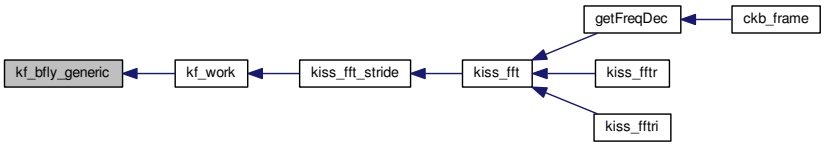
References C_ADDTO, C_FIXDIV, C_MUL, KISS_FFT_TMP_ALLOC, KISS_FFT_TMP_FREE, kiss_fft_state::nfft, and kiss_fft_state::twiddles.

Referenced by kf_work().

```
205 {
206     int u,k,q1,q;
207     kiss_fft_cpx * twiddles = st->twiddles;
208     kiss_fft_cpx t;
209     int Norig = st->nfft;
210
211     kiss_fft_cpx * scratch = (kiss_fft_cpx*)
    KISS_FFT_TMP_ALLOC(sizeof(kiss_fft_cpx)*p);
212
213     for ( u=0; u<m; ++u ) {
214         k=u;
215         for ( q1=0 ; q1<p ; ++q1 ) {
```

```
216             scratch[q1] = Fout[ k  ];
217             C_FIXDIV(scratch[q1],p);
218             k += m;
219         }
220
221         k=u;
222         for ( q1=0 ; q1<p ; ++q1 ) {
223             int twidx=0;
224             Fout[ k ] = scratch[0];
225             for (q=1;q<p;++q ) {
226                 twidx += fstride * k;
227                 if (twidx>=Norig) twidx-=Norig;
228                 C_MUL(t,scratch[q] , twiddles[twidx] );
229                 C_ADDTO( Fout[ k ] ,t);
230             }
231             k += m;
232         }
233     }
234     KISS_FFT_TMP_FREE(scratch);
235 }
```

Here is the caller graph for this function:



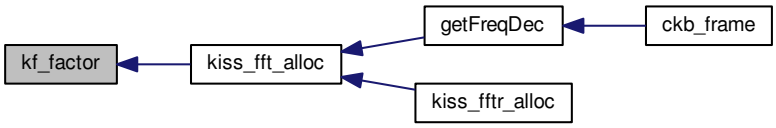
13.53.1.6 static void kf_factor (int *n*, int * *facbuf*) [static]

Definition at line 309 of file kiss_fft.c.

Referenced by kiss_fft_alloc().

```
310 {
311     int p=4;
312     double floor_sqrt;
313     floor_sqrt = floor( sqrt((double)n) );
314
315     /*factor out powers of 4, powers of 2, then any remaining primes */
316     do {
317         while (n % p) {
318             switch (p) {
319                 case 4: p = 2; break;
320                 case 2: p = 3; break;
321                 default: p += 2; break;
322             }
323             if (p > floor_sqrt)
324                 p = n;          /* no more factors, skip to end */
325         }
326         n /= p;
327         *facbuf++ = p;
328         *facbuf++ = n;
329     } while (n > 1);
330 }
```

Here is the caller graph for this function:



13.53.1.7 static void kf_work (kiss_fft_cpx * *Fout*, const kiss_fft_cpx * *f*, const size_t *fstride*, int *in_stride*, int * *factors*, const kiss_fft_cfg *st*) [static]

Definition at line 238 of file kiss_fft.c.

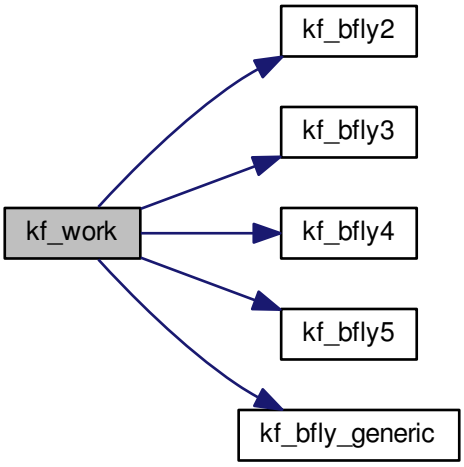
References kf_bfly2(), kf_bfly3(), kf_bfly4(), kf_bfly5(), and kf_bfly_generic().

Referenced by kiss_fft_stride().

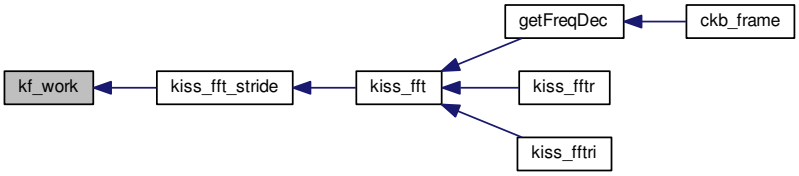
```
246 {
247     kiss_fft_cpx * Fout_beg=Fout;
248     const int p=*factors++; /* the radix  */
249     const int m=*factors++; /* stage's fft length/p */
250     const kiss_fft_cpx * Fout_end = Fout + p*m;
251
252 #ifdef _OPENMP
253     // use openmp extensions at the
254     // top-level (not recursive)
255     if (fstride==1 && p<=5)
256     {
257         int k;
258
259         // execute the p different work units in different threads
260 #       pragma omp parallel for
261         for (k=0;k<p;++k)
262             kf_work( Fout +k*m, f+ fstride*in_stride*k,fstride*p,in_stride,factors,st);
263         // all threads have joined by this point
264
265         switch (p) {
266             case 2: kf_bfly2(Fout,fstride,st,m); break;
267             case 3: kf_bfly3(Fout,fstride,st,m); break;
268             case 4: kf_bfly4(Fout,fstride,st,m); break;
269             case 5: kf_bfly5(Fout,fstride,st,m); break;
270             default: kf_bfly_generic(Fout,fstride,st,m,p); break;
271         }
272         return;
273     }
274 #endif
275
276     if (m==1) {
277         do{
278             *Fout = *f;
279             f += fstride*in_stride;
280         }while(++Fout != Fout_end );
281     }else{
282         do{
283             // recursive call:
284             // DFT of size m*p performed by doing
285             // p instances of smaller DFTs of size m,
286             // each one takes a decimated version of the input
287             kf_work( Fout , f, fstride*p, in_stride, factors,st);
288             f += fstride*in_stride;
289         }while( (Fout += m) != Fout_end );
290     }
291
292     Fout=Fout_beg;
293
294     // recombine the p smaller DFTs
295     switch (p) {
296         case 2: kf_bfly2(Fout,fstride,st,m); break;
297         case 3: kf_bfly3(Fout,fstride,st,m); break;
298         case 4: kf_bfly4(Fout,fstride,st,m); break;
299         case 5: kf_bfly5(Fout,fstride,st,m); break;
```

```
300         default: kf_bfly_generic(Fout,fstride,st,m,p); break;
301     }
302 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.53.1.8 void kiss_fft (kiss_fft_cfg *cfg,* const kiss_fft_cpx * *fin,* kiss_fft_cpx * *fout*)

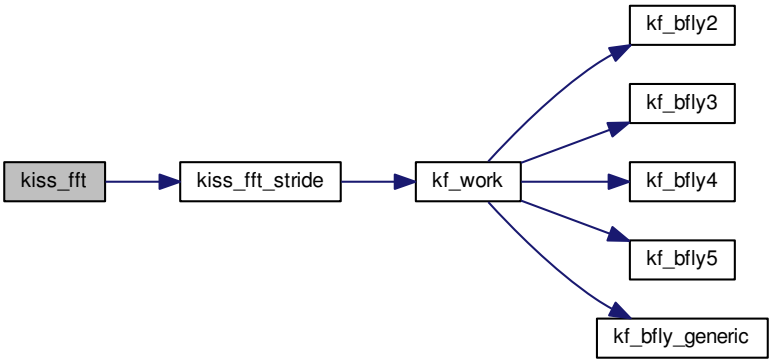
Definition at line 385 of file kiss_fft.c.

References kiss_fft_stride().

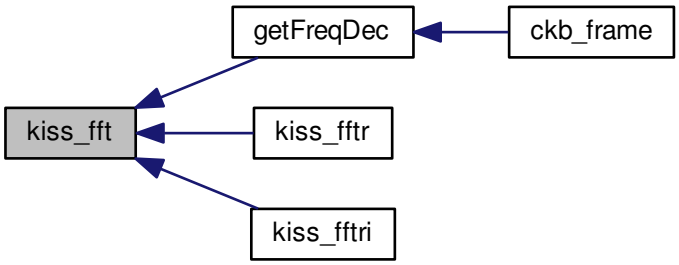
Referenced by getFreqDec(), kiss_fftr(), and kiss_fftri().

```
386 {
387     kiss_fft_stride(cfg,fin,fout,1);
388 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.53.1.9 kiss_fft_cfg kiss_fft_alloc (int *nfft,* int *inverse_fft,* void * *mem,* size_t * *lenmem*)

Definition at line 339 of file kiss_fft.c.

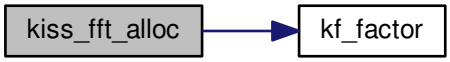
References kiss_fft_state::factors, kiss_fft_state::inverse, kf_cexp, kf_factor(), KISS_FFT_MALLOC, kiss_fft_state::nfft, phase, and kiss_fft_state::twiddles.

Referenced by getFreqDec(), and kiss_fftr_alloc().

```
340 {
341     kiss_fft_cfg st=NULL;
342     size_t memneeded = sizeof(struct kiss_fft_state)
343         + sizeof(kiss_fft_cpx)*(nfft-1); /* twiddle factors*/
344
345     if ( lenmem==NULL ) {
346         st = ( kiss_fft_cfg)KISS_FFT_MALLOC( memneeded );
347     }else{
348         if (mem != NULL && *lenmem >= memneeded)
349             st = (kiss_fft_cfg)mem;
350         *lenmem = memneeded;
351     }
352     if (st) {
353         int i;
```

```
354         st->nfft=nfft;
355         st->inverse = inverse_fft;
356
357         for (i=0;i<nfft;++i) {
358             const double pi=3.14159265358979323846264338327950288419716939937510582097494;
359             double phase = -2*pi*i / nfft;
360             if (st->inverse)
361                 phase *= -1;
362             kf_cexp(st->twiddles+i, phase );
363         }
364
365         kf_factor(nfft,st->factors);
366     }
367     return st;
368 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

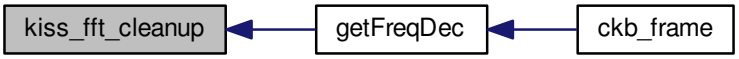
13.53.1.10 void kiss_fft_cleanup (void)

Definition at line 391 of file kiss_fft.c.

Referenced by getFreqDec().

```
392 {
393     // nothing needed any more
394 }
```

Here is the caller graph for this function:



13.53.1.11 int kiss_fft_next_fast_size (int *n*)

Definition at line 396 of file kiss_fft.c.

```
397 {
398     while(1) {
399         int m=n;
400         while ( (m%2) == 0 ) m/=2;
401         while ( (m%3) == 0 ) m/=3;
402         while ( (m%5) == 0 ) m/=5;
403         if (m<=1)
404             break; /* n is completely factorable by twos, threes, and fives */
405         n++;
406     }
407     return n;
408 }
```

13.53.1.12 void kiss_fft_stride (kiss_fft_cfg *st,* const kiss_fft_cpx * *fin,* kiss_fft_cpx * *fout,* int *in_stride*)

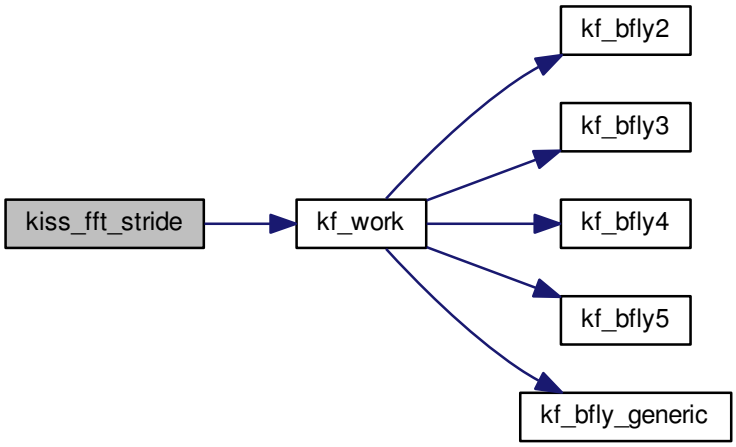
Definition at line 371 of file kiss_fft.c.

References kiss_fft_state::factors, kf_work(), KISS_FFT_TMP_ALLOC, KISS_FFT_TMP_FREE, and kiss_fft_state::nfft.

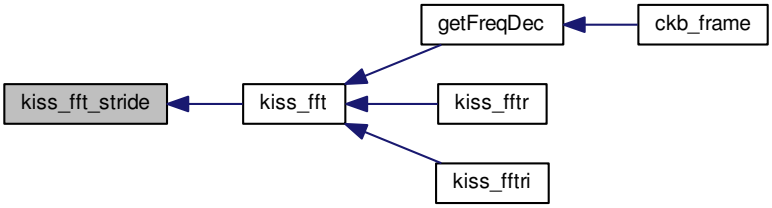
Referenced by kiss_fft().

```
372 {
373     if (fin == fout) {
374         //NOTE: this is not really an in-place FFT algorithm.
375         //It just performs an out-of-place FFT into a temp buffer
376         kiss_fft_cpx * tmpbuf = (kiss_fft_cpx*)
      KISS_FFT_TMP_ALLOC( sizeof(kiss_fft_cpx)*st->nfft);
377         kf_work(tmpbuf,fin,1,in_stride, st->factors,st);
378         memcpy(fout,tmpbuf,sizeof(kiss_fft_cpx)*st->nfft);
379         KISS_FFT_TMP_FREE(tmpbuf);
380     }else{
381         kf_work( fout, fin, 1,in_stride, st->factors,st );
382     }
383 }
```

Here is the call graph for this function:



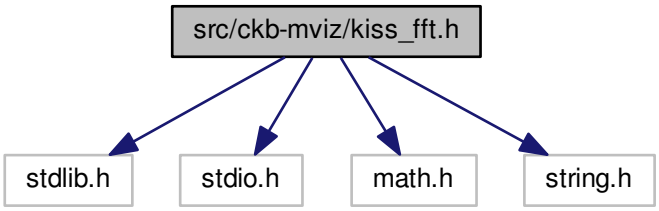
Here is the caller graph for this function:



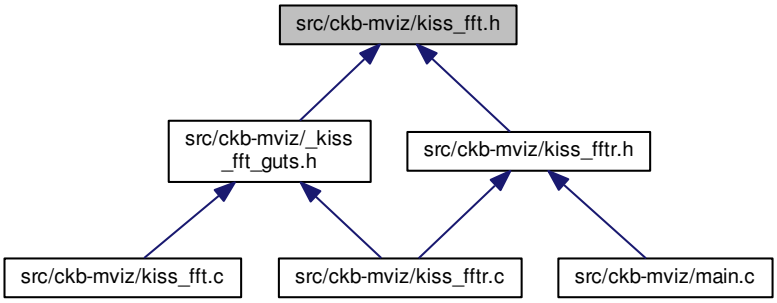
13.54 src/ckb-mviz/kiss_fft.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <string.h>
```

Include dependency graph for kiss_fft.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct kiss_fft_cpx

Macros

- #define KISS_FFT_MALLOC malloc
- #define KISS_FFT_FREE free
- #define kiss_fft_scalar float
- #define kiss_fft_free free
- #define kiss_fftr_next_fast_size_real(n) (kiss_fft_next_fast_size(((n)+1)>>1)<<1)

Typedefs

- typedef struct kiss_fft_state * kiss_fft_cfg

Functions

- kiss_fft_cfg kiss_fft_alloc (int nfft, int inverse_fft, void *mem, size_t *lenmem)

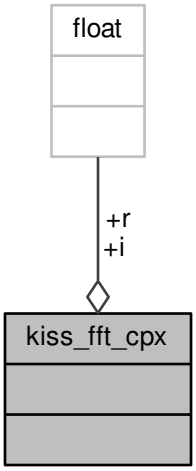
- void kiss_fft (kiss_fft_cfg cfg, const kiss_fft_cpx *fin, kiss_fft_cpx *fout)
- void kiss_fft_stride (kiss_fft_cfg cfg, const kiss_fft_cpx *fin, kiss_fft_cpx *fout, int fin_stride)
- void kiss_fft_cleanup (void)
- int kiss_fft_next_fast_size (int n)

13.54.1 Data Structure Documentation

13.54.1.1 struct kiss_fft_cpx

Definition at line 51 of file kiss_fft.h.

Collaboration diagram for kiss_fft_cpx:



Data Fields

float	i	
float	r	

13.54.2 Macro Definition Documentation

13.54.2.1 #define KISS_FFT_FREE free

Definition at line 33 of file kiss_fft.h.

13.54.2.2 #define kiss_fft_free free

Definition at line 102 of file kiss_fft.h.

Referenced by getFreqDec().

13.54.2.3 #define KISS_FFT_MALLOC malloc

Definition at line 32 of file kiss_fft.h.

Referenced by kiss_fft_alloc(), and kiss_fftr_alloc().

13.54.2.4 #define kiss_fft_scalar float

Definition at line 47 of file kiss_fft.h.

13.54.2.5 #define kiss_fftr_next_fast_size_real(*n*) (kiss_fft_next_fast_size(((n)+1)>>1)<<1)

Definition at line 117 of file kiss_fft.h.

13.54.3 Typedef Documentation

13.54.3.1 typedef struct kiss_fft_state* kiss_fft_cfg

Definition at line 56 of file kiss_fft.h.

13.54.4 Function Documentation

13.54.4.1 void kiss_fft (kiss_fft_cfg *cfg,* const kiss_fft_cpx * *fin,* kiss_fft_cpx * *fout*)

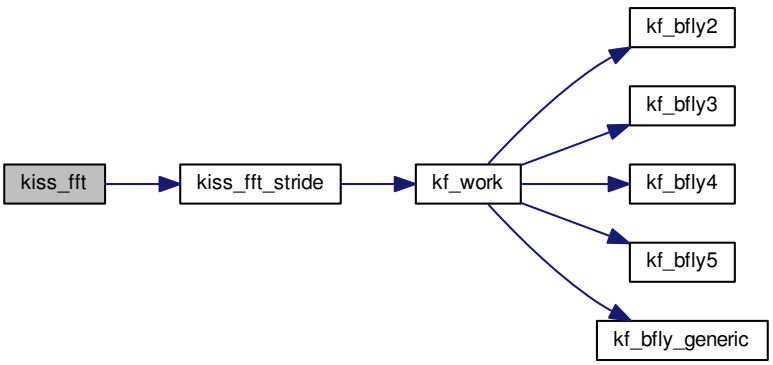
Definition at line 385 of file kiss_fft.c.

References kiss_fft_stride().

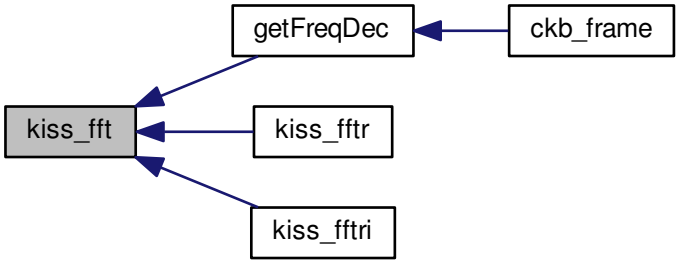
Referenced by getFreqDec(), kiss_fftr(), and kiss_fftri().

```
386 {
387     kiss_fft_stride(cfg,fin,fout,1);
388 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.54.4.2 kiss_fft_cfg kiss_fft_alloc (int *nfft,* int *inverse_fft,* void * *mem,* size_t * *lenmem*)

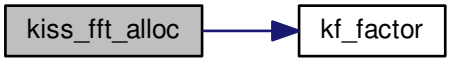
Definition at line 339 of file kiss_fft.c.

References kiss_fft_state::factors, kiss_fft_state::inverse, kf_cexp, kf_factor(), KISS_FFT_MALLOC, kiss_fft_state::nfft, phase, and kiss_fft_state::twiddles.

Referenced by getFreqDec(), and kiss_fftr_alloc().

```
340 {
341     kiss_fft_cfg st=NULL;
342     size_t memneeded = sizeof(struct kiss_fft_state)
343         + sizeof(kiss_fft_cpx)*(nfft-1); /* twiddle factors*/
344
345     if ( lenmem==NULL ) {
346         st = ( kiss_fft_cfg)KISS_FFT_MALLOC( memneeded );
347     }else{
348         if (mem != NULL && *lenmem >= memneeded)
349             st = (kiss_fft_cfg)mem;
350         *lenmem = memneeded;
351     }
352     if (st) {
353         int i;
354         st->nfft=nfft;
355         st->inverse = inverse_fft;
356
357         for (i=0;i<nfft;++i) {
358             const double pi=3.14159265358979323846264338327950288419716939937510582097494 4;
359             double phase = -2*pi*i / nfft;
360             if (st->inverse)
361                 phase *= -1;
362             kf_cexp(st->twiddles+i, phase );
363         }
364
365         kf_factor(nfft,st->factors);
366     }
367     return st;
368 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

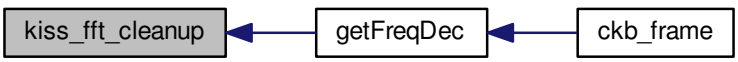
13.54.4.3 void kiss_fft_cleanup (void)

Definition at line 391 of file kiss_fft.c.

Referenced by getFreqDec().

```
392 {
393     // nothing needed any more
394 }
```

Here is the caller graph for this function:



13.54.4.4 int kiss_fft_next_fast_size (int *n*)

Definition at line 396 of file kiss_fft.c.

```
397 {
398     while(1) {
399         int m=n;
```

```
400         while ( (m%2) == 0 ) m/=2;
401         while ( (m%3) == 0 ) m/=3;
402         while ( (m%5) == 0 ) m/=5;
403         if (m<=1)
404             break; /* n is completely factorable by twos, threes, and fives */
405         n++;
406     }
407     return n;
408 }
```

13.54.4.5 void kiss_fft_stride (kiss_fft_cfg *cfg,* const kiss_fft_cpx * *fin,* kiss_fft_cpx * *fout,* int *fin_stride*)

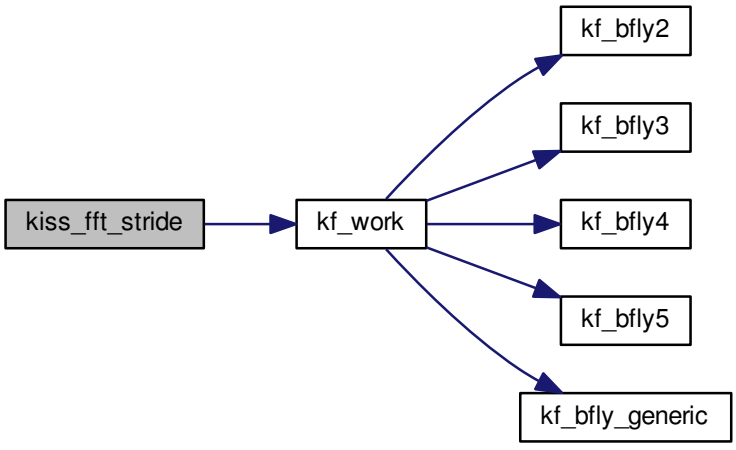
Definition at line 371 of file kiss_fft.c.

References kiss_fft_state::factors, kf_work(), KISS_FFT_TMP_ALLOC, KISS_FFT_TMP_FREE, and kiss_fft_state::nfft.

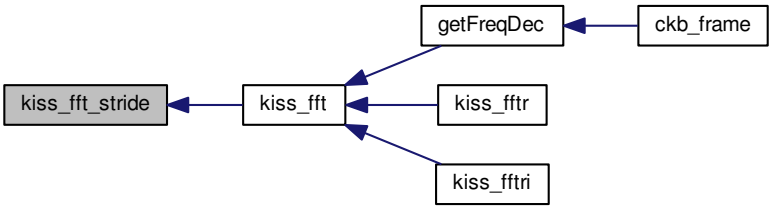
Referenced by kiss_fft().

```
372 {
373     if (fin == fout) {
374         //NOTE: this is not really an in-place FFT algorithm.
375         //It just performs an out-of-place FFT into a temp buffer
376         kiss_fft_cpx * tmpbuf = (kiss_fft_cpx*)
      KISS_FFT_TMP_ALLOC( sizeof(kiss_fft_cpx)*st->nfft);
377         kf_work(tmpbuf,fin,1,in_stride, st->factors,st);
378         memcpy(fout,tmpbuf,sizeof(kiss_fft_cpx)*st->nfft);
379         KISS_FFT_TMP_FREE(tmpbuf);
380     }else{
381         kf_work( fout, fin, 1,in_stride, st->factors,st );
382     }
383 }
```

Here is the call graph for this function:



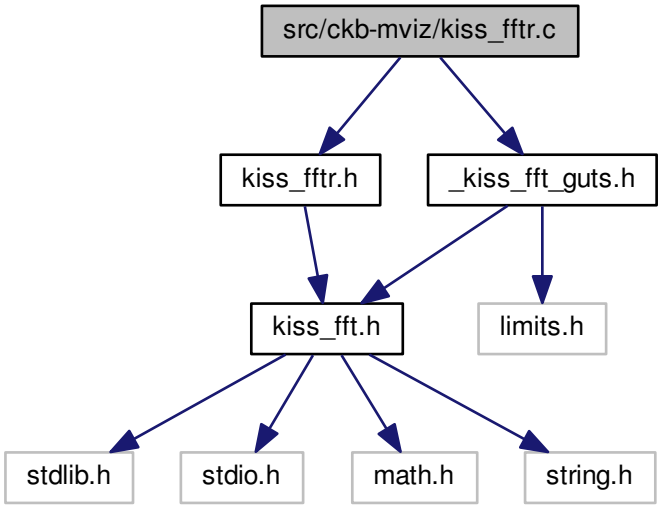
Here is the caller graph for this function:



13.55 src/ckb-mviz/kiss_fftr.c File Reference

```
#include "kiss_fftr.h"
#include "_kiss_fft_guts.h"
```

Include dependency graph for kiss_fftr.c:



Data Structures

- struct kiss_fftr_state

Functions

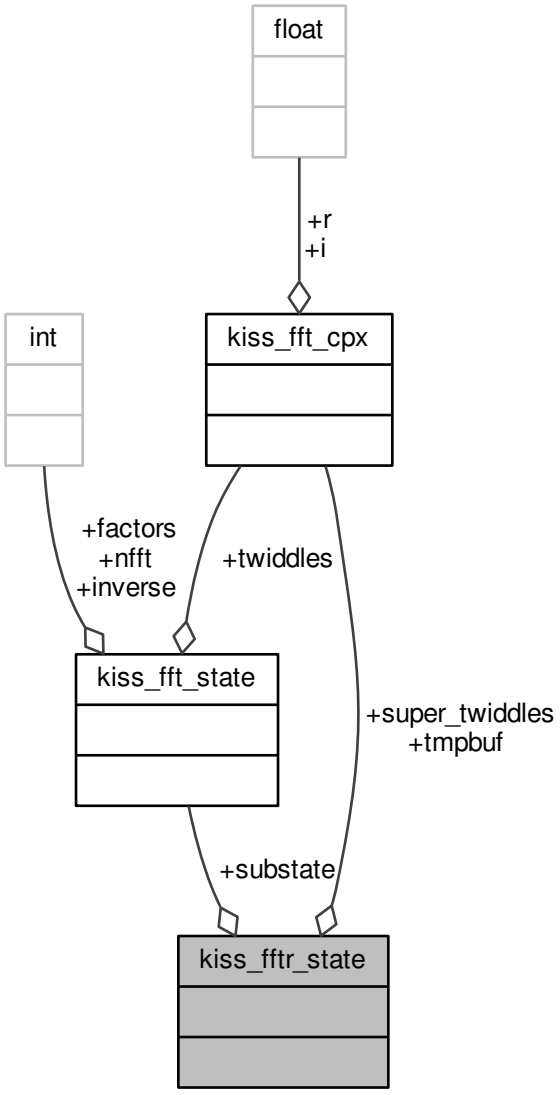
- kiss_fftr_cfg kiss_fftr_alloc (int nfft, int inverse_fft, void ∗mem, size_t ∗lenmem)
- void kiss_fftr (kiss_fftr_cfg st, const float ∗timedata, kiss_fft_cpx ∗freqdata)
- void kiss_fftri (kiss_fftr_cfg st, const kiss_fft_cpx ∗freqdata, float ∗timedata)

13.55.1 Data Structure Documentation

13.55.1.1 struct kiss_fftr_state

Definition at line 18 of file kiss_fftr.c.

Collaboration diagram for kiss_fftr_state:



Data Fields

kiss_fft_cfg	substate	
kiss_fft_cpx *	super_twiddles	

kiss_fft_cpx *	tmpbuf	

13.55.2 Function Documentation

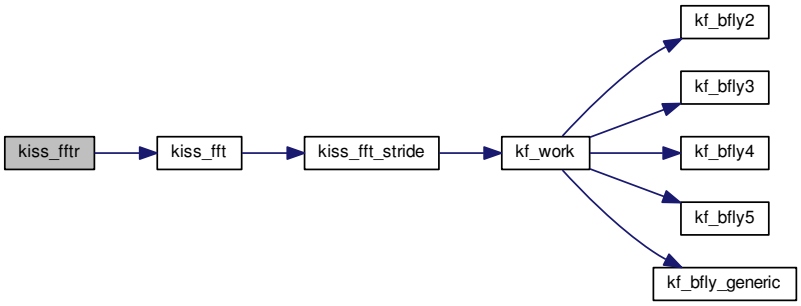
13.55.2.1 void kiss_fftr (kiss_fftr_cfg *st*, const float * *timedata*, kiss_fft_cpx * *freqdata*)

Definition at line 67 of file kiss_fftr.c.

References C_ADD, C_FIXDIV, C_MUL, C_SUB, CHECK_OVERFLOW_OP, HALF_OF, kiss_fft_cpx::i, kiss_fft_state::inverse, kiss_fft(), kiss_fft_state::nfft, kiss_fft_cpx::r, kiss_fftr_state::substate, kiss_fftr_state::super_twiddles, and kiss_fftr_state::tmpbuf.

```
68 {
69     /* input buffer timedata is stored row-wise */
70     int k,ncfft;
71     kiss_fft_cpx fpnk,fpk,f1k,f2k,tw,tdc;
72
73     if ( st->substate->inverse) {
74         fprintf(stderr,"kiss fft usage error: improper alloc\n");
75         exit(1);
76     }
77
78     ncfft = st->substate->nfft;
79
80     /*perform the parallel fft of two real signals packed in real,imag*/
81     kiss_fft( st->substate , (const kiss_fft_cpx*)timedata, st->
      tmpbuf );
82     /* The real part of the DC element of the frequency spectrum in st->tmpbuf
83      * contains the sum of the even-numbered elements of the input time sequence
84      * The imag part is the sum of the odd-numbered elements
85      *
86      * The sum of tdc.r and tdc.i is the sum of the input time sequence.
87      *      yielding DC of input time sequence
88      * The difference of tdc.r - tdc.i is the sum of the input (dot product) [1,-1,1,-1...
89      *      yielding Nyquist bin of input time sequence
90      */
91
92     tdc.r = st->tmpbuf[0].r;
93     tdc.i = st->tmpbuf[0].i;
94     C_FIXDIV(tdc,2);
95     CHECK_OVERFLOW_OP(tdc.r ,+, tdc.i);
96     CHECK_OVERFLOW_OP(tdc.r ,-, tdc.i);
97     freqdata[0].r = tdc.r + tdc.i;
98     freqdata[ncfft].r = tdc.r - tdc.i;
99 #ifdef USE_SIMD
100     freqdata[ncfft].i = freqdata[0].i = _mm_set1_ps(0);
101 #else
102     freqdata[ncfft].i = freqdata[0].i = 0;
103 #endif
104
105     for ( k=1;k <= ncfft/2 ; ++k ) {
106         fpk    = st->tmpbuf[k];
107         fpnk.r =   st->tmpbuf[ncfft-k].r;
108         fpnk.i = - st->tmpbuf[ncfft-k].i;
109         C_FIXDIV(fpk,2);
110         C_FIXDIV(fpnk,2);
111
112         C_ADD( f1k, fpk , fpnk );
113         C_SUB( f2k, fpk , fpnk );
114         C_MUL( tw , f2k , st->super_twiddles[k-1]);
115
116         freqdata[k].r = HALF_OF(f1k.r + tw.r);
117         freqdata[k].i = HALF_OF(f1k.i + tw.i);
118         freqdata[ncfft-k].r = HALF_OF(f1k.r - tw.r);
119         freqdata[ncfft-k].i = HALF_OF(tw.i - f1k.i);
120     }
121 }
```

Here is the call graph for this function:



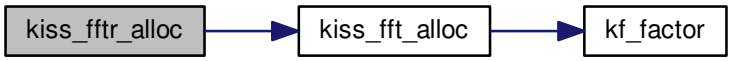
13.55.2.2 kiss_fftr_cfg kiss_fftr_alloc (int *nfft*, int *inverse_fft*, void * *mem*, size_t * *lenmem*)

Definition at line 27 of file kiss_fftr.c.

References kf_cexp, kiss_fft_alloc(), KISS_FFT_MALLOC, phase, kiss_fftr_state::substate, kiss_fftr_state::super_twiddles, and kiss_fftr_state::tmpbuf.

```
28 {
29     int i;
30     kiss_fftr_cfg st = NULL;
31     size_t subsize, memneeded;
32
33     if (nfft & 1) {
34         fprintf(stderr,"Real FFT optimization must be even.\n");
35         return NULL;
36     }
37     nfft >>= 1;
38
39     kiss_fft_alloc (nfft, inverse_fft, NULL, &subsize);
40     memneeded = sizeof(struct kiss_fftr_state) + subsize + sizeof(
      kiss_fft_cpx) * ( nfft * 3 / 2);
41
42     if (lenmem == NULL) {
43         st = (kiss_fftr_cfg) KISS_FFT_MALLOC (memneeded);
44     } else {
45         if (*lenmem >= memneeded)
46             st = (kiss_fftr_cfg) mem;
47         *lenmem = memneeded;
48     }
49     if (!st)
50         return NULL;
51
52     st->substate = (kiss_fft_cfg) (st + 1); /*just beyond kiss_fftr_state struct */
53     st->tmpbuf = (kiss_fft_cpx *) (((char *) st->substate) + subsize);
54     st->super_twiddles = st->tmpbuf + nfft;
55     kiss_fft_alloc(nfft, inverse_fft, st->substate, &subsize);
56
57     for (i = 0; i < nfft/2; ++i) {
58         double phase =
59             -3.14159265358979323846264338327 * ((double) (i+1) / nfft + .5);
60         if (inverse_fft)
61             phase *= -1;
62         kf_cexp (st->super_twiddles+i,phase);
63     }
64     return st;
65 }
```

Here is the call graph for this function:



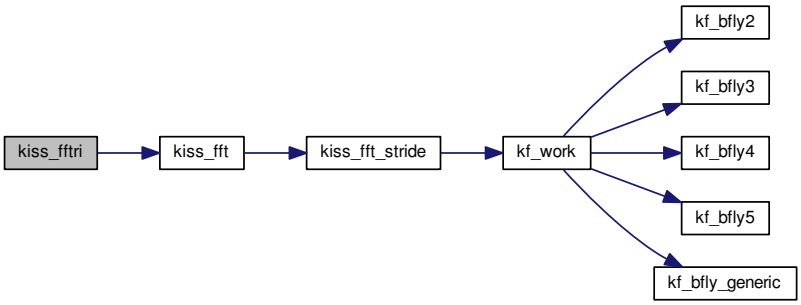
13.55.2.3 void kiss_fftri (kiss_fftr_cfg *st*, const kiss_fft_cpx * *freqdata*, float * *timedata*)

Definition at line 123 of file kiss_fftr.c.

References C_ADD, C_FIXDIV, C_MUL, C_SUB, kiss_fft_cpx::i, kiss_fft_state::inverse, kiss_fft(), kiss_fft_state::nfft, kiss_fft_cpx::r, kiss_fftr_state::substate, kiss_fftr_state::super_twiddles, and kiss_fftr_state::tmpbuf.

```
124 {
125     /* input buffer timedata is stored row-wise */
126     int k, ncfft;
127
128     if (st->substate->inverse == 0) {
129         fprintf (stderr, "kiss fft usage error: improper alloc\n");
130         exit (1);
131     }
132
133     ncfft = st->substate->nfft;
134
135     st->tmpbuf[0].r = freqdata[0].r + freqdata[ncfft].r;
136     st->tmpbuf[0].i = freqdata[0].r - freqdata[ncfft].r;
137     C_FIXDIV(st->tmpbuf[0],2);
138
139     for (k = 1; k <= ncfft / 2; ++k) {
140         kiss_fft_cpx fk, fnkc, fek, fok, tmp;
141         fk = freqdata[k];
142         fnkc.r = freqdata[ncfft - k].r;
143         fnkc.i = -freqdata[ncfft - k].i;
144         C_FIXDIV( fk , 2 );
145         C_FIXDIV( fnkc , 2 );
146
147         C_ADD (fek, fk, fnkc);
148         C_SUB (tmp, fk, fnkc);
149         C_MUL (fok, tmp, st->super_twiddles[k-1]);
150         C_ADD (st->tmpbuf[k],     fek, fok);
151         C_SUB (st->tmpbuf[ncfft - k], fek, fok);
152 #ifdef USE_SIMD
153         st->tmpbuf[ncfft - k].i *= _mm_set1_ps(-1.0);
154 #else
155         st->tmpbuf[ncfft - k].i *= -1;
156 #endif
157     }
158     kiss_fft (st->substate, st->tmpbuf, (kiss_fft_cpx *) timedata);
159 }
```

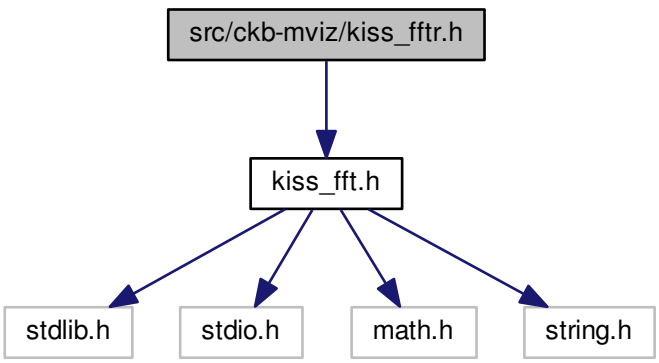
Here is the call graph for this function:



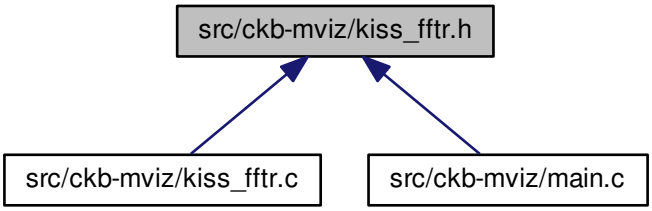
13.56 src/ckb-mviz/kiss_fftr.h File Reference

```
#include "kiss_fft.h"
```

Include dependency graph for kiss_fftr.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define kiss_fftr_free free

Typedefs

- typedef struct kiss_fftr_state * kiss_fftr_cfg

Functions

- kiss_fftr_cfg kiss_fftr_alloc (int nfft, int inverse_fft, void *mem, size_t *lenmem)
- void kiss_fftr (kiss_fftr_cfg cfg, const float *timedata, kiss_fft_cpx *freqdata)
- void kiss_fftri (kiss_fftr_cfg cfg, const kiss_fft_cpx *freqdata, float *timedata)

13.56.1 Macro Definition Documentation

13.56.1.1 #define kiss_fftr_free free

Definition at line 41 of file kiss_fftr.h.

13.56.2 Typedef Documentation

13.56.2.1 typedef struct kiss_fftr_state* kiss_fftr_cfg

Definition at line 18 of file kiss_fftr.h.

13.56.3 Function Documentation

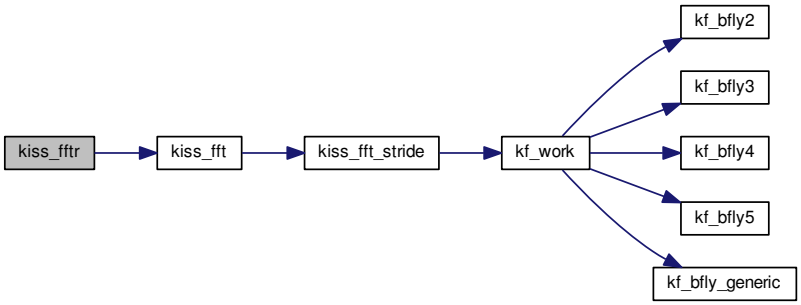
13.56.3.1 void kiss_fftr (kiss_fftr_cfg *cfg,* const float * *timedata,* kiss_fft_cpx * *freqdata*)

Definition at line 67 of file kiss_fftr.c.

References C_ADD, C_FIXDIV, C_MUL, C_SUB, CHECK_OVERFLOW_OP, HALF_OF, kiss_fft_cpx::i, kiss_fft_state::inverse, kiss_fft(), kiss_fft_state::nfft, kiss_fft_cpx::r, kiss_fftr_state::substate, kiss_fftr_state::super_twiddles, and kiss_fftr_state::tmpbuf.

```
68 {
69     /* input buffer timedata is stored row-wise */
70     int k,ncfft;
71     kiss_fft_cpx fpnk,fpk,f1k,f2k,tw,tdc;
72 
73     if ( st->substate->inverse) {
74         fprintf(stderr,"kiss fft usage error: improper alloc\n");
75         exit(1);
76     }
77 
78     ncfft = st->substate->nfft;
79 
80     /*perform the parallel fft of two real signals packed in real,imag*/
81     kiss_fft( st->substate , (const kiss_fft_cpx*)timedata, st->tmpbuf );
82     /* The real part of the DC element of the frequency spectrum in st->tmpbuf
83      * contains the sum of the even-numbered elements of the input time sequence
84      * The imag part is the sum of the odd-numbered elements
85      *
86      * The sum of tdc.r and tdc.i is the sum of the input time sequence.
87      *      yielding DC of input time sequence
88      * The difference of tdc.r - tdc.i is the sum of the input (dot product) [1,-1,1,-1...
89      *      yielding Nyquist bin of input time sequence
90      */
91 
92     tdc.r = st->tmpbuf[0].r;
93     tdc.i = st->tmpbuf[0].i;
94     C_FIXDIV(tdc,2);
95     CHECK_OVERFLOW_OP(tdc.r ,+, tdc.i);
96     CHECK_OVERFLOW_OP(tdc.r ,-, tdc.i);
97     freqdata[0].r = tdc.r + tdc.i;
98     freqdata[ncfft].r = tdc.r - tdc.i;
99 #ifdef USE_SIMD
100     freqdata[ncfft].i = freqdata[0].i = _mm_set1_ps(0);
101 #else
102     freqdata[ncfft].i = freqdata[0].i = 0;
103 #endif
104 
105     for ( k=1;k <= ncfft/2 ; ++k ) {
106         fpk    = st->tmpbuf[k];
107         fpnk.r =   st->tmpbuf[ncfft-k].r;
108         fpnk.i = - st->tmpbuf[ncfft-k].i;
109         C_FIXDIV(fpk,2);
110         C_FIXDIV(fpnk,2);
111 
112         C_ADD( f1k, fpk , fpnk );
113         C_SUB( f2k, fpk , fpnk );
114         C_MUL( tw , f2k , st->super_twiddles[k-1]);
115 
116         freqdata[k].r = HALF_OF(f1k.r + tw.r);
117         freqdata[k].i = HALF_OF(f1k.i + tw.i);
118         freqdata[ncfft-k].r = HALF_OF(f1k.r - tw.r);
119         freqdata[ncfft-k].i = HALF_OF(tw.i - f1k.i);
120     }
121 }
```

Here is the call graph for this function:



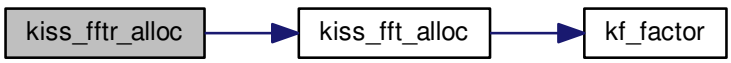
13.56.3.2 kiss_fftr_cfg kiss_fftr_alloc (int *nfft*, int *inverse_fft*, void * *mem*, size_t * *lenmem*)

Definition at line 27 of file kiss_fftr.c.

References kf_cexp, kiss_fft_alloc(), KISS_FFT_MALLOC, phase, kiss_fftr_state::substate, kiss_fftr_state::super_twiddles, and kiss_fftr_state::tmpbuf.

```
28 {
29     int i;
30     kiss_fftr_cfg st = NULL;
31     size_t subsize, memneeded;
32
33     if (nfft & 1) {
34         fprintf(stderr,"Real FFT optimization must be even.\n");
35         return NULL;
36     }
37     nfft >>= 1;
38
39     kiss_fft_alloc (nfft, inverse_fft, NULL, &subsize);
40     memneeded = sizeof(struct kiss_fftr_state) + subsize + sizeof(
      kiss_fft_cpx) * ( nfft * 3 / 2);
41
42     if (lenmem == NULL) {
43         st = (kiss_fftr_cfg) KISS_FFT_MALLOC (memneeded);
44     } else {
45         if (*lenmem >= memneeded)
46             st = (kiss_fftr_cfg) mem;
47         *lenmem = memneeded;
48     }
49     if (!st)
50         return NULL;
51
52     st->substate = (kiss_fft_cfg) (st + 1); /*just beyond kiss_fftr_state struct */
53     st->tmpbuf = (kiss_fft_cpx *) (((char *) st->substate) + subsize);
54     st->super_twiddles = st->tmpbuf + nfft;
55     kiss_fft_alloc(nfft, inverse_fft, st->substate, &subsize);
56
57     for (i = 0; i < nfft/2; ++i) {
58         double phase =
59             -3.14159265358979323846264338327 * ((double) (i+1) / nfft + .5);
60         if (inverse_fft)
61             phase *= -1;
62         kf_cexp (st->super_twiddles+i,phase);
63     }
64     return st;
65 }
```

Here is the call graph for this function:



13.56.3.3 void kiss_fftri (kiss_fftr_cfg *cfg*, const kiss_fft_cpx * *freqdata*, float * *timedata*)

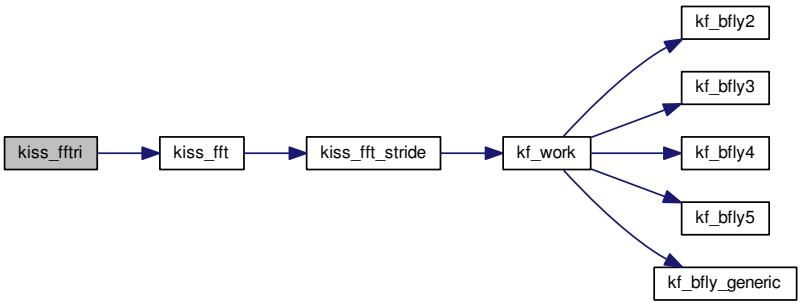
Definition at line 123 of file kiss_fftr.c.

References C_ADD, C_FIXDIV, C_MUL, C_SUB, kiss_fft_cpx::i, kiss_fft_state::inverse, kiss_fft(), kiss_fft_state::nfft, kiss_fft_cpx::r, kiss_fftr_state::substate, kiss_fftr_state::super_twiddles, and kiss_fftr_state::tmpbuf.

```
124 {
125     /* input buffer timedata is stored row-wise */
126     int k, ncfft;
127
128     if (st->substate->inverse == 0) {
129         fprintf (stderr, "kiss fft usage error: improper alloc\n");
```

```
130         exit (1);
131     }
132
133     ncfft = st->substate->nfft;
134
135     st->tmpbuf[0].r = freqdata[0].r + freqdata[ncfft].r;
136     st->tmpbuf[0].i = freqdata[0].r - freqdata[ncfft].r;
137     C_FIXDIV(st->tmpbuf[0],2);
138
139     for (k = 1; k <= ncfft / 2; ++k) {
140         kiss_fft_cpx fk, fnkc, fek, fok, tmp;
141         fk = freqdata[k];
142         fnkc.r = freqdata[ncfft - k].r;
143         fnkc.i = -freqdata[ncfft - k].i;
144         C_FIXDIV( fk , 2 );
145         C_FIXDIV( fnkc , 2 );
146
147         C_ADD (fek, fk, fnkc);
148         C_SUB (tmp, fk, fnkc);
149         C_MUL (fok, tmp, st->super_twiddles[k-1]);
150         C_ADD (st->tmpbuf[k],     fek, fok);
151         C_SUB (st->tmpbuf[ncfft - k], fek, fok);
152 #ifdef USE_SIMD
153         st->tmpbuf[ncfft - k].i *= _mm_set1_ps(-1.0);
154 #else
155         st->tmpbuf[ncfft - k].i *= -1;
156 #endif
157     }
158     kiss_fft (st->substate, st->tmpbuf, (kiss_fft_cpx *) timedata);
159 }
```

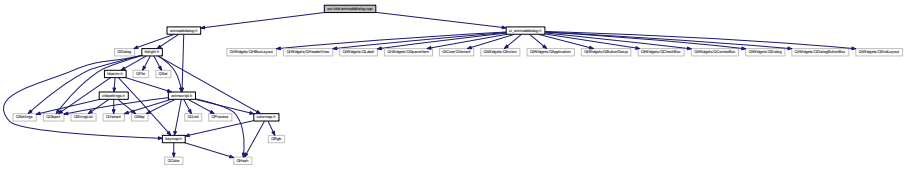
Here is the call graph for this function:



13.57 src/ckb/animadddialog.cpp File Reference

```
#include "animadddialog.h"
#include "ui_animadddialog.h"
```

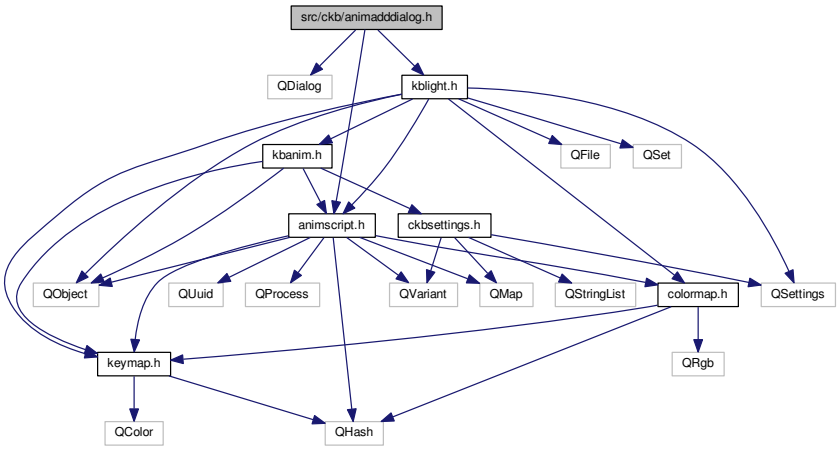
Include dependency graph for animadddialog.cpp:



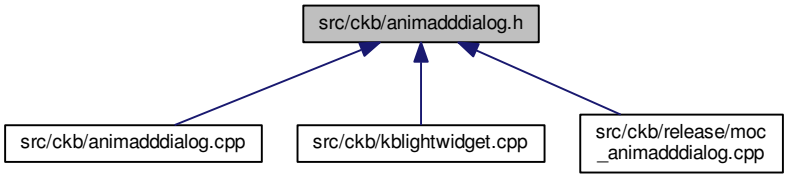
13.58 src/ckb/animadddialog.h File Reference

```
#include <QDialog>
#include "animscript.h"
#include "kblight.h"
```

Include dependency graph for animadddialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class AnimAddDialog

Namespaces

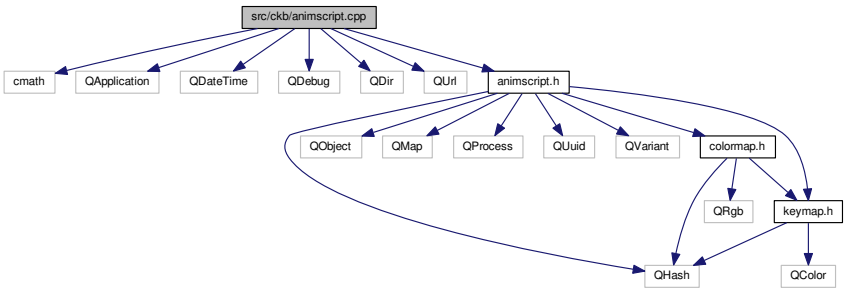
- Ui

13.59 src/ckb/animscript.cpp File Reference

```
#include <cmath>
```

```
#include <QApplication>
#include <QDateTime>
#include <QDebug>
#include <QDir>
#include <QUrl>
#include "animscript.h"
```

Include dependency graph for animscript.cpp:



Functions

- QString urlParam (const QString &param)

Variables

- static const double ONE_DAY = 24. * 60. * 60.

13.59.1 Function Documentation

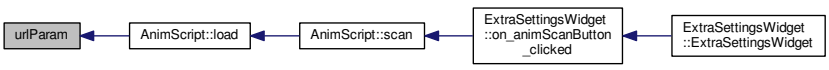
13.59.1.1 QString urlParam (const QString & *param*) [inline]

Definition at line 71 of file animscript.cpp.

Referenced by AnimScript::load().

```
71                                                {
72     return QUrl::fromPercentEncoding(param.trimmed().toLatin1()).trimmed();
73 }
```

Here is the caller graph for this function:



13.59.2 Variable Documentation

13.59.2.1 const double ONE_DAY = 24. * 60. * 60. [static]

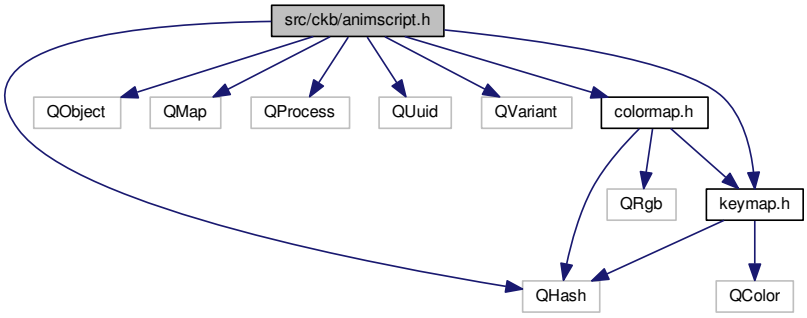
Definition at line 75 of file animscript.cpp.

Referenced by AnimScript::load().

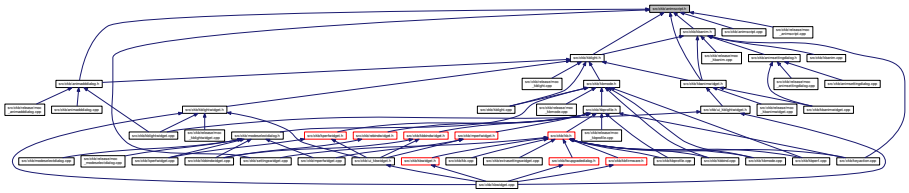
13.60 src/ckb/animscript.h File Reference

```
#include <QHash>
#include <QObject>
#include <QMap>
#include <QProcess>
#include <QUuid>
#include <QVariant>
#include "keymap.h"
#include "colormap.h"
```

Include dependency graph for animscript.h:



This graph shows which files directly or indirectly include this file:



Data Structures

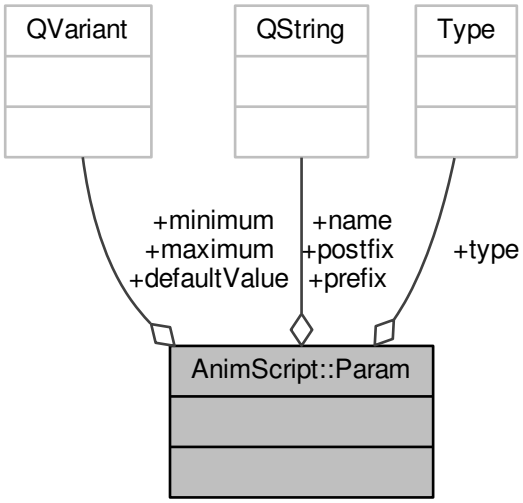
- class AnimScript
- struct AnimScript::Param

13.60.1 Data Structure Documentation

13.60.1.1 struct AnimScript::Param

Definition at line 21 of file animscript.h.

Collaboration diagram for AnimScript::Param:



Data Fields

	Type	

Data Fields

QVariant	defaultValue	
QVariant	maximum	
QVariant	minimum	
QString	name	
QString	postfix	
QString	prefix	
Type	type	

13.61 src/ckb/animsettingdialog.cpp File Reference

```
#include <cstdio>
#include <QDial>
#include <QDoubleSpinBox>
#include <QLineEdit>
#include <QComboBox>
#include "animsettingdialog.h"
#include "ui_animsettingdialog.h"
#include "colorbutton.h"
#include "gradientbutton.h"
```

Include dependency graph for animsettingdialog.cpp:

Functions

- static int angleFlip (int angle)

13.61.1 Function Documentation

13.61.1.1 static int angleFlip (int *angle*) [inline], [static]

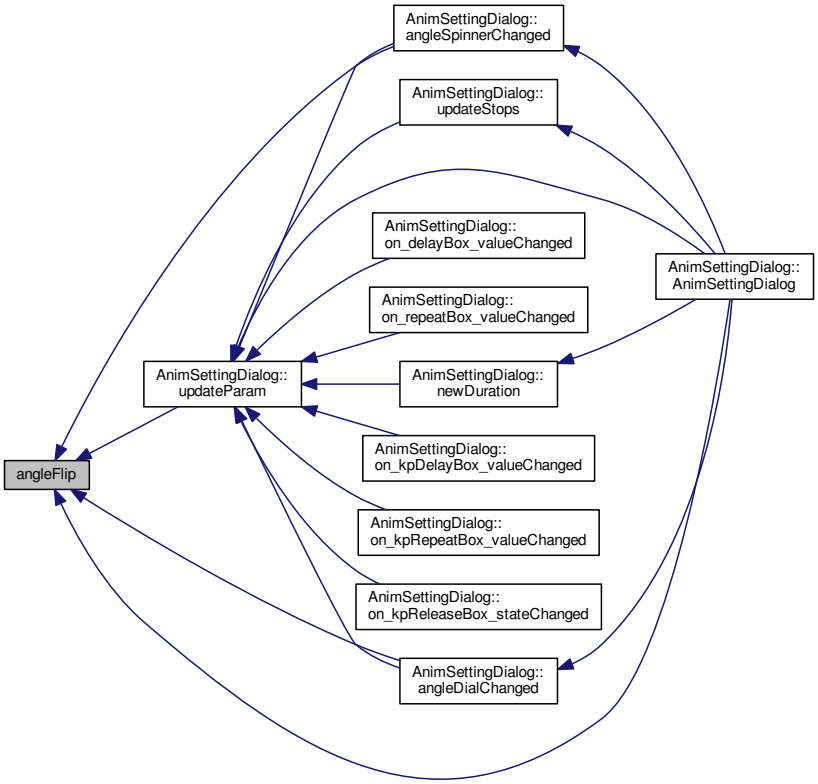
Definition at line 12 of file animsettingdialog.cpp.

References angle.

Referenced by AnimSettingDialog::angleDialChanged(), AnimSettingDialog::angleSpinnerChanged(), AnimSettingDialog::AnimSettingDialog(), and AnimSettingDialog::updateParam().

```
12                                      {
13     if(angle <= 180)
14         return 180 - angle;
15     else
16         return 360 - (angle - 180);
17 }
```

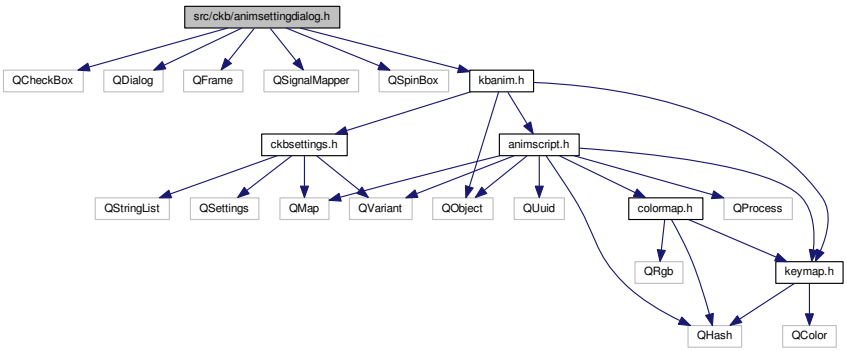
Here is the caller graph for this function:



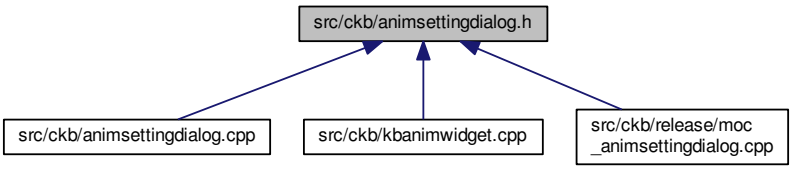
13.62 src/ckb/animsettingdialog.h File Reference

```
#include <QCheckBox>
#include <QDialog>
#include <QFrame>
#include <QSignalMapper>
#include <QSpinBox>
#include "kbanim.h"
```

Include dependency graph for animsettingdialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class AnimSettingDialog

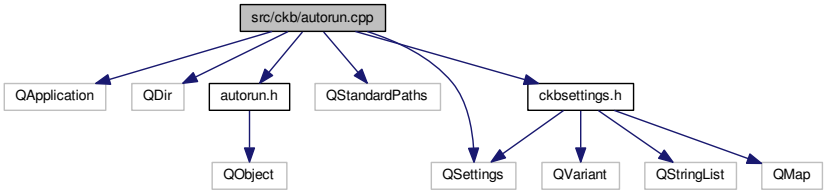
Namespaces

- Ui

13.63 src/ckb/autorun.cpp File Reference

```
#include <QApplication>
#include <QDir>
#include <QSettings>
#include <QStandardPaths>
#include "autorun.h"
#include "ckbsettings.h"
```

Include dependency graph for autorun.cpp:



Variables

- static const QString settingPath = "Program/DidLoginItem"

13.63.1 Variable Documentation

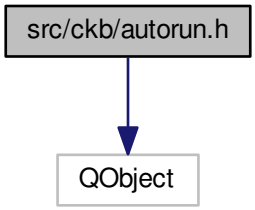
13.63.1.1 const QString settingPath = "Program/DidLoginItem" [static]

Definition at line 18 of file autorun.cpp.

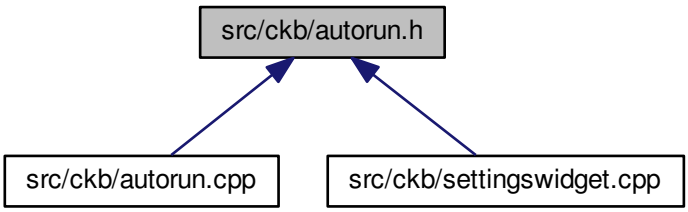
13.64 src/ckb/autorun.h File Reference

```
#include <QObject>
```

Include dependency graph for autorun.h:



This graph shows which files directly or indirectly include this file:



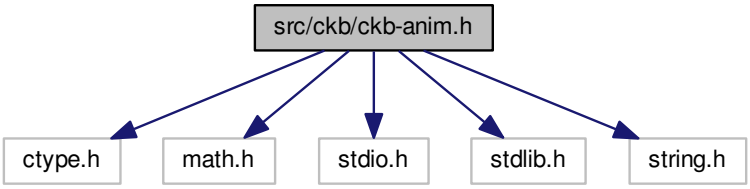
Data Structures

- class AutoRun

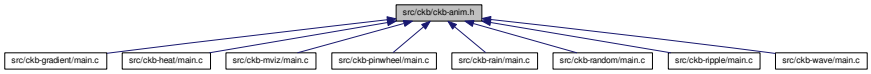
13.65 src/ckb/ckb-anim.h File Reference

```
#include <ctype.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for ckb-anim.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct ckb_key
- struct ckb_runctx
- struct ckb_gradient

Macros

- #define M_PI 3.14159265358979323846
- #define TRUE 1
- #define FALSE 0
- #define CKB_DOWN 1
- #define CKB_UP 0
- #define CKB_CONTAINER(macro) do { macro } while(0)
- #define CKB_GUID(guid) CKB_CONTAINER(printf("guid "); printurl(guid); printf("\n");)
- #define CKB_NAME(name) CKB_CONTAINER(printf("name "); printurl(name); printf("\n");)
- #define CKB_VERSION(version) CKB_CONTAINER(printf("version "); printurl(version); printf("\n");)
- #define CKB_COPYRIGHT(year, author) CKB_CONTAINER(printf("author "); printurl(author); printf("\nyear %s\n", year);)
- #define CKB_LICENSE(license) CKB_CONTAINER(printf("license "); printurl(license); printf("\n");)
- #define CKB_DESCRIPTION(description) CKB_CONTAINER(printf("description "); printurl(description); printf("\n");)
- #define CKB_PARAM(type, name, prefix, postfix, extra) CKB_CONTAINER(printf("param %s %s ", type, name); printurl(prefix); printf(" "); printurl(postfix); printf(" "); extra; printf("\n");)
- #define CKB_PARAM_LONG(name, prefix, postfix, default, min, max) CKB_PARAM("long", name, prefix, postfix, printf("%ld %ld %ld", (long)(default), (long)(min), (long)(max)))
- #define CKB_PARAM_DOUBLE(name, prefix, postfix, default, min, max) CKB_PARAM("double", name, prefix, postfix, printf("%lf %lf %lf", (double)(default), (double)(min), (double)(max)))
- #define CKB_PARAM_BOOL(name, text, default) CKB_PARAM("bool", name, text, "", printf((default) ? "1" : "0"))

- #define CKB_PARAM_RGB(name, prefix, postfix, r, g, b) CKB_PARAM("rgb", name, prefix, postfix, printf("%02x%02x%02x", (unsigned char)(r), (unsigned char)(g), (unsigned char)(b)))
- #define CKB_PARAM_ARGB(name, prefix, postfix, a, r, g, b) CKB_PARAM("argb", name, prefix, postfix, printf("%02x%02x%02x%02x", (unsigned char)(a), (unsigned char)(r), (unsigned char)(g), (unsigned char)(b)))
- #define CKB_PARAM_GRADIENT(name, prefix, postfix, default) CKB_PARAM("gradient", name, prefix, postfix, printurl(default))
- #define CKB_PARAM_AGRADIENT(name, prefix, postfix, default) CKB_PARAM("agradient", name, prefix, postfix, printurl(default))
- #define CKB_PARAM_ANGLE(name, prefix, postfix, default) CKB_PARAM("angle", name, prefix, postfix, printf("%ld", (long)(default)))
- #define CKB_PARAM_STRING(name, prefix, postfix, default) CKB_PARAM("string", name, prefix, postfix, printurl(default))
- #define CKB_PARAM_LABEL(name, text) CKB_PARAM("label", name, text, "",)
- #define CKB_PRESET_START(name) CKB_CONTAINER(printf("preset "); printurl(name);)
- #define CKB_PRESET_PARAM(name, value) CKB_CONTAINER(printf(" %s=", name); printurl(value);)
- #define CKB_PRESET_END CKB_CONTAINER(printf("\n");)
- #define CKB_KP_NONE "none"
- #define CKB_KP_NAME "name"
- #define CKB_KP_POSITION "position"
- #define CKB_KPMODE(mode) CKB_CONTAINER(printf("kpmode %s\n", mode);)
- #define CKB_TIME_DURATION "duration"
- #define CKB_TIME_ABSOLUTE "absolute"
- #define CKB_TIMEMODE(mode) CKB_CONTAINER(printf("time %s\n", mode);)
- #define CKB_REPEAT(enable) CKB_CONTAINER(printf("repeat %s\n", (enable) ? "on" : "off");)
- #define CKB_PREEMPT(enable) CKB_CONTAINER(printf("preempt %s\n", (enable) ? "on" : "off");)
- #define CKB_LIVEPARAMS(enable) CKB_CONTAINER(printf("parammode %s\n", (enable) ? "live" : "static");)
- #define CKB_PARSE_LONG(param_name, value_ptr) if(!strcmp(name, param_name) && sscanf(value, "%ld", value_ptr) == 1)
- #define CKB_PARSE_DOUBLE(param_name, value_ptr) if(!strcmp(name, param_name) && sscanf(value, "%lf", value_ptr) == 1)
- #define CKB_PARSE_BOOL(param_name, value_ptr) if(!strcmp(name, param_name) && sscanf(value, "%u", value_ptr) == 1)
- #define CKB_PARSE_RGB(param_name, r_ptr, g_ptr, b_ptr) if(!strcmp(name, param_name) && sscanf(value, "%2hhx%2hhx%2hhx", r_ptr, g_ptr, b_ptr) == 3)
- #define CKB_PARSE_ARGB(param_name, a_ptr, r_ptr, g_ptr, b_ptr) if(!strcmp(name, param_name) && sscanf(value, "%2hhx%2hhx%2hhx%2hhx", a_ptr, r_ptr, g_ptr, b_ptr) == 4)
- #define CKB_PARSE_GRADIENT(param_name, gradient_ptr) if(!strcmp(name, param_name) && ckb_scan_grad(value, gradient_ptr, 0))
- #define CKB_PARSE_AGRADIENT(param_name, gradient_ptr) if(!strcmp(name, param_name) && ckb_scan_grad(value, gradient_ptr, 1))
- #define CKB_PARSE_ANGLE(param_name, value_ptr) if(!strcmp(name, param_name) && sscanf(value, "%ld", value_ptr) == 1)
- #define CKB_PARSE_STRING(param_name) if(!strcmp(name, param_name))
- #define CKB_REAL_ANGLE(angle) fmod((-(angle) + 90.) * M_PI / 180. + M_PI * 2., M_PI * 2.)
- #define CKB_KEYNAME_MAX 12
- #define CKB_KEYCLEAR(context) CKB_CONTAINER(ckb_key* key = context->keys; unsigned count = context->keycount; unsigned i = 0; for(; i < count; i++) key[i].a = key[i].r = key[i].g = key[i].b = 0;)
- #define CKB_GRAD_MAX 100
- #define CKB_MAX_WORD (4 * 1024)

Functions

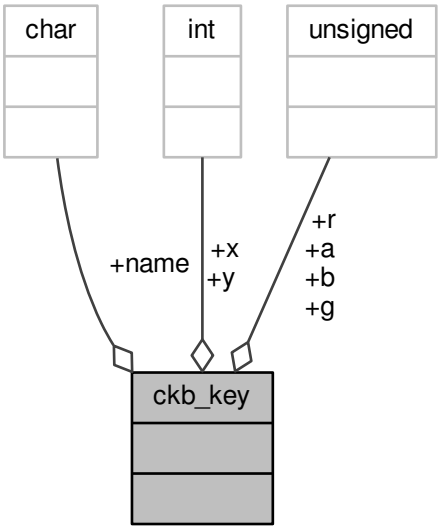
- int ckb_scan_grad (const char *string, ckb_gradient *gradient, int alpha)
- void ckb_grad_color (float *a, float *r, float *g, float *b, const ckb_gradient *grad, float pos)
- void ckb_alpha_blend (ckb_key *key, float a, float r, float g, float b)
- void printurl (const char *src)
- void urldecode (char *dst, const char *src)
- void ckb_getline (char word1[(4 *1024)], char word2[(4 *1024)], char word3[(4 *1024)])
- void ckb_info ()
- void ckb_init (ckb_runctx *context)
- void ckb_parameter (ckb_runctx *, const char *, const char *)
- void ckb_keypress (ckb_runctx *, ckb_key *, int, int, int)
- void ckb_start (ckb_runctx *, int)
- void ckb_time (ckb_runctx *, double)
- int ckb_frame (ckb_runctx *)
- void ckb_read_params (ckb_runctx *ctx)
- int main (int argc, char *argv[])

13.65.1 Data Structure Documentation

13.65.1.1 struct ckb_key

Definition at line 131 of file ckb-anim.h.

Collaboration diagram for ckb_key:



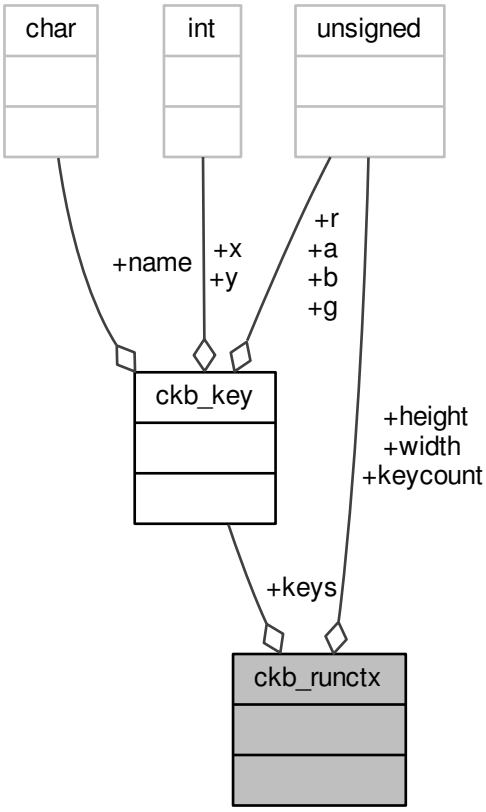
Data Fields

unsigned char	a	
unsigned char	b	
unsigned char	g	
char	name[12]	
unsigned char	r	
int	x	
int	y	

13.65.1.2 struct ckb_runctx

Definition at line 138 of file ckb-anim.h.

Collaboration diagram for ckb_runctx:



Data Fields

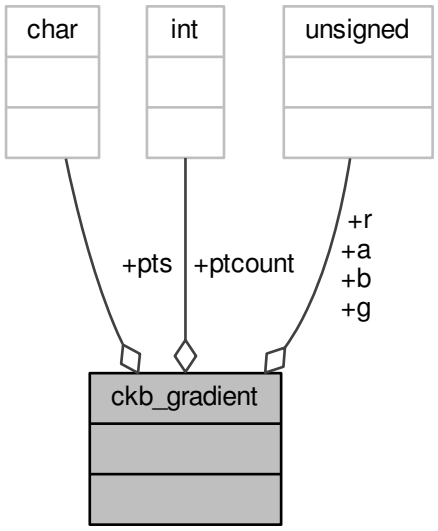
unsigned	height	
unsigned	keycount	
ckb_key *	keys	

unsigned	width	

13.65.1.3 struct ckb_gradient

Definition at line 152 of file ckb-anim.h.

Collaboration diagram for ckb_gradient:



Data Fields

unsigned char	a[100]	
unsigned char	b[100]	
unsigned char	g[100]	
int	ptcount	
char	pts[100]	
unsigned char	r[100]	

13.65.2 Macro Definition Documentation

13.65.2.1 #define CKB_CONTAINER(*macro*) do { macro } while(0)

Definition at line 57 of file ckb-anim.h.

13.65.2.2 #define CKB_COPYRIGHT(*year, author*) CKB_CONTAINER(printf("author "); printurl(author); printf("\nyear %s\n", year);)

Definition at line 66 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.3 #define CKB_DESCRIPTION(*description*) CKB_CONTAINER(printf("description "); printurl(description); printf("\n");)

Definition at line 70 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.4 #define CKB_DOWN 1

Definition at line 52 of file ckb-anim.h.

13.65.2.5 #define CKB_GRAD_MAX 100

Definition at line 151 of file ckb-anim.h.

13.65.2.6 #define CKB_GUID(*guid*) CKB_CONTAINER(printf("guid "); printurl(guid); printf("\n");)

Definition at line 60 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.7 #define CKB_KEYCLEAR(*context*) CKB_CONTAINER(ckb_key* key = context->keys; unsigned count = context->keycount; unsigned i = 0; for(; i < count; i++) key[i].a = key[i].r = key[i].g = key[i].b = 0;)

Definition at line 148 of file ckb-anim.h.

Referenced by ckb_frame().

13.65.2.8 #define CKB_KEYNAME_MAX 12

Definition at line 130 of file ckb-anim.h.

Referenced by main().

13.65.2.9 #define CKB_KP_NAME "name"

Definition at line 91 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.10 #define CKB_KP_NONE "none"

Definition at line 90 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.11 #define CKB_KP_POSITION "position"

Definition at line 92 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.12 #define CKB_KPMODE(*mode*) CKB_CONTAINER(printf("kpmode %s\n", mode);)

Definition at line 93 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.13 #define CKB_LICENSE(*license*) CKB_CONTAINER(printf("license "); printurl(license); printf("\n");)

Definition at line 68 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.14 #define CKB_LIVEPARAMS(*enable*) CKB_CONTAINER(printf("parammode %s\n", (enable) ? "live" : "static");)

Definition at line 104 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.15 #define CKB_MAX_WORD (4 * 1024)

Definition at line 238 of file ckb-anim.h.

Referenced by ckb_read_params(), and main().

13.65.2.16 #define CKB_NAME(*name*) CKB_CONTAINER(printf("name "); printurl(name); printf("\n");)

Definition at line 62 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.17 #define CKB_PARAM(*type, name, prefix, postfix, extra*) CKB_CONTAINER(printf("param %s %s ", type, name); printurl(prefix); printf(" "); printurl(postfix); printf(" "); extra; printf("\n");)

Definition at line 73 of file ckb-anim.h.

13.65.2.18 #define CKB_PARAM_AGRADIENT(*name, prefix, postfix, default*) CKB_PARAM("agradient", name, prefix, postfix, printurl(default))

Definition at line 80 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.19 #define CKB_PARAM_ANGLE(*name, prefix, postfix, default*) CKB_PARAM("angle", name, prefix, postfix, printf("%ld", (long)(default)))

Definition at line 81 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.20 #define CKB_PARAM_ARGB(*name, prefix, postfix, a, r, g, b*) CKB_PARAM("argb", name, prefix, postfix, printf("%02x%02x%02x%02x", (unsigned char)(a), (unsigned char)(r), (unsigned char)(g), (unsigned char)(b)))

Definition at line 78 of file ckb-anim.h.

13.65.2.21 #define CKB_PARAM_BOOL(*name, text, default*) **CKB_PARAM**("bool", name, text, "", printf((default) ? "1" : "0"))

Definition at line 76 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.22 #define CKB_PARAM_DOUBLE(*name, prefix, postfix, default,* **min**, **max**) **CKB_PARAM**("double", name, prefix, postfix, printf("%lf %lf %lf", (double)(default), (double)(**min**), (double)(**max**)))

Definition at line 75 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.23 #define CKB_PARAM_GRADIENT(*name, prefix, postfix, default*) **CKB_PARAM**("gradient", name, prefix, postfix, **printurl**(default))

Definition at line 79 of file ckb-anim.h.

13.65.2.24 #define CKB_PARAM_LABEL(*name, text*) **CKB_PARAM**("label", name, text, "",)

Definition at line 83 of file ckb-anim.h.

13.65.2.25 #define CKB_PARAM_LONG(*name, prefix, postfix, default,* **min**, **max**) **CKB_PARAM**("long", name, prefix, postfix, printf("%ld %ld %ld", (long)(default), (long)(**min**), (long)(**max**)))

Definition at line 74 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.26 #define CKB_PARAM_RGB(*name, prefix, postfix, r, g, b*) **CKB_PARAM**("rgb", name, prefix, postfix, printf("%02x%02x%02x", (unsigned char)(r), (unsigned char)(g), (unsigned char)(b)))

Definition at line 77 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.27 #define CKB_PARAM_STRING(*name, prefix, postfix, default*) **CKB_PARAM**("string", name, prefix, postfix, **printurl**(default))

Definition at line 82 of file ckb-anim.h.

13.65.2.28 #define CKB_PARSE_AGRADIENT(*param_name, gradient_ptr*) if(!strcmp(name, param_name) && **ckb_scan_grad**(value, gradient_ptr, 1))

Definition at line 121 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.29 #define CKB_PARSE_ANGLE(*param_name, value_ptr*) if(!strcmp(name, param_name) && sscanf(value, "%ld", value_ptr) == 1)

Definition at line 122 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.30 #define CKB_PARSE_ARGB(*param_name, a_ptr, r_ptr, g_ptr, b_ptr*) if(!strcmp(name, param_name) && sscanf(value, "%2hhx%2hhx%2hhx%2hhx", a_ptr, r_ptr, g_ptr, b_ptr) == 4)

Definition at line 119 of file ckb-anim.h.

13.65.2.31 #define CKB_PARSE_BOOL(*param_name, value_ptr*) if(!strcmp(name, param_name) && sscanf(value, "%u", value_ptr) == 1)

Definition at line 117 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.32 #define CKB_PARSE_DOUBLE(*param_name, value_ptr*) if(!strcmp(name, param_name) && sscanf(value, "%lf", value_ptr) == 1)

Definition at line 116 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.33 #define CKB_PARSE_GRADIENT(*param_name, gradient_ptr*) if(!strcmp(name, param_name) && **ckb_scan_grad**(value, gradient_ptr, 0))

Definition at line 120 of file ckb-anim.h.

13.65.2.34 #define CKB_PARSE_LONG(*param_name, value_ptr*) if(!strcmp(name, param_name) && sscanf(value, "%ld", value_ptr) == 1)

Definition at line 115 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.35 #define CKB_PARSE_RGB(*param_name, r_ptr, g_ptr, b_ptr*) if(!strcmp(name, param_name) && sscanf(value, "%2hhx%2hhx%2hhx", r_ptr, g_ptr, b_ptr) == 3)

Definition at line 118 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.36 #define CKB_PARSE_STRING(*param_name*) if(!strcmp(name, param_name))

Definition at line 123 of file ckb-anim.h.

13.65.2.37 #define CKB_PREEMPT(*enable*) **CKB_CONTAINER**(printf("preempt %s\n", (enable) ? "on" : "off");)

Definition at line 102 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.38 #define CKB_PRESET_END **CKB_CONTAINER**(printf("\n");)

Definition at line 87 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.39 #define CKB_PRESET_PARAM(*name, value*) CKB_CONTAINER(printf(" %s=", name); printurl(value);)

Definition at line 86 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.40 #define CKB_PRESET_START(*name*) CKB_CONTAINER(printf("preset "); printurl(name);)

Definition at line 85 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.41 #define CKB_REAL_ANGLE(angle) fmod((-(angle) + 90.) * M_PI / 180. + M_PI * 2., M_PI * 2.)

Definition at line 127 of file ckb-anim.h.

Referenced by ckb_parameter().

13.65.2.42 #define CKB_REPEAT(*enable*) CKB_CONTAINER(printf("repeat %s\n", (enable) ? "on" : "off");)

Definition at line 99 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.43 #define CKB_TIME_ABSOLUTE "absolute"

Definition at line 96 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.44 #define CKB_TIME_DURATION "duration"

Definition at line 95 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.45 #define CKB_TIMEMODE(*mode*) CKB_CONTAINER(printf("time %s\n", mode);)

Definition at line 97 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.46 #define CKB_UP 0

Definition at line 53 of file ckb-anim.h.

13.65.2.47 #define CKB_VERSION(*version*) CKB_CONTAINER(printf("version "); printurl(version); printf("\n");)

Definition at line 64 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.48 #define FALSE 0

Definition at line 49 of file ckb-anim.h.

Referenced by ckb_info().

13.65.2.49 #define M_PI 3.14159265358979323846

Definition at line 42 of file ckb-anim.h.

Referenced by ckb_frame(), and ckb_parameter().

13.65.2.50 #define TRUE 1

Definition at line 46 of file ckb-anim.h.

Referenced by ckb_info().

13.65.3 Function Documentation

13.65.3.1 void ckb_alpha_blend (ckb_key * *key,* float *a,* float *r,* float *g,* float *b*)

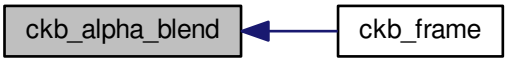
Definition at line 283 of file ckb-anim.h.

References ckb_key::a, ckb_key::b, ckb_key::g, and ckb_key::r.

Referenced by ckb_frame().

```
283                                                                          {
284     a /= 255.f;
285     float ka = key->a / 255.f;
286     float a2 = a + (1.f - a) * ka;
287     if(a2 == 0.f){
288         key->a = key->r = key->g = key->b = 0;
289         return;
290     }
291     key->a = round(a2 * 255.f);
292     key->r = round((r * a + key->r * ka * (1.f - a)) / a2);
293     key->g = round((g * a + key->g * ka * (1.f - a)) / a2);
294     key->b = round((b * a + key->b * ka * (1.f - a)) / a2);
295 }
```

Here is the caller graph for this function:



13.65.3.2 int ckb_frame (ckb_runctx *)

Definition at line 137 of file main.c.

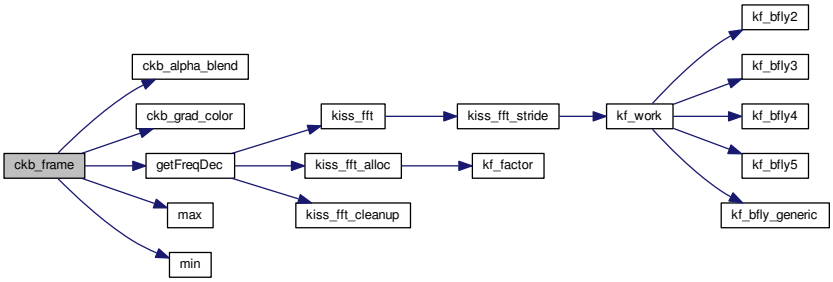
References ckb_key::a, aa, ab, ag, angle, ANGLE, anim, ANIM_MAX, animcolor, animlength, ar, ckb_key::b, ckb_alpha_blend(), ckb_grad_color(), CKB_KEYCLEAR, counter_clock, drop, DROP_MAX, ffade, frame, ckb_key::g, getFreqDec(), keyAnim::gradient, ckb_runctx::height, HOLD, ckb_runctx::keycount, ckb_runctx::keys, M_PI, max(),

min(), phase, power, powers, pressestofull, ckb_key::r, randomBright, randomize, symmetric, keyAnim::target, width, ckb_runctx::width, x, and y.

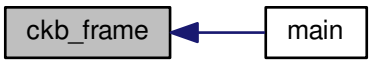
Referenced by main().

```
137                                 {
138     // Draw key colors
139     unsigned count = context->keycount;
140     for(unsigned i = 0; i < count; i++){
141         float phase = anim[i].target;
142 
143         if(phase == HOLD)
144             phase = 0.f;
145         else if(phase < 0.f)
146             phase = 1.f;
147         ckb_key* key = context->keys + i;
148         float a, r, g, b;
149         ckb_gradient thisGradient;
150         if (randomize) {
151             thisGradient = anim[i].gradient;
152         } else {
153             thisGradient = animcolor;
154         }
155         ckb_grad_color(&a, &r, &g, &b, &thisGradient, phase * 100.);
156         key->a = a;
157         key->r = r;
158         key->g = g;
159         key->b = b;
160     }
161     return 0;
162 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.65.3.3 void ckb_getline (char *word1[(4 ∗ 1024)]*, char *word2[(4 ∗ 1024)]*, char *word3[(4 ∗ 1024)]*)

Definition at line 239 of file ckb-anim.h.

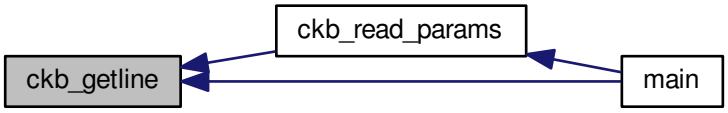
References urldecode().

Referenced by ckb_read_params(), and main().

```
239                                                                                                 {
240     char line[CKB_MAX_WORD * 3 + 3];
241     fgets(line, sizeof(line), stdin);
242     word1[0] = word1[1] = word1[2] = 0;
243     int res = sscanf(line, "%s %s %s", word1, word2, word3);
244     if(res >= 1)
245         urldecode(word1, word1);
246     if(res >= 2)
247         urldecode(word2, word2);
248     if(res >= 3)
249         urldecode(word3, word3);
250     line[strlen(line) - 1] = 0;
251 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.65.3.4 void ckb_grad_color (float * *a*, float * *r*, float * *g*, float * *b*, const ckb_gradient * *grad*, float *pos*)

Definition at line 254 of file ckb-anim.h.

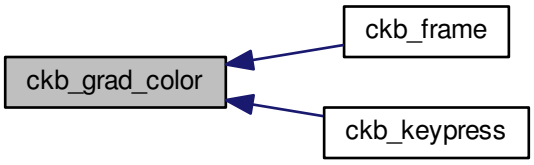
References ckb_gradient::a, ckb_gradient::b, ckb_gradient::g, ckb_gradient::ptcount, ckb_gradient::pts, and ckb_-gradient::r.

Referenced by ckb_frame(), and ckb_keypress().

```
254                                                                                                 {
255     // Find the points surrounding this position
256     int count = grad->ptcount;
257     if(count == 0){
258         *a = *r = *g = *b = 0.f;
259         return;
260     }
261     int i = 1;
262     for(; i < count; i++){
263         if(grad->pts[i] >= pos)
264             break;
265     }
266     // Get color by linear interpolation. Premultiply the alpha value so that it returns the expected color
267     // (i.e. stops with zero opacity won't contribute to color)
```

```
268     float distance = grad->pts[i] - grad->pts[i - 1];
269     float dx = (pos - grad->pts[i - 1]) / distance;
270     float a1 = grad->a[i] / 255., a2 = grad->a[i - 1] / 255.;
271     float a3 = *a = a1 * dx + a2 * (1.f - dx);
272     if(a3 == 0.){
273         *a = *r = *g = *b = 0.f;
274         return;
275     }
276     *a *= 255.f;
277     *r = (grad->r[i] * a1 * dx + grad->r[i - 1] * a2 * (1.f - dx)) / a3;
278     *g = (grad->g[i] * a1 * dx + grad->g[i - 1] * a2 * (1.f - dx)) / a3;
279     *b = (grad->b[i] * a1 * dx + grad->b[i - 1] * a2 * (1.f - dx)) / a3;
280 }
```

Here is the caller graph for this function:



13.65.3.5 void ckb_info ( )

Definition at line 5 of file main.c.

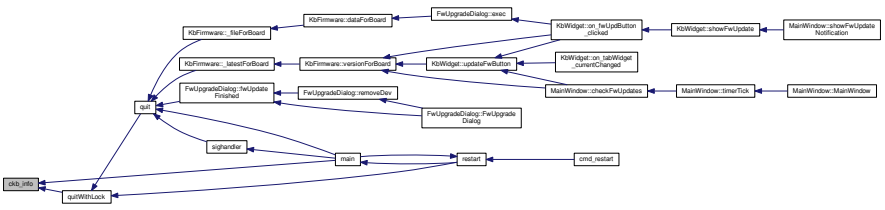
References CKB_COPYRIGHT, CKB_DESCRIPTION, CKB_GUID, CKB_KP_NAME, CKB_KP_NONE, CKB_KP-_POSITION, CKB_KPMODE, CKB_LICENSE, CKB_LIVEPARAMS, CKB_NAME, CKB_PARAM_AGRADIENT, C-KB_PARAM_ANGLE, CKB_PARAM_BOOL, CKB_PARAM_DOUBLE, CKB_PARAM_LONG, CKB_PARAM_RGB, CKB_PREEMPT, CKB_PRESET_END, CKB_PRESET_PARAM, CKB_PRESET_START, CKB_REPEAT, CKB_T-IME_ABSOLUTE, CKB_TIME_DURATION, CKB_TIMEMODE, CKB_VERSION, FALSE, and TRUE.

Referenced by main(), and quitWithLock().

```
5               {
6     // Plugin info
7     CKB_NAME("Gradient");
8     CKB_VERSION("0.9");
9     CKB_COPYRIGHT("2014-2016", "MSC");
10     CKB_LICENSE("GPLv2");
11     CKB_GUID("{54DD2975-E192-457D-BCFC-D912A24E33B4}");
12     CKB_DESCRIPTION("A transition from one color to another.");
13
14     // Effect parameters
15     CKB_PARAM_AGRADIENT("color", "Color:", "", "ffffffff");
16     CKB_PARAM_BOOL("kphold", "Freeze until key is released", TRUE);
17     CKB_PARAM_BOOL("randomize", "Randomly select from gradient", 0);
18
19     // Timing/input parameters
20     CKB_KPMODE(CKB_KP_NAME);
21     CKB_TIMEMODE(CKB_TIME_DURATION);
22     CKB_LIVEPARAMS(TRUE);
23
24     // Presets
25     CKB_PRESET_START("Fade in");
26     CKB_PRESET_PARAM("color", "ff000000");
27     CKB_PRESET_PARAM("duration", "1.0");
28     CKB_PRESET_PARAM("stop", "0");
29     CKB_PRESET_PARAM("kpstop", "0");
30     CKB_PRESET_END;
31
32     CKB_PRESET_START("Rainbow");
33     CKB_PRESET_PARAM("color", "0:ffff0000 17:ffffff00 33:ff00ff00 50:ff00ffff 67:ff0000ff
       83:ffff00ff 100:ffff0000");
```

```
34     CKB_PRESET_PARAM("duration", "2.0");
35     CKB_PRESET_END;
36
37     CKB_PRESET_START("Type lighting");
38     CKB_PRESET_PARAM("duration", "1.0");
39     CKB_PRESET_PARAM("trigger", "0");
40     CKB_PRESET_PARAM("kptrigger", "1");
41     CKB_PRESET_END;
42 }
```

Here is the caller graph for this function:



13.65.3.6 void ckb_init (ckb_runctx * *context*)

Definition at line 57 of file main.c.

References ckb_runctx::height, kbsize, ckb_runctx::keycount, ckb_runctx::keys, NONE, pas, ckb_runctx::width, keyanim::x, ckb_key::x, keyanim::y, and ckb_key::y.

Referenced by main().

```
57                                  {
58     // Initialize all keys to 100% (animation over)
59
60     srand((unsigned)time(NULL));
61     unsigned count = context->keycount;
62     anim =  malloc(count * sizeof *anim);
63     for(unsigned i = 0; i < count; i++) {
64         anim[i].target = NONE;
65     }
66 }
```

Here is the caller graph for this function:

13.65.3.7 void ckb_keypress (ckb_runctx * , ckb_key * , int , int , int)

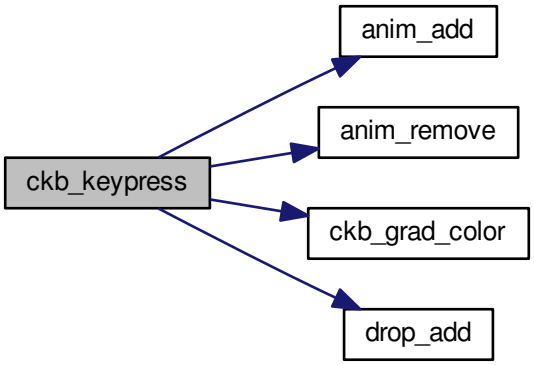
Definition at line 75 of file main.c.

References ckb_gradient::a, anim_add(), anim_remove(), ckb_gradient::b, ckb_grad_color(), drop_add(), ckb_gradient::g, keyAnim::gradient, ckb_runctx::height, HOLD, ckb_runctx::keys, kphold, kprelease, NONE, ckb_gradient::ptcount, ckb_gradient::pts, ckb_gradient::r, randomize, keyAnim::target, and ckb_runctx::width.

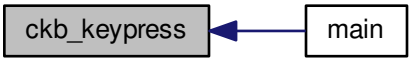
Referenced by main().

```
75                                                                                  {
76     // Start or stop animation on key
77     int i = key - context->keys;
78     if(state){
79         if (randomize) {
80             float a, r, g, b;
81             float colorChoice = (float)rand()/(float)(RAND_MAX);
82             ckb_grad_color(&a, &r, &g, &b, &animcolor, colorChoice * 100.);
83             ckb_gradient newGradient = { 0 };
84             newGradient.ptcount = 2;
85             newGradient.pts[0] = 0;
86             newGradient.pts[1] = 100;
87             newGradient.a[0] = 255;
88             newGradient.b[0] = b;
89             newGradient.g[0] = g;
90             newGradient.r[0] = r;
91             newGradient.a[1] = 0;
92             newGradient.b[1] = b;
93             newGradient.g[1] = g;
94             newGradient.r[1] = r;
95
96             anim[i].gradient = newGradient;
97         }
98
99         if(kphold) {
100             anim[i].target = HOLD;
101         } else {
102             anim[i].target = 0.f;
103         }
104     } else {
105         if(kprelease) {
106             anim[i].target = NONE;
107         }
108         else if(kphold) {
109             anim[i].target = 0.f;
110         }
111     }
112 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.65.3.8 void ckb_parameter (ckb_runctx * , const char * , const char *)

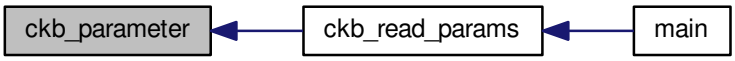
Definition at line 68 of file main.c.

References aa, ab, ag, angle, animlength, ar, CKB_PARSE_AGRADIENT, CKB_PARSE_ANGLE, CKB_PARSE_BOOL, CKB_PARSE_DOUBLE, CKB_PARSE_LONG, CKB_PARSE_RGB, CKB_REAL_ANGLE, counter_clock, fadein, ffade, ckb_runctx::height, kphold, kprelease, left, M_PI, maxsize, period, power, pressestofull, randomBright, randomize, spawn, speed, symmetric, top, useopacity, width, ckb_runctx::width, x, x_offset, y, and y_offset.

Referenced by ckb_read_params().

```
68                                                                      {
69     CKB_PARSE_AGRADIENT("color", &animcolor){}
70     CKB_PARSE_BOOL("kphold", &kphold){}
71     CKB_PARSE_BOOL("kprelease", &kprelease){}
72     CKB_PARSE_BOOL("randomize", &randomize){}
73 }
```

Here is the caller graph for this function:



13.65.3.9 void ckb_read_params (ckb_runctx * *ctx*)

Definition at line 349 of file ckb-anim.h.

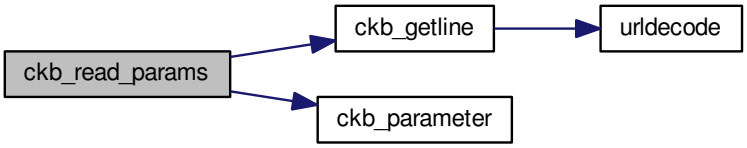
References ckb_getline(), CKB_MAX_WORD, and ckb_parameter().

Referenced by main().

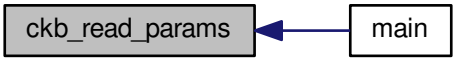
```
349                                     {
350     char cmd[CKB_MAX_WORD], param[CKB_MAX_WORD], value[
      CKB_MAX_WORD];
351     do {
352         ckb_getline(cmd, param, value);
353         if(!*cmd){
354             printf("Error [ckb-main]: Reached EOF reading parameters");
355             return;
356         }
357         if(!strcmp(cmd, "end") && !strcmp(param, "params"))
358             break;
359         if(strcmp(cmd, "param"))
```

```
360             continue;
361         ckb_parameter(ctx, param, value);
362     } while(1);
363 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.65.3.10 int ckb_scan_grad (const char * *string*, ckb_gradient * *gradient*, int *alpha*)

Definition at line 298 of file ckb-anim.h.

References ckb_gradient::a, ckb_gradient::b, ckb_gradient::g, ckb_gradient::ptcount, ckb_gradient::pts, and ckb_-gradient::r.

```
298                                                                 {
299     char pos = -1;
300     unsigned char a = 255, r, g, b;
301     int count = 0;
302     while(1){
303         int scanned = 0;
304         char newpos;
305         if(sscanf(string, "%hhd:%2hhx%2hhx%2hhx%2hhx%n", &newpos, &a, &r, &g, &b, &scanned) != 5)
306             break;
307         string += scanned;
308         // Don't allow stops out-of-order or past 100
309         if(newpos <= pos || newpos > 100)
310             return 0;
311         pos = newpos;
312         if(!alpha)
313             a = 255;
314         gradient->pts[count] = pos;
315         gradient->a[count] = a;
316         gradient->r[count] = r;
317         gradient->g[count] = g;
318         gradient->b[count] = b;
319         count++;
320     }
321     if(count == 0){
322         // If nothing was read, try a single ARGB constant.
323         if(sscanf(string, "%2hhx%2hhx%2hhx%2hhx", &a, &r, &g, &b) != 4)
324             return 0;
325         count = 2;
```

```
326         gradient->pts[0] = 0;
327         gradient->pts[1] = 100;
328         gradient->a[0] = a;
329         gradient->a[1] = 0;
330         gradient->r[0] = gradient->r[1] = r;
331         gradient->g[0] = gradient->g[1] = g;
332         gradient->b[0] = gradient->b[1] = b;
333     }
334     if(count < 2)
335         return 0;
336     gradient->ptcount = count;
337     return 1;
338 }
```

13.65.3.11 void ckb_start (ckb_runctx *, int)

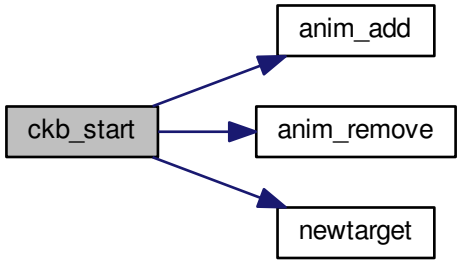
Definition at line 114 of file main.c.

References rgb::a, ckb_key::a, anim_add(), anim_remove(), rgb::b, ckb_key::b, fadein, frame, rgb::g, ckb_key::g, ckb_runctx::height, ckb_runctx::keycount, ckb_runctx::keys, kprelease, left, newtarget(), NONE, phase, rgb::r, ckb_key::r, keyAnim::target, tick, top, ckb_runctx::width, x, x_offset, y, and y_offset.

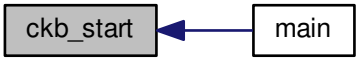
Referenced by main().

```
114                                                 {
115     // Start/stop all keys
116     unsigned count = context->keycount;
117     if(state) {
118         memset(anim, 0, count * sizeof *anim);
119     } else {
120         for(unsigned i = 0; i < count; i++) {
121             anim[i].target = NONE;
122         }
123     }
124 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.65.3.12 void ckb_time (ckb_runctx * , double)

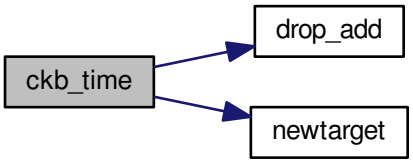
Definition at line 126 of file main.c.

References anim, ANIM_MAX, animlength, current, drop, drop_add(), DROP_MAX, frame, ckb_runctx::height, kbsize, ckb_runctx::keycount, maxsize, newtarget(), period, phase, spawn, speed, target, keyAnim::target, tick, keyanim::timing, keyanim::usages, width, and ckb_runctx::width.

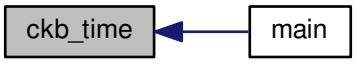
Referenced by main().

```
126                                                 {
127     // Advance animation on each key
128     unsigned count = context->keycount;
129     for(unsigned i = 0; i < count; i++){
130         float phase = anim[i].target;
131         if(phase > 1.f || phase < 0.f)
132             continue;
133         anim[i].target = phase + delta;
134     }
135 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.65.3.13 int main (int *argc*, char * *argv[]*)

Definition at line 365 of file ckb-anim.h.

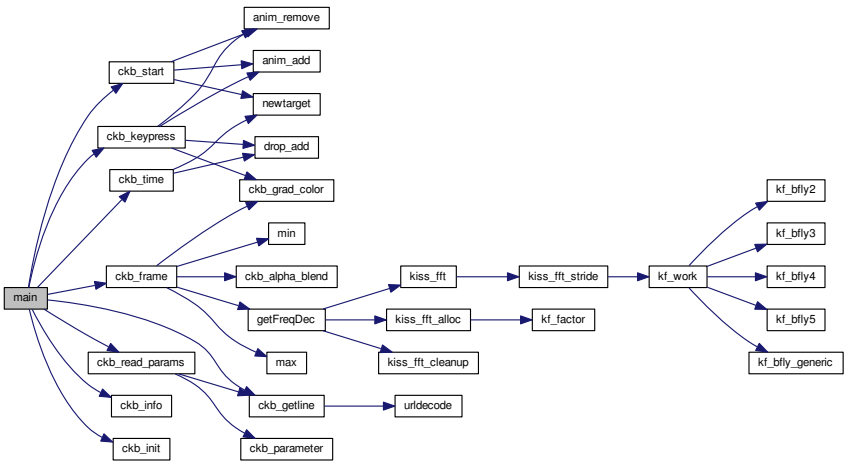
References ckb_key::a, ckb_key::b, ckb_frame(), ckb_getline(), ckb_info(), ckb_init(), CKB_KEYNAME_MAX, ckb_keypress(), CKB_MAX_WORD, ckb_read_params(), ckb_start(), ckb_time(), ckb_key::g, ckb_runctx::height, ckb_runctx::keycount, ckb_runctx::keys, ckb_key::name, ckb_key::r, ckb_runctx::width, x, ckb_key::x, y, and ckb_key::y.

```
365                                 {
366     if(argc == 2){
367         if(!strcmp(argv[1], "--ckb-info")){
368             ckb_info();
369             fflush(stdout);
```

```
370             return 0;
371         } else if(!strcmp(argv[1], "--ckb-run")){
372             ckb_runctx ctx;
373             // Read the keymap lines
374             char cmd[CKB_MAX_WORD], param[CKB_MAX_WORD], value[
    CKB_MAX_WORD];
375             // Skip anything up until "begin keymap"
376             do {
377                 ckb_getline(cmd, param, value);
378                 if(!*cmd){
379                     // If end-of-file is reached, abort
380                     printf("Error [ckb-main]: Reached EOF looking for \"begin keymap\"");
381                     return -2;
382                 }
383             } while(strcmp(cmd, "begin") || strcmp(param, "keymap"));
384             ckb_getline(cmd, param, value);
385             unsigned keycount;
386             if(strcmp(cmd, "keycount") || sscanf(param, "%u", &keycount) != 1 || keycount == 0){
387                 // If keycount isn't the next line, something is wrong
388                 printf("Error [ckb-main]: \"begin keymap\" not followed with \"keycount\"");
389                 return -3;
390             }
391             ctx.keys = (ckb_key*)calloc(keycount, sizeof(ckb_key));
392             ctx.keycount = keycount;
393             unsigned max_x = 0, max_y = 0;
394             unsigned i = 0;
395             for(; i < ctx.keycount; i++){
396                 ckb_getline(cmd, param, value);
397                 unsigned x, y;
398                 if(strcmp(cmd, "key") || !*param || !sscanf(value, "%u,%u", &x, &y)){
399                     i--;
400                     continue;
401                 }
402                 ckb_key* key = ctx.keys + i;
403                 strncpy(key->name, param, CKB_KEYNAME_MAX);
404                 key->x = x;
405                 key->y = y;
406                 if(x > max_x)
407                     max_x = x;
408                 if(y > max_y)
409                     max_y = y;
410             }
411             ctx.width = max_x + 1;
412             ctx.height = max_y + 1;
413             // Skip anything else until "end keymap"
414             do {
415                 ckb_getline(cmd, param, value);
416                 if(!*cmd){
417                     printf("Error [ckb-main]: Reached EOF looking for \"end keymap\"");
418                     return -2;
419                 }
420             } while(strcmp(cmd, "end") || strcmp(param, "keymap"));
421             // Run init function
422             ckb_init(&ctx);
423             // Skip anything else until "begin params"
424             do {
425                 ckb_getline(cmd, param, value);
426                 if(!*cmd){
427                     printf("Error [ckb-main]: Reached EOF looking for \"begin params\"");
428                     return -2;
429                 }
430             } while(strcmp(cmd, "begin") || strcmp(param, "params"));
431             // Parse parameters
432             ckb_read_params(&ctx);
433             // Skip anything else until "begin run"
434             do {
435                 ckb_getline(cmd, param, value);
436                 if(!*cmd){
437                     printf("Error [ckb-main]: Reached EOF looking for \"begin run\"");
438                     return -2;
439                 }
440             } while(strcmp(cmd, "begin") || strcmp(param, "run"));
441             // Run the main loop
442             printf("begin run\n");
443             fflush(stdout);
444             while(1){
445                 ckb_getline(cmd, param, value);
446                 if(!*cmd || (!strcmp(cmd, "end") && !strcmp(param, "run")))
447                     break;
448                 // Parse input
449                 if(!strcmp(cmd, "start"))
450                     ckb_start(&ctx, 1);
451                 else if(!strcmp(cmd, "stop"))
452                     ckb_start(&ctx, 0);
453                 else if(!strcmp(cmd, "begin") && !strcmp(param, "params"))
454                     ckb_read_params(&ctx);
455                 else if(!strcmp(cmd, "key")){
```

```
456                     int x, y;
457                     if(sscanf(param, "%d,%d", &x, &y) == 2){
458                         // Find a key with this position
459                         ckb_key* key = 0;
460                         for(i = 0; i < keycount; i++){
461                             if(ctx.keys[i].x == x && ctx.keys[i].y == y){
462                                 key = ctx.keys + i;
463                                 break;
464                             }
465                         }
466                         if(key)
467                             ckb_keypress(&ctx, key, key->x, key->y, !strcmp(value, "down"));
468                         else
469                             ckb_keypress(&ctx, 0, x, y, !strcmp(value, "down"));
470                     } else {
471                         // Find a key with this name
472                         ckb_key* key = 0;
473                         for(i = 0; i < keycount; i++){
474                             if(!strcmp(ctx.keys[i].name, param)){
475                                 key = ctx.keys + i;
476                                 break;
477                             }
478                         }
479                         if(key)
480                             ckb_keypress(&ctx, key, key->x, key->y, !strcmp(value, "down"));
481                     }
482                 } else if(!strcmp(cmd, "frame")){
483                     int end = ckb_frame(&ctx);
484                     // Output the frame
485                     printf("begin frame\n");
486                     for(i = 0; i < ctx.keycount; i++){
487                         ckb_key* key = ctx.keys + i;
488                         printf("argb %s %02hhx%02hhx%02hhx%02hhx\n", key->name, key->
    a, key->r, key->g, key->b);
489                     }
490                     printf("end frame\n");
491                     if(end)
492                         break;
493                     fflush(stdout);
494                 } else {
495                     double delta = 0.;
496                     if(!strcmp(cmd, "time") && sscanf(param, "%lf", &delta) == 1)
497                         ckb_time(&ctx, delta);
498                 }
499             }
500             printf("end run\n");
501             fflush(stdout);
502             free(ctx.keys);
503             return 0;
504         }
505     }
506     printf("This program must be run from within ckb\n");
507     return -1;
508 }
```

Here is the call graph for this function:



13.65.3.14 void printurl (const char * *src*)

Definition at line 178 of file ckb-anim.h.

```
178                                {
179     char out[strlen(src) * 3 + 1];
180     char* dst = out;
181     char s;
182     while((s = *src++)){
183         if(s <= ',' || s == '/' ||
184                 (s >= ':' && s <= '@') ||
185                 s == '[' || s == ']' ||
186                 s >= 0x7F){
187             char a = s >> 4, b = s & 0xF;
188             if(a >= 10)
189                 a += 'A' - 10;
190             else
191                 a += '0';
192             if(b >= 10)
193                 b += 'A' - 10;
194             else
195                 b += '0';
196             dst[0] = '%';
197             dst[1] = a;
198             dst[2] = b;
199             dst += 3;
200         } else
201             *dst++ = s;
202     }
203     *dst = '\0';
204     printf("%s", out);
205 }
```

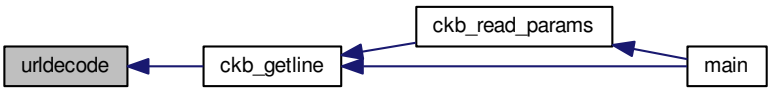
13.65.3.15 void urldecode (char * *dst*, const char * *src*)

Definition at line 208 of file ckb-anim.h.

Referenced by ckb_getline().

```
208                                            {
209     char a, b;
210     char s;
211     while((s = *src)){
212         if((s == '%') &&
213                 ((a = src[1]) && (b = src[2])) &&
214                 (isxdigit(a) && isxdigit(b))){
215             if(a >= 'a')
216                 a -= 'a'-'A';
217             if(a >= 'A')
218                 a -= 'A' - 10;
219             else
220                 a -= '0';
221             if(b >= 'a')
222                 b -= 'a'-'A';
223             if(b >= 'A')
224                 b -= 'A' - 10;
225             else
226                 b -= '0';
227             *dst++ = 16*a + b;
228             src += 3;
229         } else {
230             *dst++ = s;
231             src++;
232         }
233     }
234     *dst = '\0';
235 }
```

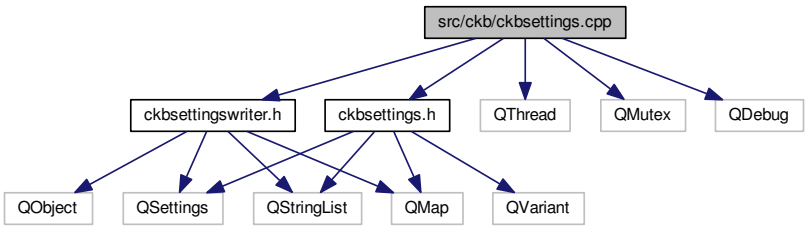
Here is the caller graph for this function:



13.66 src/ckb/ckbsettings.cpp File Reference

```
#include "ckbsettings.h"
#include "ckbsettingswriter.h"
#include <QThread>
#include <QMutex>
#include <QDebug>
```

Include dependency graph for ckbsettings.cpp:



Macros

- #define lockMutex QMutexLocker locker(backing == _globalSettings ? &settingsMutex : 0)
- #define lockMutexStatic QMutexLocker locker(&settingsMutex)
- #define lockMutexStatic2 QMutexLocker locker2(&settingsMutex)
- #define lockMutexCache QMutexLocker locker(&settingsCacheMutex)

Functions

- QAtomicInt cacheWritesInProgress (0)
- QMutex settingsMutex (QMutex::Recursive)
- QMutex settingsCacheMutex (QMutex::Recursive)
- static QSettings * globalSettings ()

Variables

- static QSettings * _globalSettings = 0
- static QThread * globalThread = 0
- static QMap< QString, QVariant > globalCache

13.66.1 Macro Definition Documentation

13.66.1.1 #define lockMutex QMutexLocker locker(backing == _globalSettings ? &settingsMutex : 0)

Definition at line 15 of file ckbsettings.cpp.

Referenced by CkbSettings::childGroups(), CkbSettings::childKeys(), CkbSettings::contains(), and CkbSettings::value().

13.66.1.2 #define lockMutexCache QMutexLocker locker(&settingsCacheMutex)

Definition at line 18 of file ckbsettings.cpp.

Referenced by CkbSettings::get(), CkbSettings::set(), and CkbSettings::setValue().

13.66.1.3 #define lockMutexStatic QMutexLocker locker(&settingsMutex)

Definition at line 16 of file ckbsettings.cpp.

Referenced by globalSettings(), and CkbSettings::set().

13.66.1.4 #define lockMutexStatic2 QMutexLocker locker2(&settingsMutex)

Definition at line 17 of file ckbsettings.cpp.

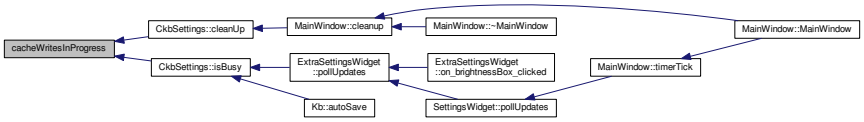
Referenced by CkbSettings::get().

13.66.2 Function Documentation

13.66.2.1 QAtomicInt cacheWritesInProgress (0)

Referenced by CkbSettings::cleanUp(), and CkbSettings::isBusy().

Here is the caller graph for this function:



13.66.2.2 static QSettings* globalSettings ( ) [static]

Definition at line 27 of file ckbsettings.cpp.

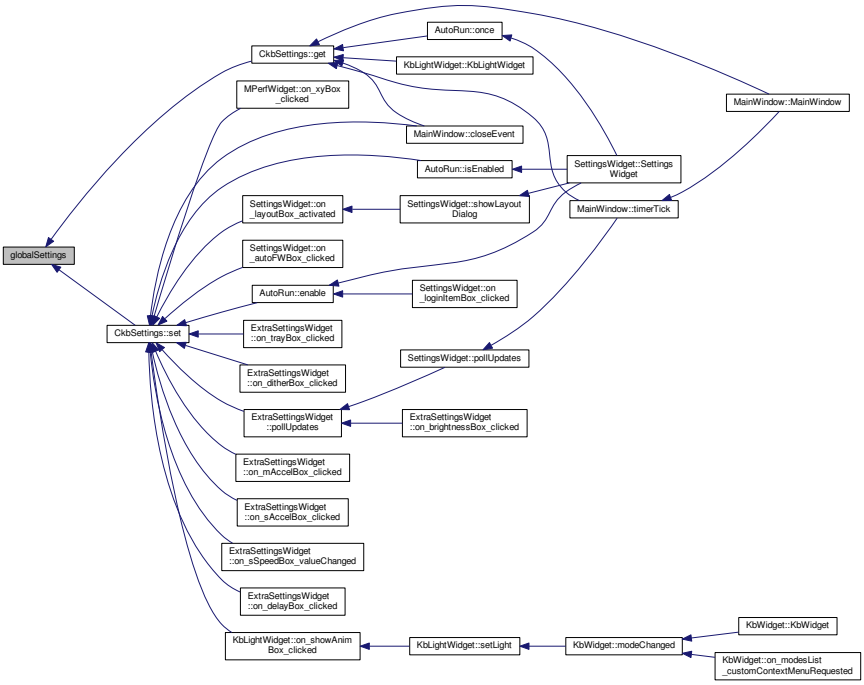
References _globalSettings, and lockMutexStatic.

Referenced by CkbSettings::get(), and CkbSettings::set().

```
27                                  {
28     if(!_globalSettings){
29         lockMutexStatic;
30         if(!(volatile QSettings*)_globalSettings){   // Check again after locking mutex in
       case another thread created the object
31             // Put the settings object in a separate thread so that it won't lock up the GUI when it syncs
32             globalThread = new QThread;
33             globalThread->start();
```

```
34             _globalSettings = new QSettings;
35             qInfo() << "Path  to settings is" << _globalSettings->fileName();
36             _globalSettings->moveToThread(globalThread);
37         }
38     }
39     return _globalSettings;
40 }
```

Here is the caller graph for this function:



13.66.2.3 QMutex settingsCacheMutex (QMutex::Recursive)

13.66.2.4 QMutex settingsMutex (QMutex::Recursive)

13.66.3 Variable Documentation

13.66.3.1 QSettings* _globalSettings = 0 [static]

Definition at line 8 of file ckbsettings.cpp.

Referenced by CkbSettings::cleanUp(), and globalSettings().

13.66.3.2 QMap<QString, QVariant> globalCache [static]

Definition at line 12 of file ckbsettings.cpp.

Referenced by CkbSettings::get(), CkbSettings::set(), and CkbSettings::setValue().

13.66.3.3 QThread* globalThread = 0 [static]

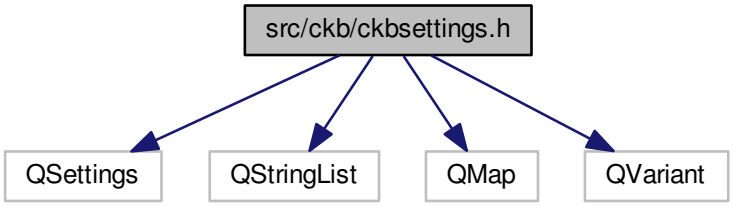
Definition at line 9 of file ckbsettings.cpp.

Referenced by CkbSettings::cleanUp().

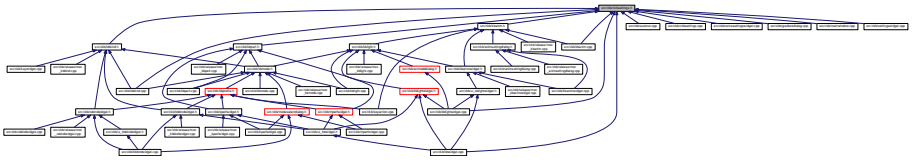
13.67 src/ckb/ckbsettings.h File Reference

```
#include <QSettings>
#include <QStringList>
#include <QMap>
#include <QVariant>
```

Include dependency graph for ckbsettings.h:



This graph shows which files directly or indirectly include this file:



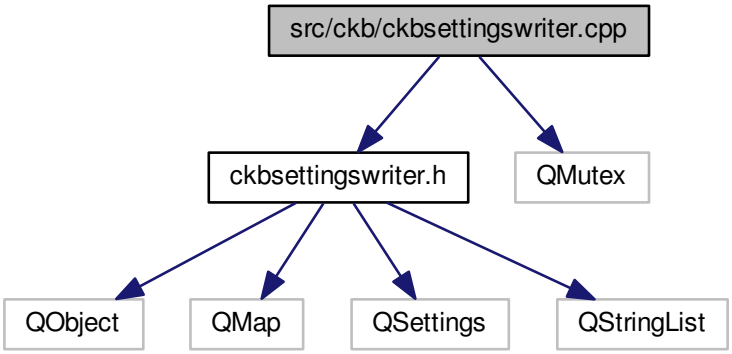
Data Structures

- class CkbSettings
- class SGroup

13.68 src/ckb/ckbsettingswriter.cpp File Reference

```
#include "ckbsettingswriter.h"
#include <QMutex>
```

Include dependency graph for ckbsettingswriter.cpp:



Macros

- #define lockMutex QMutexLocker locker(backing == _globalSettings ? &mutex : 0)
- #define lockMutexStatic QMutexLocker locker(&settingsMutex)
- #define lockMutexStatic2 QMutexLocker locker2(&settingsMutex)
- #define lockMutexCache QMutexLocker locker(&settingsCacheMutex)

Variables

- QAtomicInt cacheWritesInProgress
- QMutex settingsMutex
- QMutex settingsCacheMutex

13.68.1 Macro Definition Documentation

13.68.1.1 #define lockMutex QMutexLocker locker(backing == _globalSettings ? &mutex : 0)

Definition at line 7 of file ckbsettingswriter.cpp.

13.68.1.2 #define lockMutexCache QMutexLocker locker(&settingsCacheMutex)

Definition at line 10 of file ckbsettingswriter.cpp.

13.68.1.3 #define lockMutexStatic QMutexLocker locker(&settingsMutex)

Definition at line 8 of file ckbsettingswriter.cpp.

Referenced by CkbSettingsWriter::run().

13.68.1.4 #define lockMutexStatic2 QMutexLocker locker2(&settingsMutex)

Definition at line 9 of file ckbsettingswriter.cpp.

13.68.2 Variable Documentation

13.68.2.1 QAtomicInt cacheWritesInProgress

Referenced by CkbSettingsWriter::CkbSettingsWriter(), and CkbSettingsWriter::~CkbSettingsWriter().

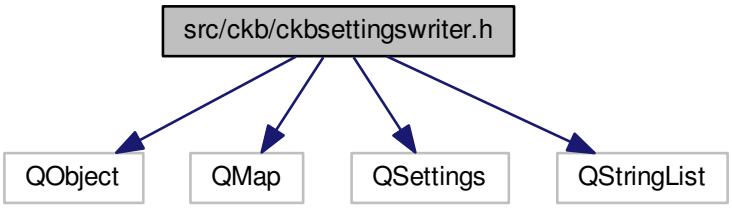
13.68.2.2 QMutex settingsCacheMutex

13.68.2.3 QMutex settingsMutex

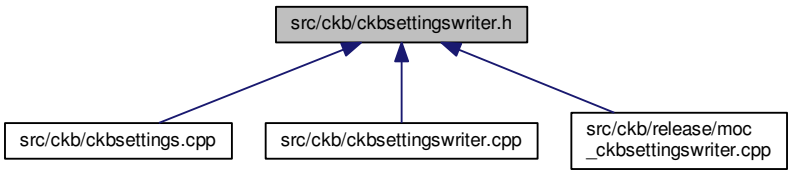
13.69 src/ckb/ckbsettingswriter.h File Reference

```
#include <QObject>
#include <QMap>
#include <QSettings>
#include <QStringList>
```

Include dependency graph for ckbsettingswriter.h:



This graph shows which files directly or indirectly include this file:



Data Structures

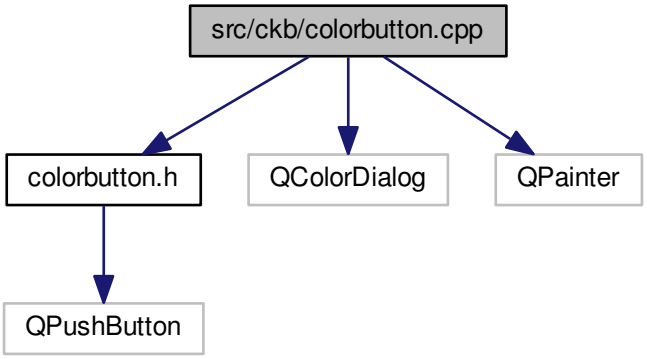
- class CkbSettingsWriter

13.70 src/ckb/colorbutton.cpp File Reference

```
#include "colorbutton.h"
```

```
#include <QColorDialog>
#include <QPainter>
```

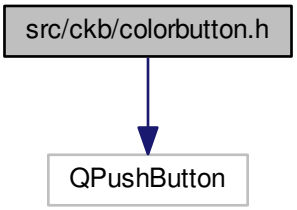
Include dependency graph for colorbutton.cpp:



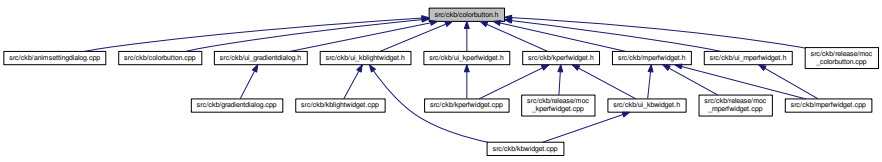
13.71 src/ckb/colorbutton.h File Reference

```
#include <QPushButton>
```

Include dependency graph for colorbutton.h:



This graph shows which files directly or indirectly include this file:



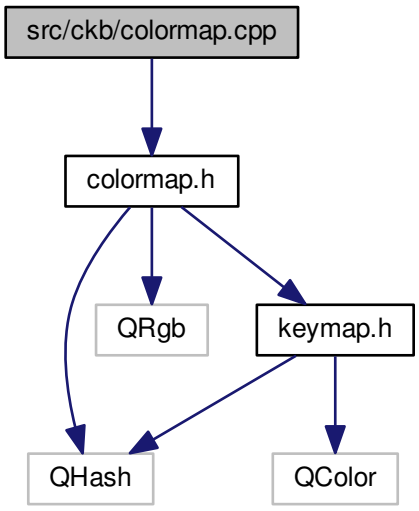
Data Structures

- class ColorButton

13.72 src/ckb/colormap.cpp File Reference

```
#include "colormap.h"
```

Include dependency graph for colormap.cpp:



Functions

- static int qs_strcmp (const void *lhs, const void *rhs)

13.72.1 Function Documentation

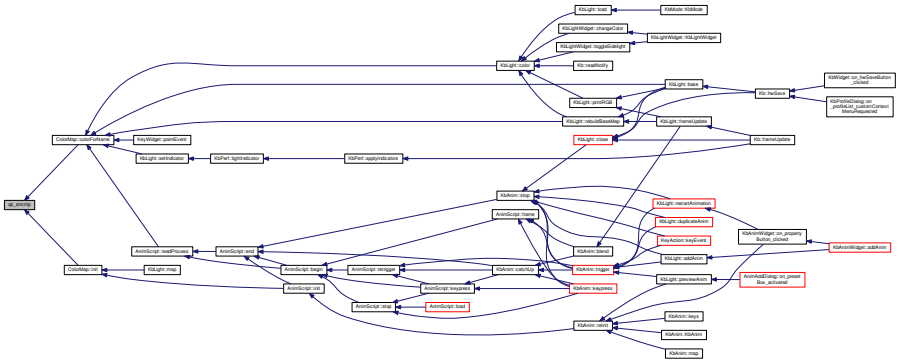
13.72.1.1 static int qs_strcmp (const void * *lhs*, const void * *rhs*) [static]

Definition at line 65 of file colormap.cpp.

Referenced by ColorMap::colorForName(), and ColorMap::init().

```
65                                                  {
66     return strcmp(*(const char**)lhs, *(const char**)rhs);
67 }
```

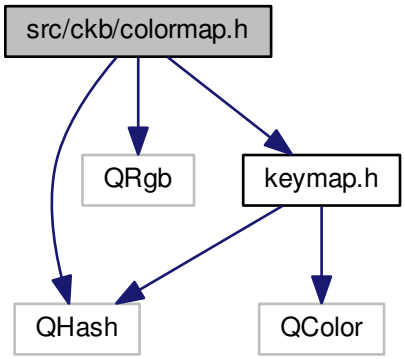
Here is the caller graph for this function:



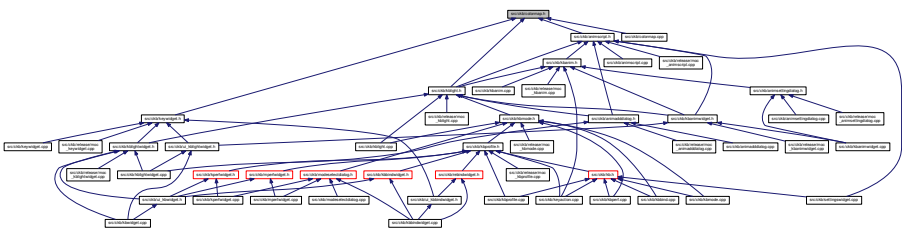
13.73 src/ckb/colormap.h File Reference

```
#include <QHash>
#include <QRgb>
#include "keymap.h"
```

Include dependency graph for colormap.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class ColorMap

Typedefs

- typedef QHash< QString, QRgb > QColorMap
- typedef QHashIterator< QString, QRgb > QColorMapIterator
- typedef QMutableHashIterator < QString, QRgb > QMutableColorMapIterator

13.73.1 Typedef Documentation

13.73.1.1 typedef QHash<QString, QRgb> QColorMap

Definition at line 9 of file colormap.h.

13.73.1.2 typedef QHashIterator<QString, QRgb> QColorMapIterator

Definition at line 10 of file colormap.h.

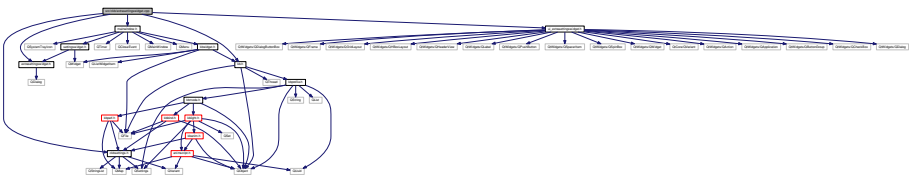
13.73.1.3 typedef QMutableHashIterator<QString, QRgb> QMutableColorMapIterator

Definition at line 11 of file colormap.h.

13.74 src/ckb/extrasettingswidget.cpp File Reference

```
#include "extrasettingswidget.h"
#include "ui_extrasettingswidget.h"
#include "kb.h"
#include "mainwindow.h"
#include "ckbsettings.h"
```

Include dependency graph for extrasettingswidget.cpp:



Variables

- static int lastSharedDimming = -2

13.74.1 Variable Documentation

13.74.1.1 int lastSharedDimming = -2 [static]

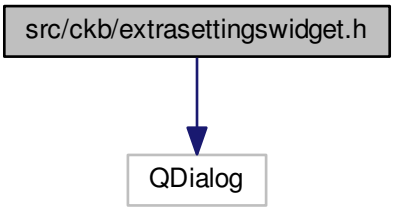
Definition at line 8 of file extrasettingswidget.cpp.

Referenced by ExtraSettingsWidget::ExtraSettingsWidget(), and ExtraSettingsWidget::pollUpdates().

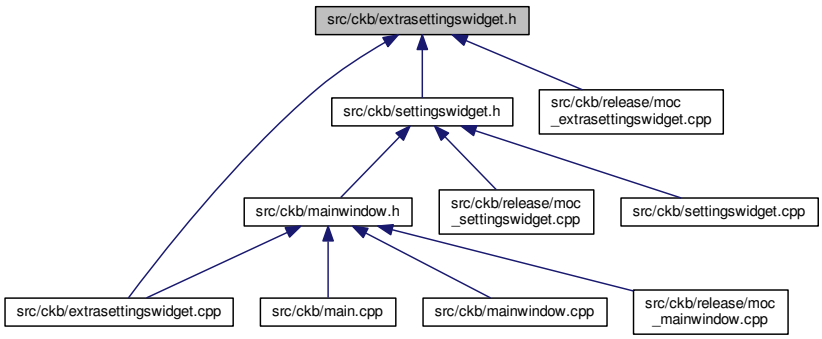
13.75 src/ckb/extrasettingswidget.h File Reference

```
#include <QDialog>
```

Include dependency graph for extrasettingswidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class ExtraSettingsWidget

Namespaces

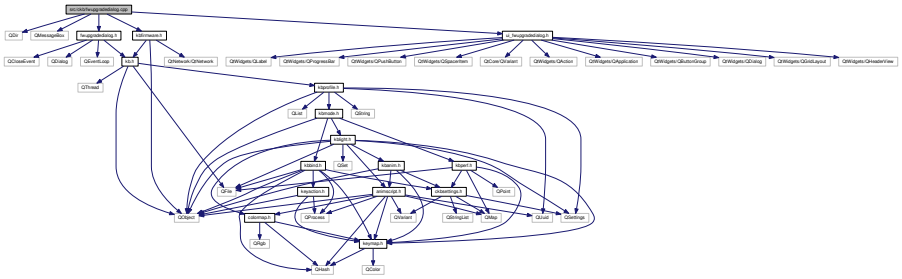
- Ui

13.76 src/ckb/fwupgradedialog.cpp File Reference

```
#include <QDir>
```

```
#include <QMessageBox>
#include "fwupgradedialog.h"
#include "kbfirmware.h"
#include "ui_fwupgradedialog.h"
```

Include dependency graph for fwupgradedialog.cpp:



Data Structures

- struct Kbld

Functions

- static const int DIALOG_HEIGHT_MAX (240)
- static float verifyFw (const QByteArray &blob, const QString &features)

Variables

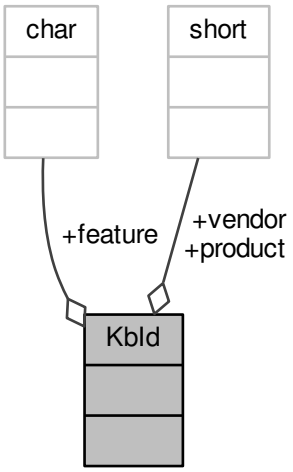
- static Kbld ids []
- static const int DIALOG_WIDTH = 420
- static const int DIALOG_HEIGHT_MIN = 200

13.76.1 Data Structure Documentation

13.76.1.1 struct Kbld

Definition at line 8 of file fwupgradedialog.cpp.

Collaboration diagram for KbId:



Data Fields

const char *	feature	
short	product	
short	vendor	

13.76.2 Function Documentation

13.76.2.1 static const int DIALOG_HEIGHT_MAX (240) [static]

Referenced by FwUpgradeDialog::exec().

Here is the caller graph for this function:

13.76.2.2 static float verifyFw (const QByteArray & *blob*, const QString & *features*) [static]

Definition at line 86 of file fwupgradedialog.cpp.

References ids.

Referenced by FwUpgradeDialog::exec().

```
86                                                                          {
87     if(blob.length() < 0x0108)
88         return 0.f;
89     const char* bData = blob.data();
90     // Make sure it matches this device based on the vendor and product IDs embedded in the blob
91     bool match = false;
```

```
92     for(uint i = 0; i < sizeof(ids)/sizeof(KbId); i++){
93         if(!memcmp(&ids[i].vendor, bData + 0x102, 2) && !memcmp(&ids[i].product, bData + 0x104, 2)
94                 && features.startsWith(ids[i].feature, Qt::CaseInsensitive)){
95             match = true;
96             break;
97         }
98     }
99     if(!match)
100         return 0.f;
101     // Copy the version from the blob
102     short version;
103     memcpy(&version, bData + 0x106, 2);
104     // Un-hexify it
105     return QString::number(version, 16).toFloat() / 100.f;
106 }
```

Here is the caller graph for this function:

13.76.3 Variable Documentation

13.76.3.1 const int DIALOG_HEIGHT_MIN = 200 [static]

Definition at line 30 of file fwupgradedialog.cpp.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), and FwUpgradeDialog::on_actionButton_clicked().

13.76.3.2 const int DIALOG_WIDTH = 420 [static]

Definition at line 29 of file fwupgradedialog.cpp.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::FwUpgradeDialog(), and FwUpgradeDialog::on_actionButton_clicked().

13.76.3.3 KbId ids[] [static]

Initial value:

```
= {

    { 0x1b1c, 0x1b17, "corsair k65 rgb" },
    { 0x1b1c, 0x1b3f, "corsair k68" },
    { 0x1b1c, 0x1b13, "corsair k70 rgb" },
    { 0x1b1c, 0x1b11, "corsair k95 rgb" },
    { 0x1b1c, 0x1b15, "corsair strafe monochrome" },
    { 0x1b1c, 0x1b20, "corsair strafe rgb" },

    { 0x1b1c, 0x1b12, "corsair m65 rgb" },
    { 0x1b1c, 0x1b1e, "corsair scimitar rgb" },

    { 0x1b1c, 0x1b32, "corsair sabre optical rgb" }
}
```

Definition at line 14 of file fwupgradedialog.cpp.

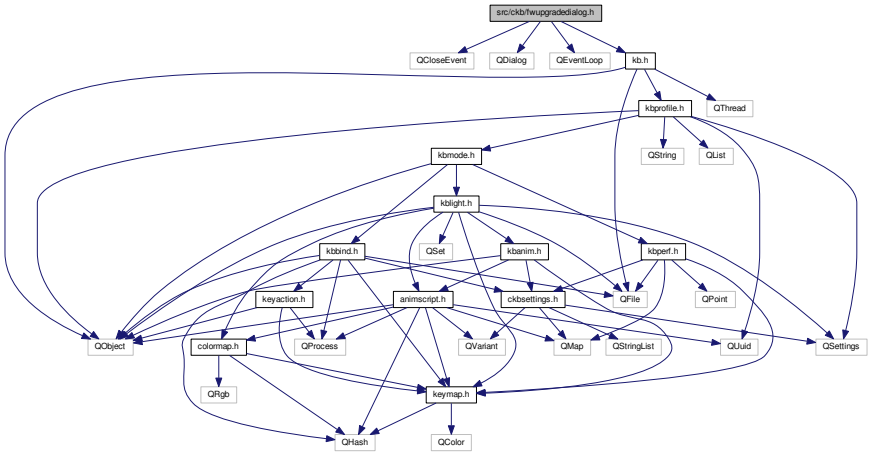
Referenced by verifyFw().

13.77 src/ckb/fwupgradedialog.h File Reference

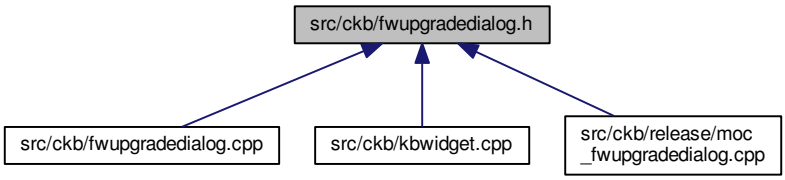
```
#include <QCloseEvent>
```

```
#include <QDialog>
#include <QEventLoop>
#include "kb.h"
```

Include dependency graph for fwupgradedialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class FwUpgradeDialog

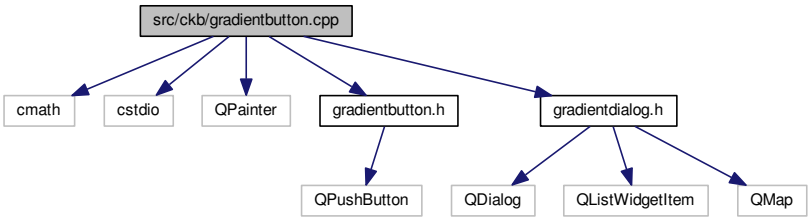
Namespaces

- Ui

13.78 src/ckb/gradientbutton.cpp File Reference

```
#include <cmath>
#include <cstdio>
#include <QPainter>
#include "gradientbutton.h"
#include "gradientdialog.h"
```

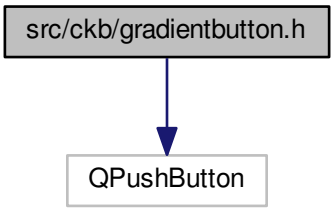
Include dependency graph for gradientbutton.cpp:



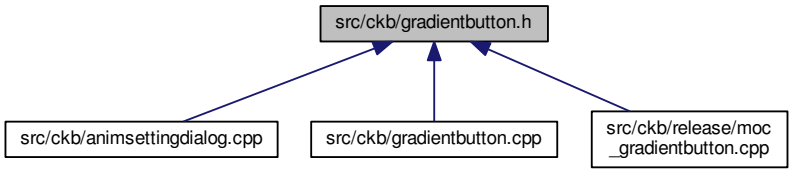
13.79 src/ckb/gradientbutton.h File Reference

```
#include <QPushButton>
```

Include dependency graph for gradientbutton.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class GradientButton

13.80 src/ckb/gradientdialog.cpp File Reference

```
#include <cmath>
#include <QMessageBox>
#include <QPainter>
#include <QSettings>
#include "ckbsettings.h"
#include "gradientdialog.h"
#include "ui_gradientdialog.h"
```

Include dependency graph for gradientdialog.cpp:

Variables

- static const QString prefsPath = "Stored Gradients"

13.80.1 Variable Documentation

13.80.1.1 const QString prefsPath = "Stored Gradients" [static]

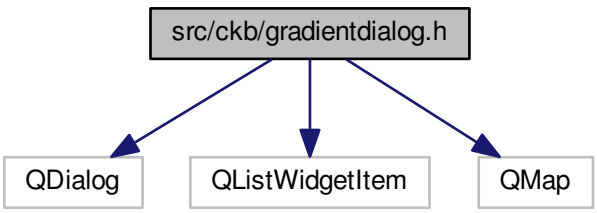
Definition at line 9 of file gradientdialog.cpp.

Referenced by GradientDialog::GradientDialog(), and GradientDialog::~GradientDialog().

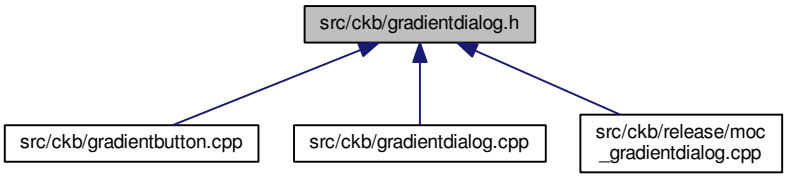
13.81 src/ckb/gradientdialog.h File Reference

```
#include <QDialog>
#include <QListWidgetItem>
#include <QMap>
```

Include dependency graph for gradientdialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class GradientDialog
- struct GradientDialog::Preset

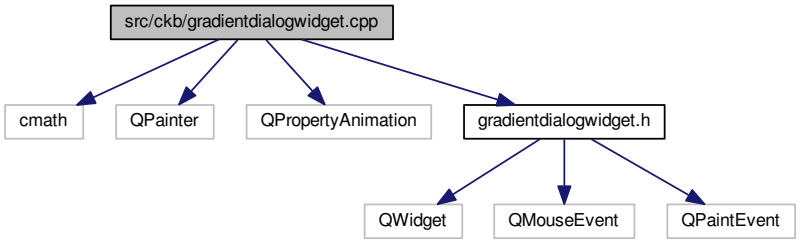
Namespaces

- Ui

13.82 src/ckb/gradientdialogwidget.cpp File Reference

```
#include <cmath>
#include <QPainter>
#include <QPropertyAnimation>
#include "gradientdialogwidget.h"
```

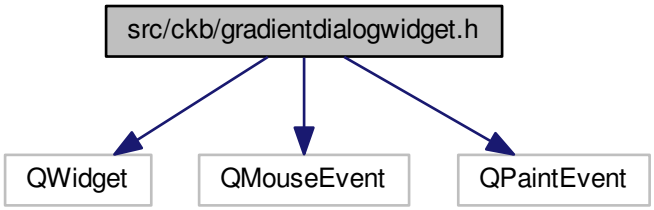
Include dependency graph for gradientdialogwidget.cpp:



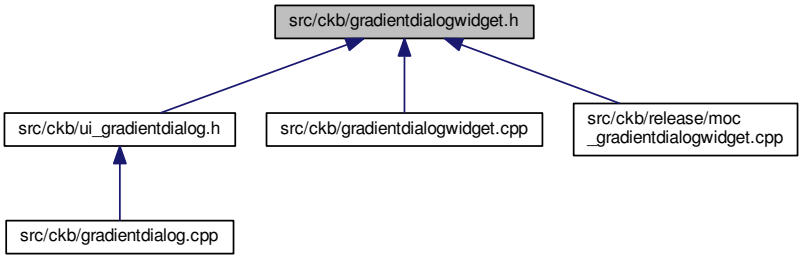
13.83 src/ckb/gradientdialogwidget.h File Reference

```
#include <QWidget>
#include <QMouseEvent>
#include <QPaintEvent>
```

Include dependency graph for gradientdialogwidget.h:



This graph shows which files directly or indirectly include this file:



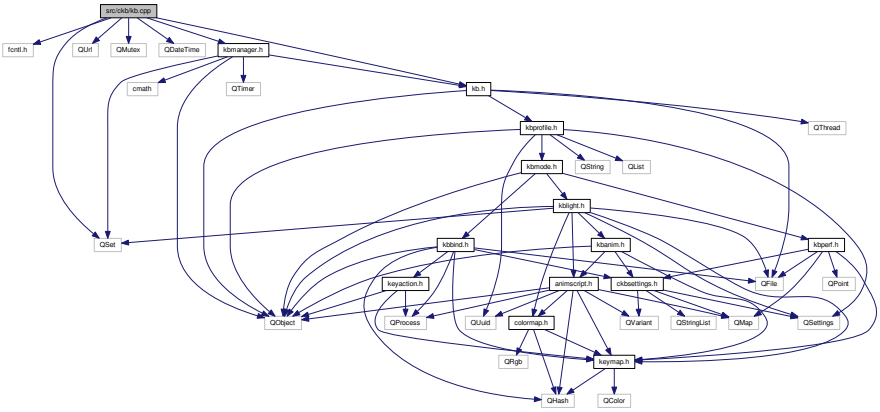
Data Structures

- class GradientDialogWidget

13.84 src/ckb/kb.cpp File Reference

```
#include <fcntl.h>
#include <QSet>
#include <QUrl>
#include <QMutex>
#include <QDateTime>
#include "kb.h"
#include "kbmanager.h"
```

Include dependency graph for kb.cpp:



Variables

- static QSet< Kb * > activeDevices
- static QSet< QString > notifyPaths
- static QMutex notifyPathMutex

13.84.1 Variable Documentation

13.84.1.1 QSet<Kb*> activeDevices [static]

Definition at line 10 of file kb.cpp.

Referenced by Kb::dither(), Kb::frameRate(), Kb::Kb(), Kb::layout(), Kb::macroDelay(), Kb::mouseAccel(), Kb::scrollSpeed(), and Kb::~Kb().

13.84.1.2 QMutex notifyPathMutex [static]

Definition at line 13 of file kb.cpp.

Referenced by Kb::Kb(), and Kb::run().

13.84.1.3 QSet<QString> notifyPaths [static]

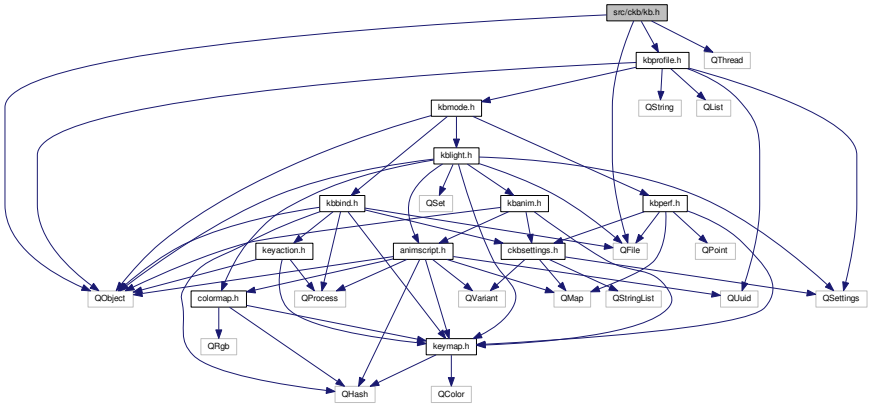
Definition at line 12 of file kb.cpp.

Referenced by Kb::Kb(), Kb::run(), and Kb::~Kb().

13.85 src/ckb/kb.h File Reference

```
#include <QObject>
#include <QFile>
#include <QThread>
#include "kbprofile.h"
```

Include dependency graph for kb.h:



This graph shows which files directly or indirectly include this file:

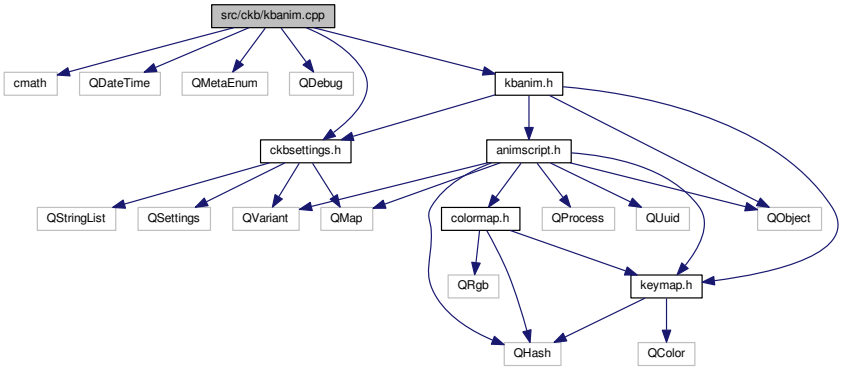
Data Structures

- class Kb

13.86 src/ckb/kbanim.cpp File Reference

```
#include <cmath>
#include <QDateTime>
#include <QMetaEnum>
#include <QDebug>
#include "ckbsettings.h"
#include "kbanim.h"
```

Include dependency graph for kbanim.cpp:



Typedefs

- typedef float(∗ blendFunc)(float, float)

Functions

- static float blendNormal (float bg, float fg)
- static float blendAdd (float bg, float fg)
- static float blendSubtract (float bg, float fg)
- static float blendMultiply (float bg, float fg)
- static float blendDivide (float bg, float fg)

Variables

- static blendFunc functions [5] = { blendNormal, blendAdd, blendSubtract, blendMultiply, blendDivide }

13.86.1 Typedef Documentation

13.86.1.1 typedef float(∗ blendFunc)(float, float)

Definition at line 362 of file kbanim.cpp.

13.86.2 Function Documentation

13.86.2.1 static float blendAdd (float *bg,* float *fg*) [static]

Definition at line 337 of file kbanim.cpp.

```
337                                             {
338     float res = bg + fg;
339     if(res > 1.f)
340         res = 1.f;
341     return res;
342 }
```

13.86.2.2 static float blendDivide (float *bg,* float *fg*) [static]

Definition at line 355 of file kbanim.cpp.

```
355                                               {
356     float res = bg / fg;
357     if(res > 1.f)
358         res = 1.f;
359     return res;
360 }
```

13.86.2.3 static float blendMultiply (float *bg,* float *fg*) [static]

Definition at line 351 of file kbanim.cpp.

```
351                                                 {
352     return bg * fg;
353 }
```

13.86.2.4 static float blendNormal (float *bg*, float *fg*) [static]

Definition at line 333 of file kbanim.cpp.

```
333                                          {
334     return fg;
335 }
```

13.86.2.5 static float blendSubtract (float *bg*, float *fg*) [static]

Definition at line 344 of file kbanim.cpp.

```
344                                             {
345     float res = bg - fg;
346     if(res < 0.f)
347         res = 0.f;
348     return res;
349 }
```

13.86.3 Variable Documentation

13.86.3.1 blendFunc functions[5] = { blendNormal, blendAdd, blendSubtract, blendMultiply, blendDivide } [static]

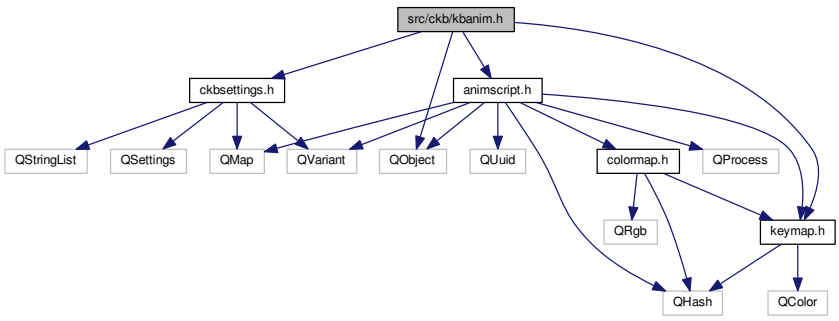
Definition at line 363 of file kbanim.cpp.

Referenced by KbAnim::blend().

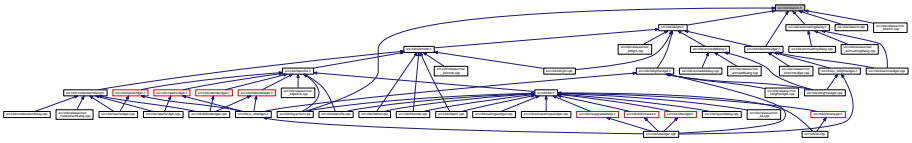
13.87 src/ckb/kbanim.h File Reference

```
#include <QObject>
#include "ckbsettings.h"
#include "animscript.h"
#include "keymap.h"
```

Include dependency graph for kbanim.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbAnim

13.88 src/ckb/kbanimwidget.cpp File Reference

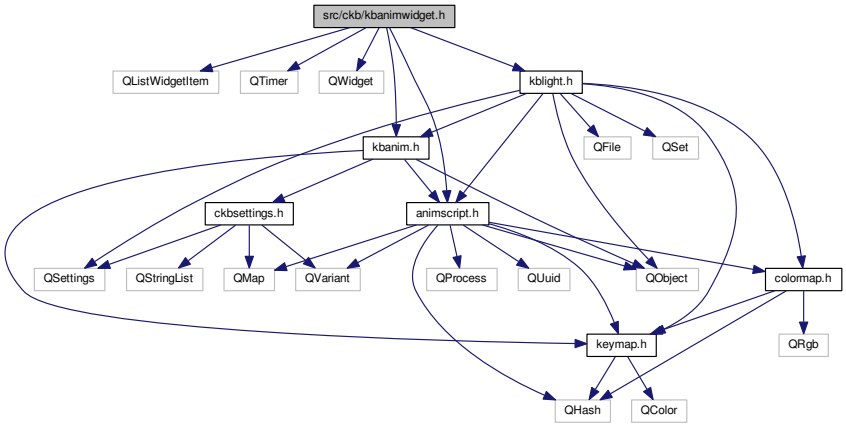
```
#include <QDateTime>
#include <QMenu>
#include "animsettingdialog.h"
#include "kbanimwidget.h"
#include "ui_kbanimwidget.h"
```

Include dependency graph for kbanimwidget.cpp:

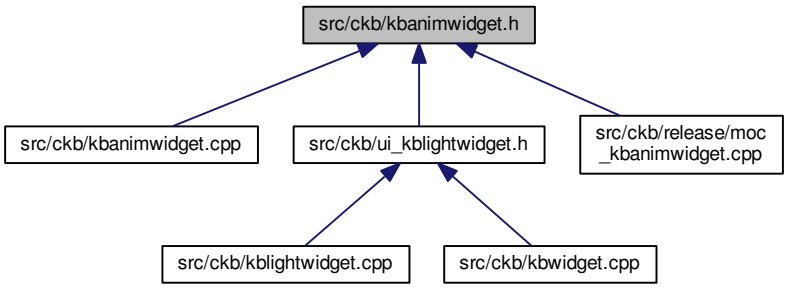
13.89 src/ckb/kbanimwidget.h File Reference

```
#include <QListWidgetItem>
#include <QTimer>
#include <QWidget>
#include "animscript.h"
#include "kbanim.h"
#include "kblight.h"
```

Include dependency graph for kbanimwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbAnimWidget

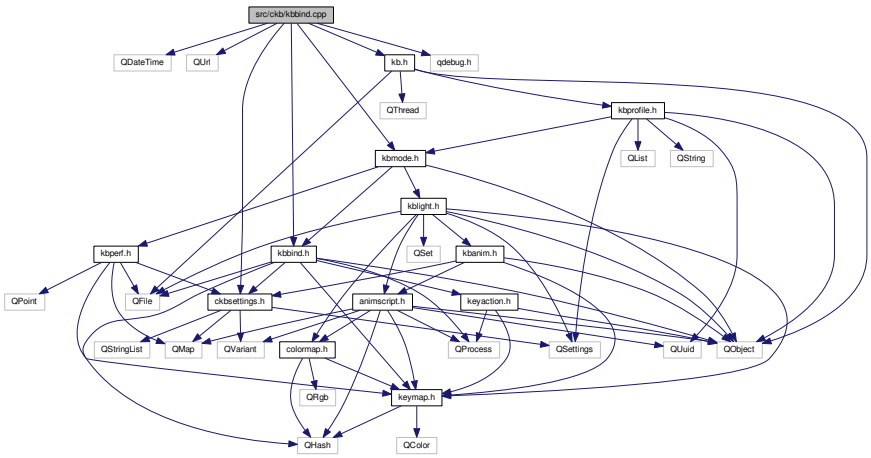
Namespaces

- Ui

13.90 src/ckb/kbbind.cpp File Reference

```
#include <QDateTime>
#include <QUrl>
#include "ckbsettings.h"
#include "kbbind.h"
#include "kbmode.h"
#include "kb.h"
#include "qdebug.h"
```

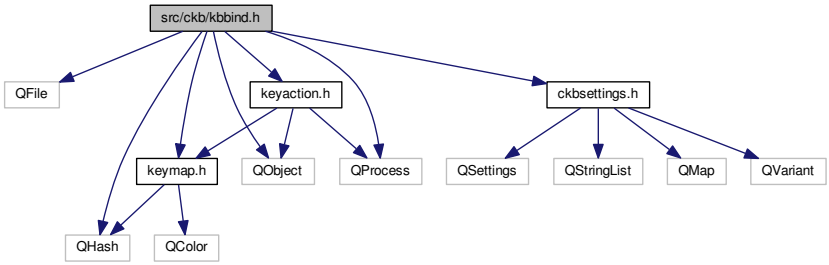
Include dependency graph for kbbind.cpp:



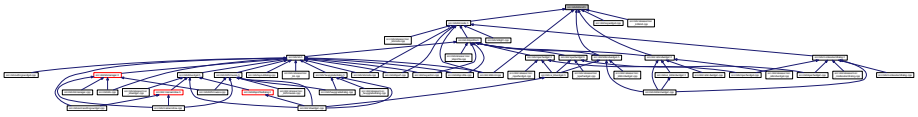
13.91 src/ckb/kbbind.h File Reference

```
#include <QFile>
#include <QHash>
#include <QObject>
#include <QProcess>
#include "ckbsettings.h"
#include "keymap.h"
#include "keyaction.h"
```

Include dependency graph for kbbind.h:



This graph shows which files directly or indirectly include this file:



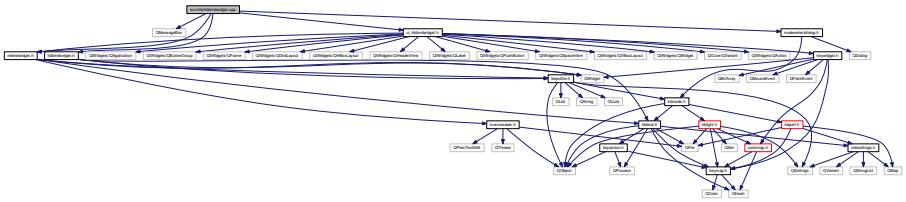
Data Structures

- class KbBind

13.92 src/ckb/kbbindwidget.cpp File Reference

```
#include <QMessageBox>
#include "kbbindwidget.h"
#include "modeselectdialog.h"
#include "rebindwidget.h"
#include "ui_kbbindwidget.h"
```

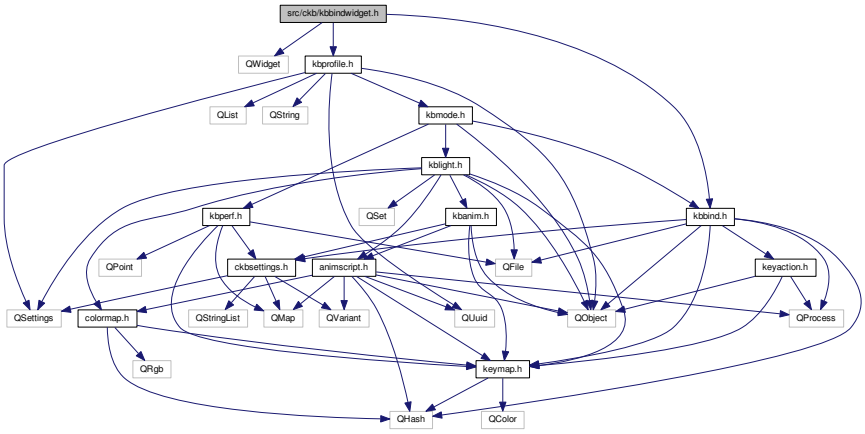
Include dependency graph for kbbindwidget.cpp:



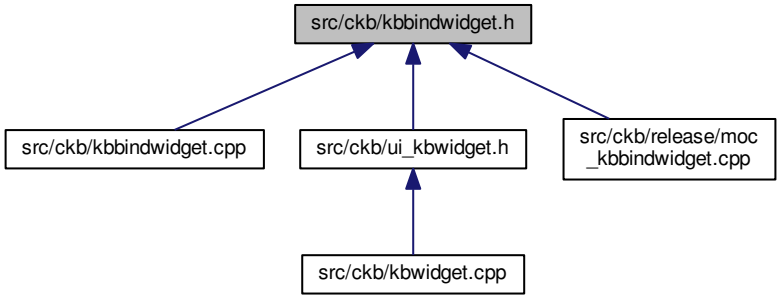
13.93 src/ckb/kbbindwidget.h File Reference

```
#include <QWidget>
#include "kbbind.h"
#include "kbprofile.h"
```

Include dependency graph for kbbindwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbBindWidget

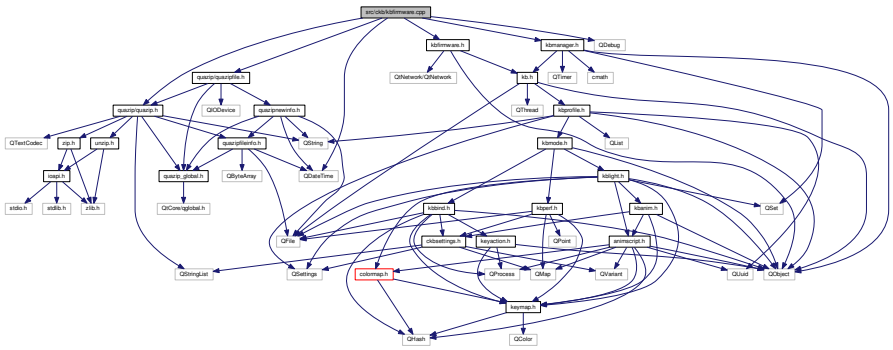
Namespaces

- Ui

13.94 src/ckb/kbfirmware.cpp File Reference

```
#include "kbfirmware.h"
#include "kbmanager.h"
#include "quazip/quazip.h"
#include "quazip/quazipfile.h"
#include <QDateTime>
#include <QDebug>
```

Include dependency graph for kbfirmware.cpp:



Functions

- static QString tableName (const QString &features)

Variables

- static const quint64 AUTO_CHECK_TIME = 60 ∗ 60 ∗ 1000

13.94.1 Function Documentation

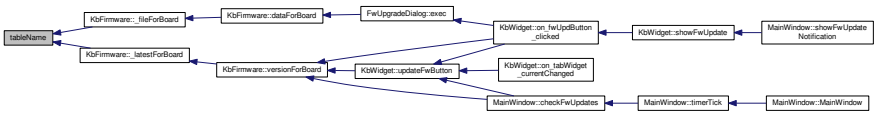
13.94.1.1 static QString tableName (const QString & *features*) [static]

Definition at line 150 of file kbfirmware.cpp.

Referenced by KbFirmware::_fileForBoard(), and KbFirmware::_latestForBoard().

```
150                                                 {
151     QStringList components = features.split(" ");
152     if(components.length() < 2)
153         return "";
154     // First two components are vendor and model
155     QString vendorModel = components[0].toUpper() + "-" + components[1].toUpper();
156     // Add "RGB" on RGB boards
157     if(features.contains("rgb") && !features.contains("monochrome"))
158         vendorModel += "RGB";
159     return vendorModel;
160 }
```

Here is the caller graph for this function:



13.94.2 Variable Documentation

13.94.2.1 const quint64 AUTO_CHECK_TIME = 60 ∗ 60 ∗ 1000 [static]

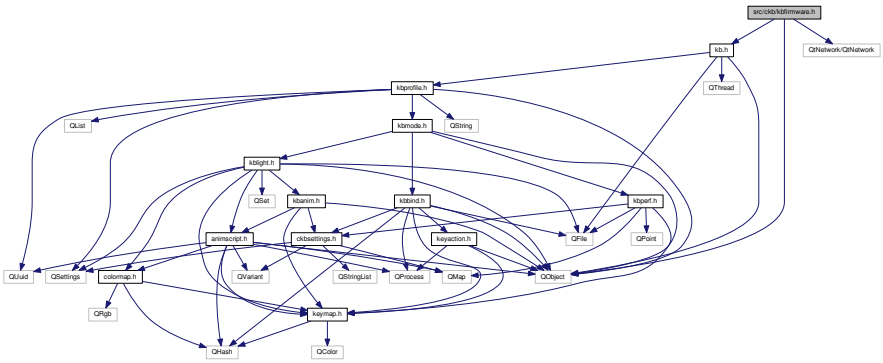
Definition at line 9 of file kbfirmware.cpp.

Referenced by KbFirmware::_checkUpdates().

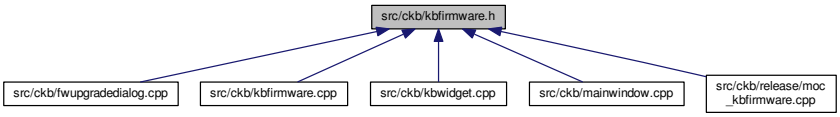
13.95 src/ckb/kbfirmware.h File Reference

```
#include "kb.h"
#include <QObject>
#include <QtNetwork/QtNetwork>
```

Include dependency graph for kbfirmware.h:



This graph shows which files directly or indirectly include this file:



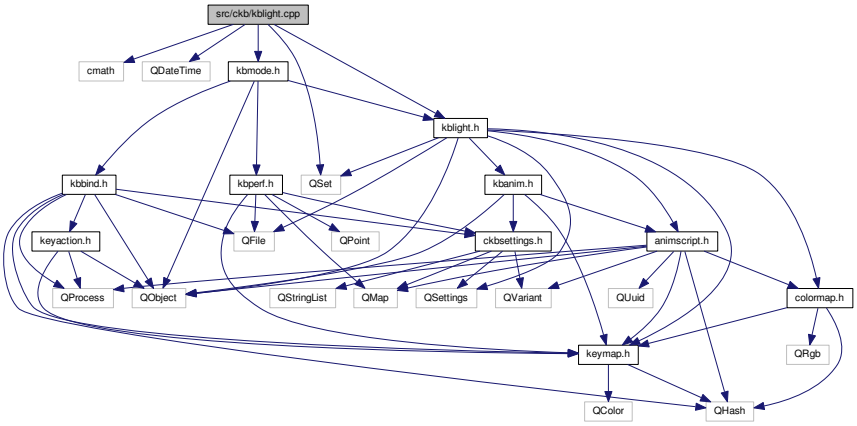
Data Structures

- class KbFirmware
- struct KbFirmware::FW

13.96 src/ckb/kblight.cpp File Reference

```
#include <cmath>
#include <QDateTime>
#include <QSet>
#include "kblight.h"
#include "kbmode.h"
```

Include dependency graph for kblight.cpp:



Functions

- static float sToL (float srgb)
- static float lToS (float linear)
- QRgb monoRgb (float r, float g, float b)

Variables

- static int _shareDimming = -1
- static QSet< KbLight * > activeLights

13.96.1 Function Documentation

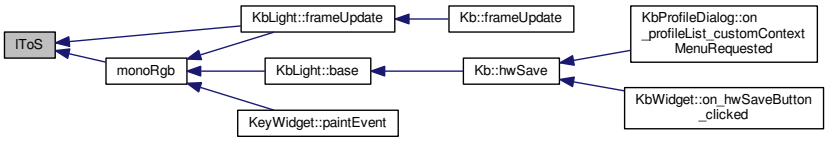
13.96.1.1 static float lToS (float *linear*) [static]

Definition at line 277 of file kblight.cpp.

Referenced by KbLight::frameUpdate(), and monoRgb().

```
277                                {
278     if(linear <= 0.0031308f)
279         return 12.92f * linear * 255.f;
280     return (1.055f * pow(linear, 1.f / 2.4f) - 0.055f) * 255.f;
281 }
```

Here is the caller graph for this function:



13.96.1.2 QRgb monoRgb (float *r*, float *g*, float *b*)

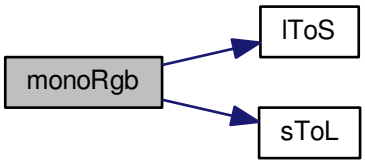
Definition at line 284 of file kblight.cpp.

References lToS(), and sToL().

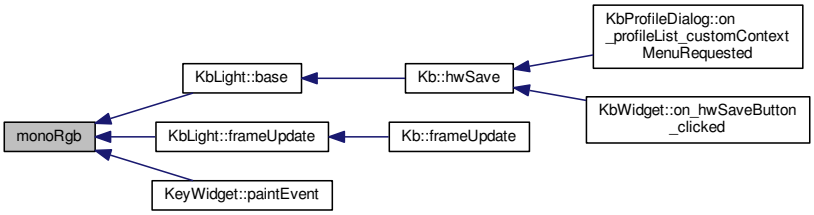
Referenced by KbLight::base(), KbLight::frameUpdate(), and KeyWidget::paintEvent().

```
284                                     {
285     // It's important to use a linear colorspace for this, otherwise the colors will appear inconsistent
286     // Note that although we could use linear space for alpha blending or the animation blending functions,
        we don't.
287     // The reason for this is that photo manipulation programs don't do it either, so even though the
        result would technically be more correct,
288     // it would look wrong to most people.
289     r = sToL(r);
290     g = sToL(g);
291     b = sToL(b);
292     int value = round(lToS((r + g + b) / 3.f));
293     return qRgb(value, value, value);
294 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



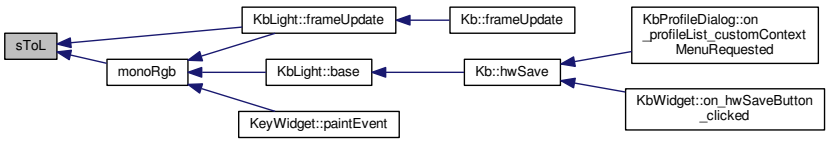
13.96.1.3 static float sToL (float *srgb*) [static]

Definition at line 270 of file kblight.cpp.

Referenced by KbLight::frameUpdate(), and monoRgb().

```
270                             {
271     srgb /= 255.f;
272     if(srgb <= 0.04045f)
273         return srgb / 12.92f;
274     return pow((srgb + 0.055f) / 1.055f, 2.4f);
275 }
```

Here is the caller graph for this function:



13.96.2 Variable Documentation

13.96.2.1 int _shareDimming = -1 [static]

Definition at line 7 of file kblight.cpp.

Referenced by KbLight::dimming(), KbLight::open(), and KbLight::shareDimming().

13.96.2.2 QSet<KbLight*> activeLights [static]

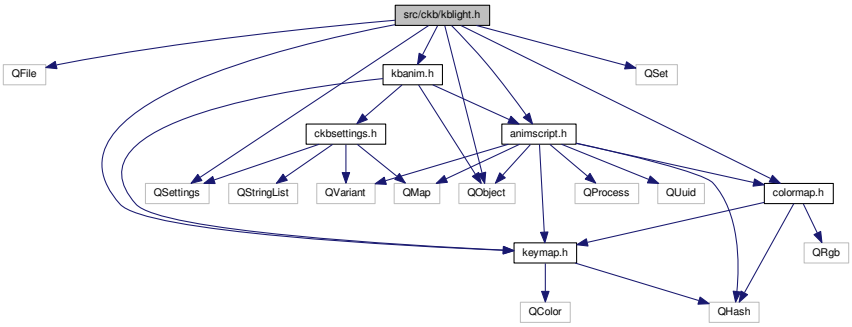
Definition at line 8 of file kblight.cpp.

Referenced by KbLight::close(), KbLight::open(), KbLight::shareDimming(), and KbLight::~KbLight().

13.97 src/ckb/kblight.h File Reference

```
#include <QFile>
#include <QObject>
#include <QSet>
#include <QSettings>
#include "animscript.h"
#include "kbanim.h"
#include "keymap.h"
#include "colormap.h"
```

Include dependency graph for kblight.h:



This graph shows which files directly or indirectly include this file:

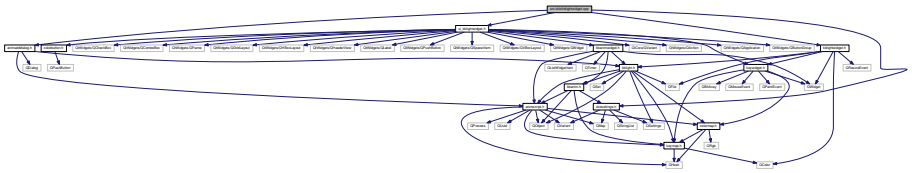
Data Structures

- class KbLight

13.98 src/ckb/kblightwidget.cpp File Reference

```
#include "animadddialog.h"
#include "ckbsettings.h"
#include "kblightwidget.h"
#include "ui_kblightwidget.h"
```

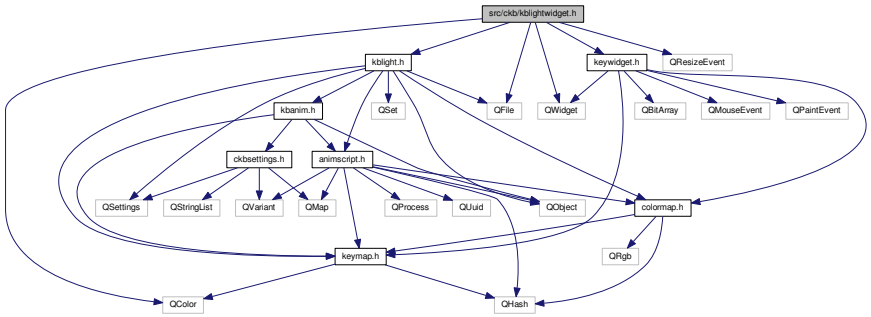
Include dependency graph for kblightwidget.cpp:



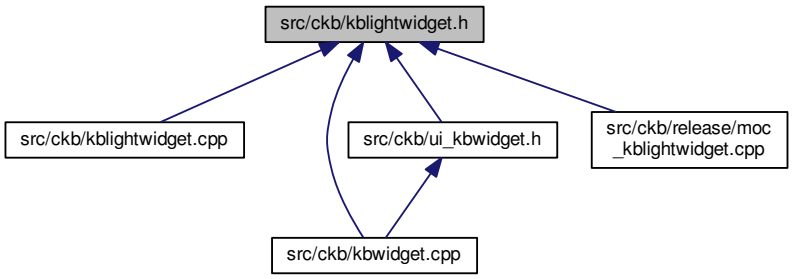
13.99 src/ckb/kblightwidget.h File Reference

```
#include <QWidget>
#include <QColor>
#include <QFile>
#include <QResizeEvent>
#include "kblight.h"
#include "keywidget.h"
```

Include dependency graph for kblightwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbLightWidget

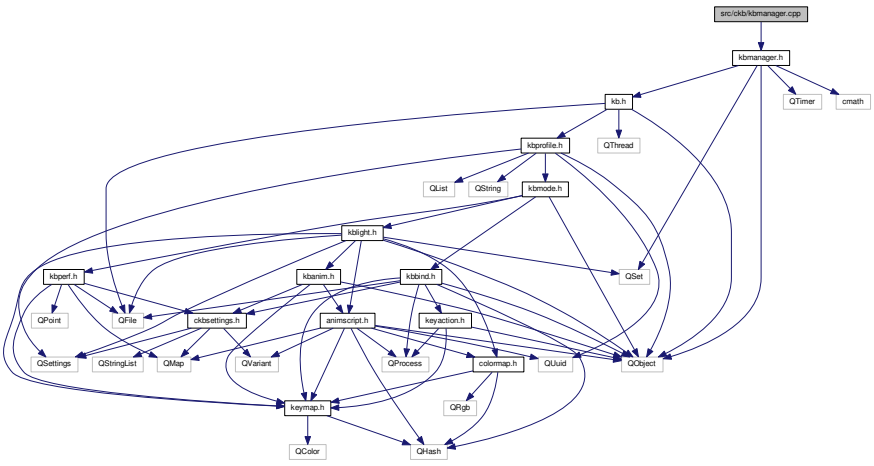
Namespaces

- Ui

13.100 src/ckb/kbmanager.cpp File Reference

```
#include "kbmanager.h"
```

Include dependency graph for kbmanager.cpp:



Variables

- QString devpath = "/dev/input/ckb%1"

13.100.1 Variable Documentation

13.100.1.1 QString devpath = "/dev/input/ckb%1"

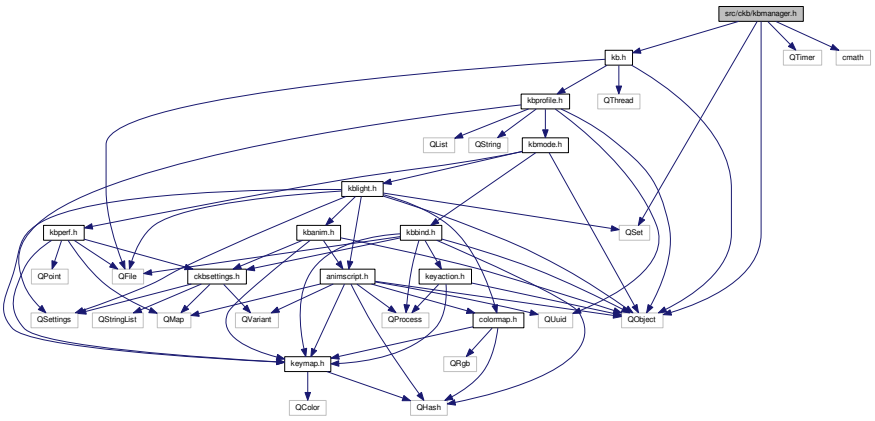
Definition at line 4 of file kbmanager.cpp.

Referenced by _setupusb(), closeusb(), main(), os_inputmain(), os_setupusb(), and KbManager::scanKeyboards().

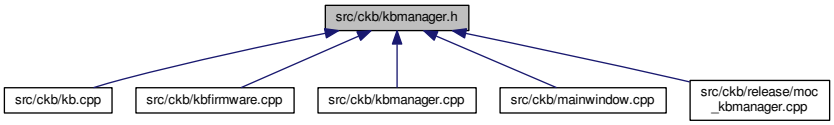
13.101 src/ckb/kbmanager.h File Reference

```
#include <QObject>
#include <QTimer>
#include <cmath>
#include <QSet>
#include "kb.h"
```

Include dependency graph for kbmanager.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbManager

Macros

- #define DAEMON_UNAVAILABLE_STR "<unavailable>"

13.101.1 Macro Definition Documentation

13.101.1.1 #define DAEMON_UNAVAILABLE_STR "<unavailable>"

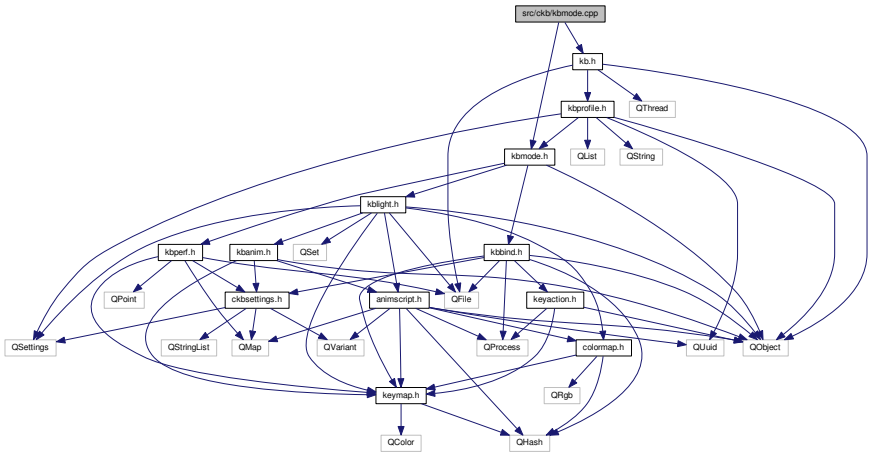
Definition at line 12 of file kbmanager.h.

Referenced by KbManager::ckbDaemonVersionF(), KbManager::scanKeyboards(), and MainWindow::updateVersion().

13.102 src/ckb/kbmode.cpp File Reference

```
#include "kbmode.h"
#include "kb.h"
```

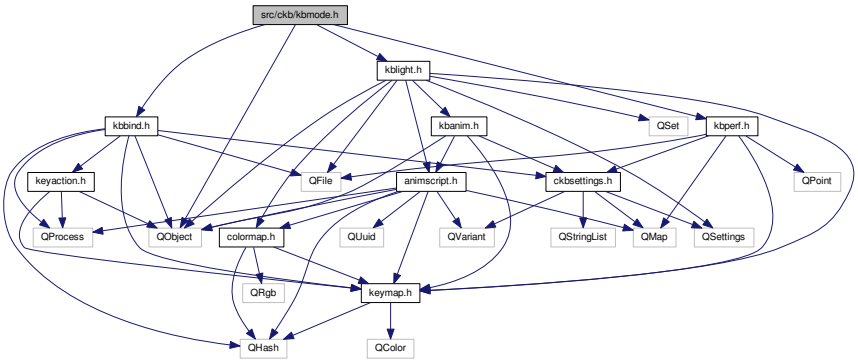
Include dependency graph for kbmode.cpp:



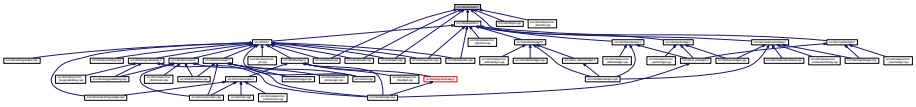
13.103 src/ckb/kbmode.h File Reference

```
#include <QObject>
#include "kblight.h"
#include "kbbind.h"
#include "kbperf.h"
```

Include dependency graph for kbmode.h:



This graph shows which files directly or indirectly include this file:



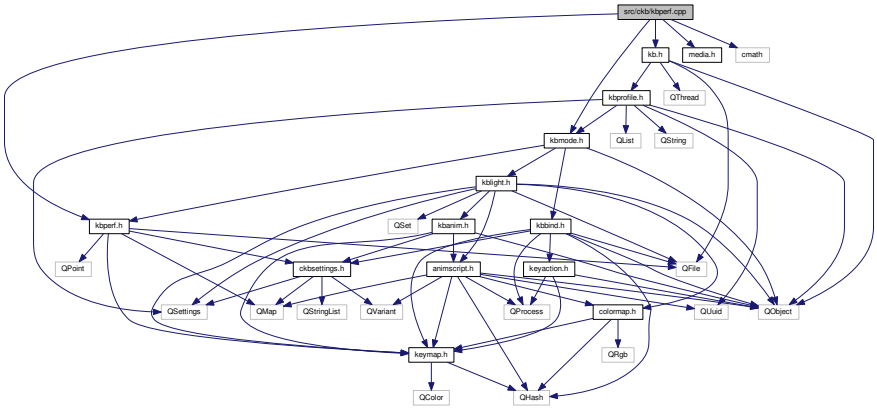
Data Structures

- struct UsbId
- class KbMode

13.104 src/ckb/kbperf.cpp File Reference

```
#include "kbperf.h"
#include "kbmode.h"
#include "kb.h"
#include "media.h"
#include <cmath>
```

Include dependency graph for kbperf.cpp:



Macros

- #define map_last(map) ((map).last())

13.104.1 Macro Definition Documentation

13.104.1.1 #define map_last(*map*) ((map).last())

Definition at line 8 of file kbperf.cpp.

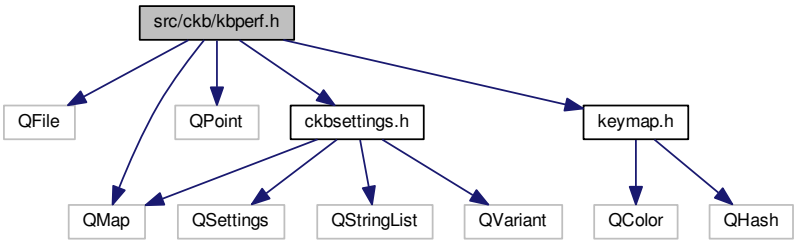
Referenced by KbPerf::popDpi().

13.105 src/ckb/kbperf.h File Reference

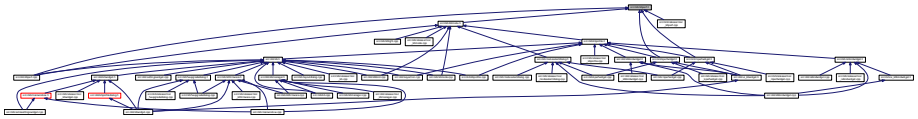
```
#include <QFile>
```

```
#include <QMap>
#include <QPoint>
#include "ckbsettings.h"
#include "keymap.h"
```

Include dependency graph for kbperf.h:



This graph shows which files directly or indirectly include this file:



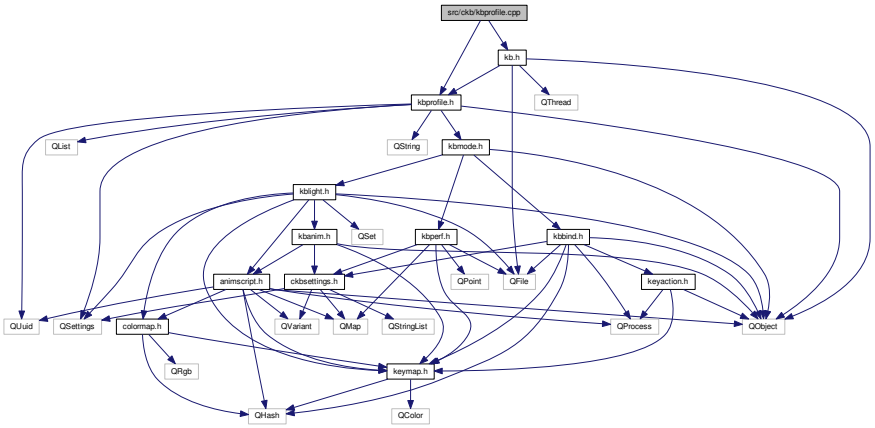
Data Structures

- class KbPerf

13.106 src/ckb/kbprofile.cpp File Reference

```
#include "kbprofile.h"
#include "kb.h"
```

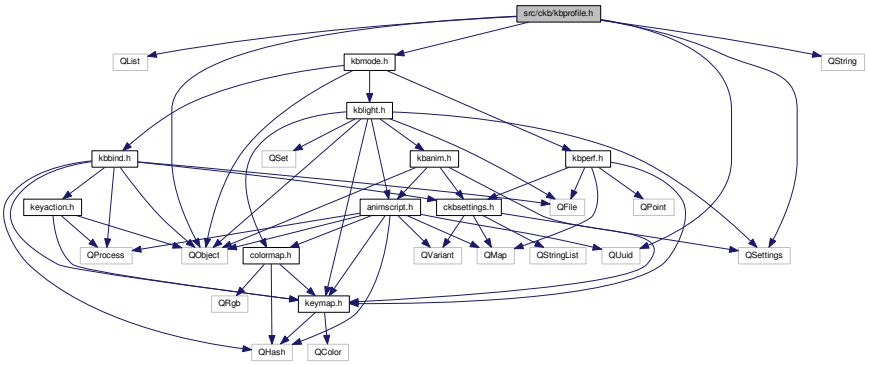
Include dependency graph for kbprofile.cpp:



13.107 src/ckb/kbprofile.h File Reference

```
#include <QList>
#include <QObject>
#include <QSettings>
#include <QString>
#include <QUuid>
#include "kbmode.h"
```

Include dependency graph for kbprofile.h:



This graph shows which files directly or indirectly include this file:

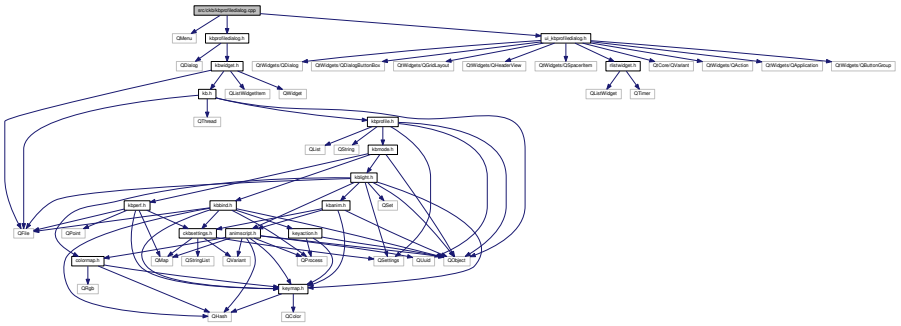
Data Structures

- class KbProfile

13.108 src/ckb/kbprofiledialog.cpp File Reference

```
#include <QMenu>
#include "kbprofiledialog.h"
#include "ui_kbprofiledialog.h"
```

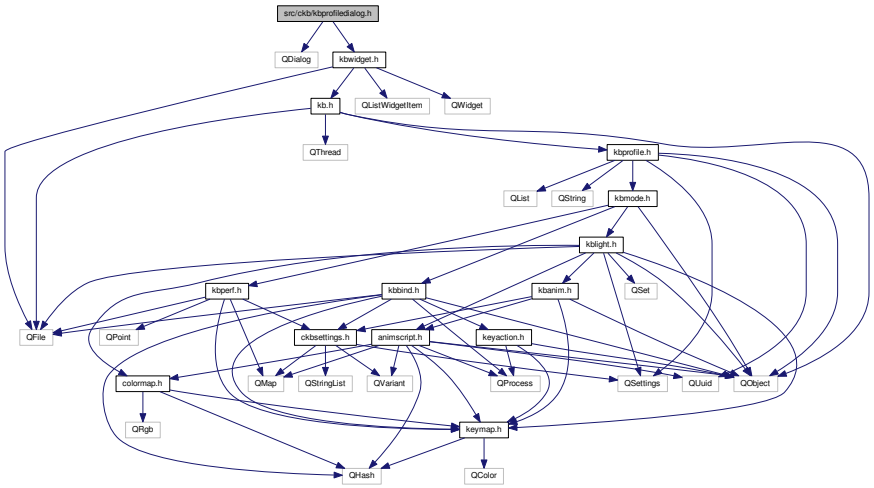
Include dependency graph for kbprofiledialog.cpp:



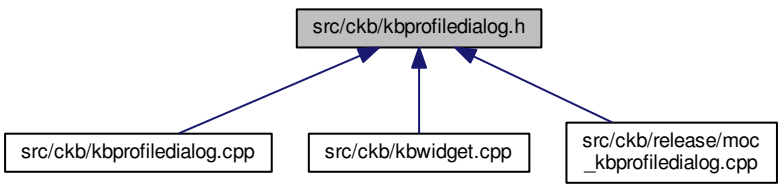
13.109 src/ckb/kbprofiledialog.h File Reference

```
#include <QDialog>
#include "kbwidget.h"
```

Include dependency graph for kbprofiledialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbProfileDialog

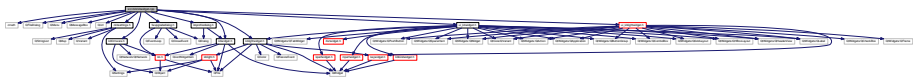
Namespaces

- Ui

13.110 src/ckb/kbwidget.cpp File Reference

```
#include <cmath>
#include <QFileDialog>
#include <QMenu>
#include <QMessageBox>
#include <QUrl>
#include "ckbsettings.h"
#include "fwupgradedialog.h"
#include "kbfirmware.h"
#include "kbwidget.h"
#include "kblightwidget.h"
#include "kbprofiledialog.h"
#include "ui_kbwidget.h"
#include "ui_kblightwidget.h"
```

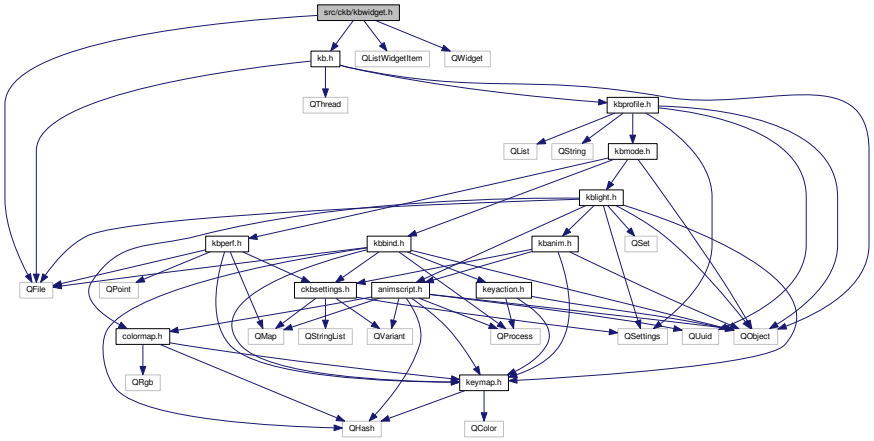
Include dependency graph for kbwidget.cpp:



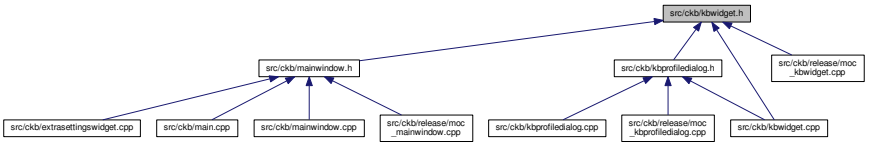
13.111 src/ckb/kbwidget.h File Reference

```
#include <QFile>
#include <QListWidgetItem>
#include <QWidget>
#include "kb.h"
```

Include dependency graph for kbwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KbWidget

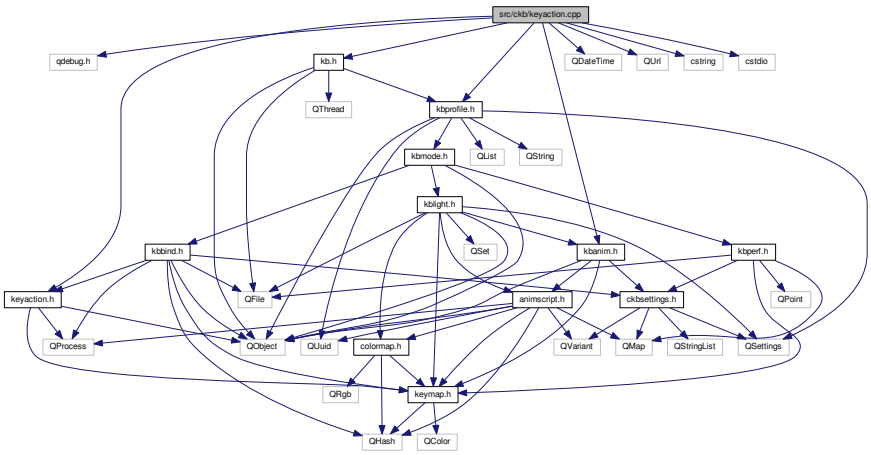
Namespaces

- Ui

13.112 src/ckb/keyaction.cpp File Reference

```
#include <qdebug.h>
#include "keyaction.h"
#include "kb.h"
#include "kbanim.h"
#include "kbprofile.h"
#include <QDateTime>
#include <QUrl>
#include <cstring>
#include <cstdio>
```

Include dependency graph for keyaction.cpp:



Variables

- static const int ANIM_ONCE = 0x1
- static const int ANIM_KRSTOP = 0x2

13.112.1 Variable Documentation

13.112.1.1 const int ANIM_KRSTOP = 0x2 [static]

Definition at line 182 of file keyaction.cpp.

Referenced by KeyAction::animAction(), and KeyAction::animInfo().

13.112.1.2 const int ANIM_ONCE = 0x1 [static]

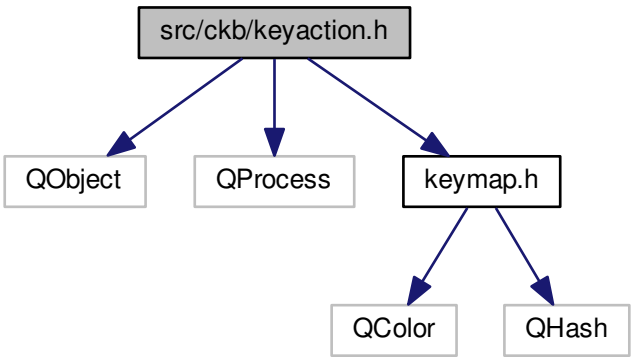
Definition at line 182 of file keyaction.cpp.

Referenced by KeyAction::animAction(), and KeyAction::animInfo().

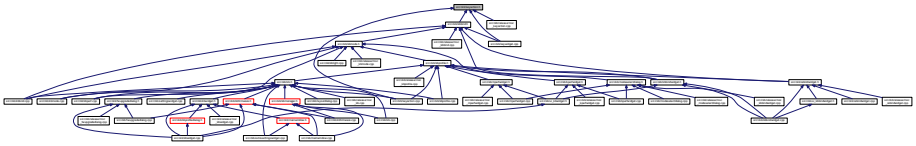
13.113 src/ckb/keyaction.h File Reference

```
#include <QObject>
#include <QProcess>
#include "keymap.h"
```

Include dependency graph for keyaction.h:



This graph shows which files directly or indirectly include this file:



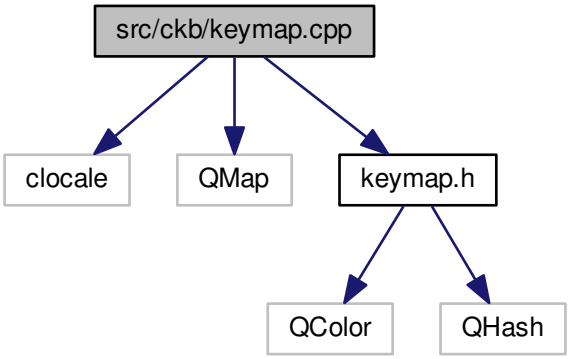
Data Structures

- class KeyAction

13.114 src/ckb/keymap.cpp File Reference

```
#include <clocale>
#include <QMap>
#include "keymap.h"
```

Include dependency graph for keymap.cpp:



Data Structures

- struct KeyPatch

Macros

- #define NS 12, 12
- #define LBS 17, 6
- #define KEYCOUNT_K95 (sizeof(K95Keys) / sizeof(Key))
- #define ANSI_ENTER_X 199
- #define ANSI_ENTER_Y 51
- #define ANSI_ENTER_W 26
- #define ANSI_ENTER_H 12
- #define ANSI_LSHIFT_X 45
- #define ANSI_LSHIFT_W 26
- #define PATCH_COUNT(patch) (sizeof(patch) / sizeof(KeyPatch))
- #define patch(map, patches) _patch(map, patches, PATCH_COUNT(patches))
- #define K95_WIDTH 298
- #define K95_HEIGHT 76
- #define K95P_HEIGHT 82
- #define K95P_X_START 20
- #define K95P_WIDTH (K95_WIDTH - K95P_X_START + 1)
- #define K70_X_START 38
- #define K70_WIDTH (K95_WIDTH - K70_X_START)
- #define K70_HEIGHT K95_HEIGHT
- #define K68_WIDTH K70_WIDTH
- #define K68_HEIGHT K70_HEIGHT
- #define K65_WIDTH 209

- #define K65_HEIGHT K70_HEIGHT
- #define K63_WIDTH K65_WIDTH
- #define K63_HEIGHT K65_HEIGHT
- #define K68_TOP_COUNT (sizeof(K68TopRow) / sizeof(Key))
- #define K65_TOP_COUNT (sizeof(K65TopRow) / sizeof(Key))
- #define K63_TOP_COUNT (sizeof(K63TopRow) / sizeof(Key))
- #define KSTRAFE_X_START 12
- #define KSTRAFE_WIDTH (K70_WIDTH + (KSTRAFE_X_START * 2))
- #define KSTRAFE_HEIGHT K95_HEIGHT
- #define KEYCOUNT_M65 (sizeof(M65Keys) / sizeof(Key))
- #define M65_WIDTH 52
- #define M65_HEIGHT 67
- #define KEYCOUNT_SABRE (sizeof(SabreKeys) / sizeof(Key))
- #define SABRE_WIDTH M65_WIDTH
- #define SABRE_HEIGHT M65_HEIGHT
- #define KEYCOUNT_HARPOON (sizeof(HarpoonKeys) / sizeof(Key))
- #define HARPOON_WIDTH M65_WIDTH
- #define HARPOON_HEIGHT M65_HEIGHT
- #define KEYCOUNT_GLAIVE (sizeof(GlaiveKeys) / sizeof(Key))
- #define GLAIVE_WIDTH M65_WIDTH
- #define GLAIVE_HEIGHT M65_HEIGHT
- #define KEYCOUNT_SCIM (sizeof(ScimKeys) / sizeof(Key))
- #define SCIM_WIDTH M65_WIDTH
- #define SCIM_HEIGHT M65_HEIGHT
- #define LBARCOUNT_K95P (sizeof(K95PLbar) / sizeof(Key))
- #define N_MODELS KeyMap::_MODEL_MAX
- #define N_LAYOUTS KeyMap::_LAYOUT_MAX

Functions

- static void _patch (QHash< QString, Key > &map, const KeyPatch *patches, int patchCount)
- static void patchnonJP106 (QHash< QString, Key > &map)
- static void patchISO (QHash< QString, Key > &map)
- static void patchANSI (QHash< QString, Key > &map)
- static void patchJP106 (QHash< QString, Key > &map)
- static void patchJP106fn (QHash< QString, Key > &map)
- static void patchABNT2 (QHash< QString, Key > &map)
- static QHash< QString, Key > getMap (KeyMap::Model model, KeyMap::Layout layout)

Variables

- static const Key K95Keys []
- static const KeyPatch patchDK []
- static const KeyPatch patchEU []
- static const KeyPatch patchFR []
- static const KeyPatch patchDE []
- static const KeyPatch patchIT []
- static const KeyPatch patchNO []
- static const KeyPatch patchMX []
- static const KeyPatch patchPTBR []
- static const KeyPatch patchES []
- static const KeyPatch patchSE []
- static const KeyPatch patchJP []
- static const KeyPatch patchDvorak []

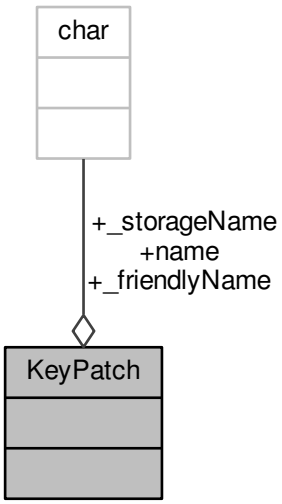
- static const Key K68TopRow []
- static const Key K65TopRow []
- static const Key K63TopRow []
- static const Key KStrafeKeys []
- static const Key M65Keys []
- static const Key SabreKeys []
- static const Key HarpoonKeys []
- static const Key GlaiveKeys []
- static const Key ScimKeys []
- static const Key K95PLbar []
- static QHash< QString, Key > K95BaseMap
- static QHash< QString, Key > standardMaps [KeyMap::_MODEL_MAX][KeyMap::_LAYOUT_MAX]

13.114.1 Data Structure Documentation

13.114.1.1 struct KeyPatch

Definition at line 32 of file keymap.cpp.

Collaboration diagram for KeyPatch:



Data Fields

const char *	_friendlyName	
const char *	_storageName	
const char *	name	

13.114.2 Macro Definition Documentation

13.114.2.1 #define ANSI_ENTER_H 12

Definition at line 27 of file keymap.cpp.

Referenced by patchANSI().

13.114.2.2 #define ANSI_ENTER_W 26

Definition at line 26 of file keymap.cpp.

Referenced by patchANSI().

13.114.2.3 #define ANSI_ENTER_X 199

Definition at line 24 of file keymap.cpp.

Referenced by patchANSI().

13.114.2.4 #define ANSI_ENTER_Y 51

Definition at line 25 of file keymap.cpp.

Referenced by patchANSI().

13.114.2.5 #define ANSI_LSHIFT_W 26

Definition at line 29 of file keymap.cpp.

Referenced by patchANSI(), and patchJP106().

13.114.2.6 #define ANSI_LSHIFT_X 45

Definition at line 28 of file keymap.cpp.

Referenced by patchANSI(), and patchJP106().

13.114.2.7 #define GLAIVE_HEIGHT M65_HEIGHT

Definition at line 310 of file keymap.cpp.

13.114.2.8 #define GLAIVE_WIDTH M65_WIDTH

Definition at line 309 of file keymap.cpp.

13.114.2.9 #define HARPOON_HEIGHT M65_HEIGHT

Definition at line 298 of file keymap.cpp.

13.114.2.10 #define HARPOON_WIDTH M65_WIDTH

Definition at line 297 of file keymap.cpp.

13.114.2.11 #define K63_HEIGHT K65_HEIGHT

Definition at line 233 of file keymap.cpp.

13.114.2.12 #define K63_TOP_COUNT (sizeof(K63TopRow) / sizeof(Key))

Definition at line 248 of file keymap.cpp.

Referenced by getMap().

13.114.2.13 #define K63_WIDTH K65_WIDTH

Definition at line 232 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.14 #define K65_HEIGHT K70_HEIGHT

Definition at line 229 of file keymap.cpp.

13.114.2.15 #define K65_TOP_COUNT (sizeof(K65TopRow) / sizeof(Key))

Definition at line 243 of file keymap.cpp.

Referenced by getMap().

13.114.2.16 #define K65_WIDTH 209

Definition at line 228 of file keymap.cpp.

Referenced by getMap(), and KeyMap::modelWidth().

13.114.2.17 #define K68_HEIGHT K70_HEIGHT

Definition at line 225 of file keymap.cpp.

13.114.2.18 #define K68_TOP_COUNT (sizeof(K68TopRow) / sizeof(Key))

Definition at line 238 of file keymap.cpp.

Referenced by getMap().

13.114.2.19 #define K68_WIDTH K70_WIDTH

Definition at line 224 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.20 #define K70_HEIGHT K95_HEIGHT

Definition at line 222 of file keymap.cpp.

13.114.2.21 #define K70_WIDTH (K95_WIDTH - K70_X_START)

Definition at line 221 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.22 #define K70_X_START 38

Definition at line 220 of file keymap.cpp.

Referenced by getMap().

13.114.2.23 #define K95_HEIGHT 76

Definition at line 213 of file keymap.cpp.

Referenced by KeyMap::modelHeight().

13.114.2.24 #define K95_WIDTH 298

Definition at line 212 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.25 #define K95P_HEIGHT 82

Definition at line 215 of file keymap.cpp.

Referenced by KeyMap::modelHeight().

13.114.2.26 #define K95P_WIDTH (K95_WIDTH - K95P_X_START + 1)

Definition at line 217 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.27 #define K95P_X_START 20

Definition at line 216 of file keymap.cpp.

Referenced by getMap().

13.114.2.28 #define KEYCOUNT_GLAIVE (sizeof(GlaiveKeys) / sizeof(Key))

Definition at line 307 of file keymap.cpp.

Referenced by getMap().

13.114.2.29 #define KEYCOUNT_HARPOON (sizeof(HarpoonKeys) / sizeof(Key))

Definition at line 295 of file keymap.cpp.

Referenced by getMap().

13.114.2.30 #define KEYCOUNT_K95 (sizeof(K95Keys) / sizeof(Key))

Definition at line 21 of file keymap.cpp.

Referenced by getMap().

13.114.2.31 #define KEYCOUNT_M65 (sizeof(M65Keys) / sizeof(Key))

Definition at line 270 of file keymap.cpp.

Referenced by getMap().

13.114.2.32 #define KEYCOUNT_SABRE (sizeof(SabreKeys) / sizeof(Key))

Definition at line 283 of file keymap.cpp.

Referenced by getMap().

13.114.2.33 #define KEYCOUNT_SCIM (sizeof(ScimKeys) / sizeof(Key))

Definition at line 324 of file keymap.cpp.

Referenced by getMap().

13.114.2.34 #define KSTRAFE_HEIGHT K95_HEIGHT

Definition at line 253 of file keymap.cpp.

13.114.2.35 #define KSTRAFE_WIDTH (K70_WIDTH + (KSTRAFE_X_START * 2))

Definition at line 252 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.36 #define KSTRAFE_X_START 12

Definition at line 251 of file keymap.cpp.

Referenced by getMap().

13.114.2.37 #define LBARCOUNT_K95P (sizeof(K95PLbar) / sizeof(Key))

Definition at line 336 of file keymap.cpp.

Referenced by getMap().

13.114.2.38 #define LBS 17, 6

Definition at line 9 of file keymap.cpp.

13.114.2.39 #define M65_HEIGHT 67

Definition at line 273 of file keymap.cpp.

Referenced by KeyMap::modelHeight().

13.114.2.40 #define M65_WIDTH 52

Definition at line 272 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

13.114.2.41 #define N_LAYOUTS KeyMap::_LAYOUT_MAX

Definition at line 341 of file keymap.cpp.

Referenced by getMap().

13.114.2.42 #define N_MODELS KeyMap::_MODEL_MAX

Definition at line 340 of file keymap.cpp.

Referenced by getMap().

13.114.2.43 #define NS 12, 12

Definition at line 6 of file keymap.cpp.

13.114.2.44 #define patch(*map, patches*) _patch(map, patches, PATCH_COUNT(patches))

Definition at line 120 of file keymap.cpp.

Referenced by getMap().

13.114.2.45 #define PATCH_COUNT(patch) (sizeof(patch) / sizeof(KeyPatch))

Definition at line 119 of file keymap.cpp.

13.114.2.46 #define SABRE_HEIGHT M65_HEIGHT

Definition at line 286 of file keymap.cpp.

13.114.2.47 #define SABRE_WIDTH M65_WIDTH

Definition at line 285 of file keymap.cpp.

13.114.2.48 #define SCIM_HEIGHT M65_HEIGHT

Definition at line 327 of file keymap.cpp.

13.114.2.49 #define SCIM_WIDTH M65_WIDTH

Definition at line 326 of file keymap.cpp.

13.114.3 Function Documentation

13.114.3.1 static void _patch (QHash< QString, Key > & *map,* const KeyPatch * *patches,* int *patchCount*) [static]

Definition at line 121 of file keymap.cpp.

References Key::_friendlyName, and Key::_storageName.

```
121                                                                                 {
122     for(const KeyPatch* p = patches; p < patches + patchCount; p++){
123         Key& key = map[p->name];
124         key._storageName = p->_storageName;
```

```
125         key._friendlyName = p->_friendlyName;
126     }
127 }
```

13.114.3.2 static QHash<QString, Key> getMap (KeyMap::Model *model*, KeyMap::Layout *layout*) [static]

Definition at line 344 of file keymap.cpp.

References KeyMap::DE, KeyMap::DK, KeyMap::ES, KeyMap::EU, KeyMap::EU_DVORAK, KeyMap::FR, KeyMap::GB_DVORAK, KeyMap::GLAIVE, KeyMap::HARPOON, Key::height, KeyMap::isISO(), KeyMap::isJP(), KeyMap::isPTBR(), KeyMap::IT, KeyMap::JP, KeyMap::K63, K63_TOP_COUNT, KeyMap::K65, K65_TOP_COUNT, K65_WIDTH, KeyMap::K68, K68_TOP_COUNT, KeyMap::K70, K70_X_START, KeyMap::K95, K95BaseMap, KeyMap::K95P, K95P_X_START, KEYCOUNT_GLAIVE, KEYCOUNT_HARPOON, KEYCOUNT_K95, KEYCOUNT_M65, KEYCOUNT_SABRE, KEYCOUNT_SCIM, KSTRAFE_X_START, LBARCOUNT_K95P, KeyMap::M65, KeyMap::MX, N_LAYOUTS, N_MODELS, key::name, KeyMap::NO, patch, patchABNT2(), patchANSI(), patchISO(), patchJP106(), patchJP106fn(), KeyMap::PT_BR, KeyMap::SABRE, KeyMap::SCIMITAR, KeyMap::SE, standardMaps, KeyMap::STRAFE, KeyMap::US_DVORAK, Key::width, Key::x, and Key::y.

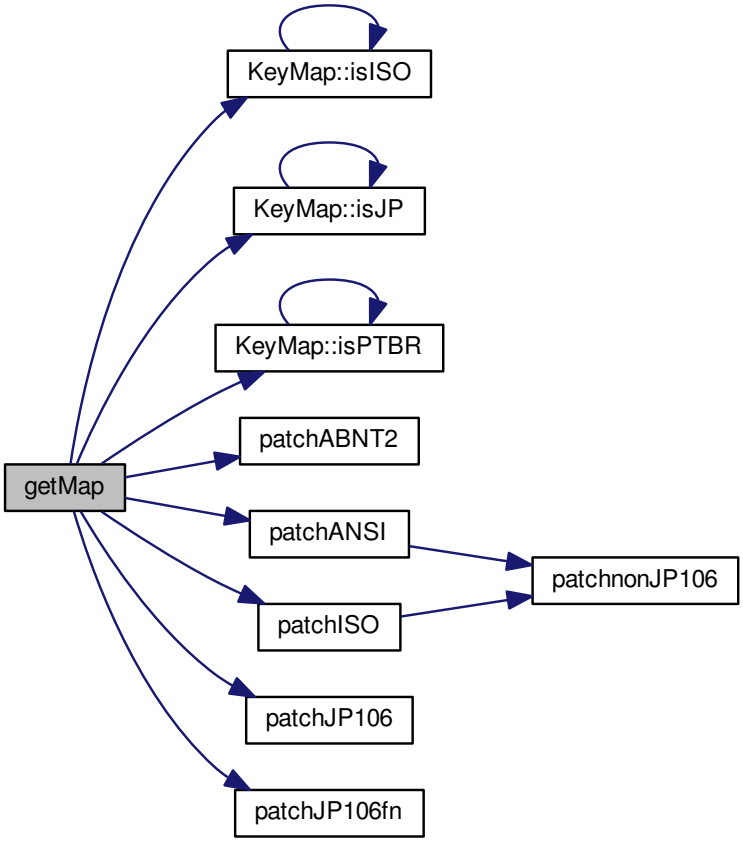
```
344                                                                     {
345     if(model < 0 || layout < 0 || model >= N_MODELS || layout >=
    N_LAYOUTS)
346         return QHash<QString, Key>();
347     // Return the map if it's already filled out
348     QHash<QString, Key>& map = standardMaps[model][layout];
349     if(!map.empty())
350         return map;
351     // Otherwise, create it
352     switch(model){
353     case KeyMap::K95:{
354         // The K95 maps serve as bases for all the other keyboards
355         // Fetch the master map, or create it if not yet done
356         if(K95BaseMap.empty()){
357             for(const Key* key = K95Keys; key < K95Keys +
    KEYCOUNT_K95; key++)
358                 K95BaseMap[key->name] = *key;
359         }
360         map = K95BaseMap;
361         // Patch the map for the layout
362         switch(layout){
363         case KeyMap::DK:
364             patch(map, patchDK);
365             break;
366         case KeyMap::EU_DVORAK:
367             patch(map, patchDvorak);    // fall through
368         case KeyMap::EU:
369             patch(map, patchEU);
370             break;
371         case KeyMap::GB_DVORAK:
372         case KeyMap::US_DVORAK:
373             patch(map, patchDvorak);
374             break;
375         case KeyMap::FR:
376             patch(map, patchFR);
377             break;
378         case KeyMap::DE:
379             patch(map, patchDE);
380             break;
381         case KeyMap::IT:
382             patch(map, patchIT);
383             break;
384         case KeyMap::NO:
385             patch(map, patchNO);
386             break;
387         case KeyMap::MX:
388             patch(map, patchMX);
389             break;
390         case KeyMap::PT_BR:
391             patch(map, patchPTBR);
392             break;
393         case KeyMap::ES:
394             patch(map, patchES);
395             break;
396         case KeyMap::SE:
397             patch(map, patchSE);
398             break;
399         case KeyMap::JP:
400             patch(map, patchJP);
401             break;
```

```
402         default:;
403             // English QWERTY - no patch needed
404         }
405         if(KeyMap::isJP(layout))
406             patchJP106(map);
407         else if(KeyMap::isPTBR(layout))
408             patchABNT2(map);
409         else if(KeyMap::isISO(layout))
410             patchISO(map);
411         else
412             patchANSI(map);
413         // Done! return the map
414         break;
415     }
416     case KeyMap::K95P:{
417         // The K95 Platinum map is based on the K95
418         map = getMap(KeyMap::K95, layout);
419         // Remove excess G keys
420         map.remove("g7");
421         map.remove("g8");
422         map.remove("g9");
423         map.remove("g10");
424         map.remove("g11");
425         map.remove("g12");
426         map.remove("g13");
427         map.remove("g14");
428         map.remove("g15");
429         map.remove("g16");
430         map.remove("g17");
431         map.remove("g18");
432         // Place the remaining G keys vertically
433         map["g1"].x = 22;
434         map["g2"].x = 22;
435         map["g3"].x = 22;
436         map["g4"].x = 22;
437         map["g5"].x = 22;
438         map["g6"].x = 22;
439         //map["g1"].y = 14+(12*0);
440         map["g2"].y = 26; //14+(12*1)
441         map["g3"].y = 38; //14+(12*2)
442         map["g4"].y = 50; //14+(12*3)
443         map["g5"].y = 62; //14+(12*4)
444         map["g6"].y = 74; //14+(12*5)
445         // Remove M keys
446         map.remove("m1");
447         map.remove("m2");
448         map.remove("m3");
449         // Centre MR/Brightness/Winlock between F1 and F4
450         map["mr"].x = 64;
451         map["light"].x = 75;
452         map["lock"].x = 86;
453         // Resize them
454         map["mr"].height = 8;
455         map["light"].height = 8;
456         map["lock"].height = 8;
457         map["mr"].width = 11;
458         map["light"].width = 11;
459         map["lock"].width = 11;
460         // Mute is above Stop
461         map["mute"].x -= 11;
462         map["voldn"].x -= 11;
463         map["volup"].x -= 11;
464         // Move the buttons on the top a bit further down to centre them
465         map["mr"].y += 1;
466         map["light"].y += 1;
467         map["lock"].y += 1;
468         map["mute"].y += 1;
469         map["voldn"].y += 1;
470         map["volup"].y += 1;
471         // Shift all keys down (to make room for the lightbar), and to the left
472         QMutableHashIterator<QString, Key> i(map);
473         while(i.hasNext()){
474             i.next();
475             i.value().x -= K95P_X_START;
476             i.value().y += 6;
477         }
478         // Add lightbar
479         for(const Key* key = K95PLbar; key < K95PLbar +
    LBARCOUNT_K95P; key++)
480             map.insert(key->name, *key);
481
482         break;
483     }
484     case KeyMap::K70:{
485         // The K70 maps are based on the K95 maps. However all the keys are shifted left and the G keys are
     removed
486         map = getMap(KeyMap::K95, layout);
```

```
487         QMutableHashIterator<QString, Key> i(map);
488         while(i.hasNext()){
489             i.next();
490             // Move key to left. Remove it if it fell off the edge
491             if((i.value().x -= K70_X_START) < 0)
492                 i.remove();
493         }
494         // Remove the M buttons as well
495         map.remove("mr");
496         map.remove("m1");
497         map.remove("m2");
498         map.remove("m3");
499         // Done!
500         break;
501     }
502     case KeyMap::K68:{
503         // Same as the K70, except volume up and down keys
504         map = getMap(KeyMap::K70, layout);
505         for(const Key* key = K68TopRow; key < K68TopRow +
    K68_TOP_COUNT; key++)
506             map[key->name] = *key;
507 
508         break;
509     }
510     case KeyMap::K65:{
511         // The K65 maps additionally remove the numpad and have a modified top row
512         map = getMap(KeyMap::K70, layout);
513         QMutableHashIterator<QString, Key> i(map);
514         while(i.hasNext()){
515             i.next();
516             if(i.value().x >= K65_WIDTH)
517                 i.remove();
518         }
519         for(const Key* key = K65TopRow; key < K65TopRow +
    K65_TOP_COUNT; key++)
520             map[key->name] = *key;
521 
522         map.remove("rwin");
523         map["fn"] = KStrafeKeys[3];
524         map["fn"].x -= 12;
525 
526         if(KeyMap::isJP(layout))
527             patchJP106fn(map);
528 
529         // Done!
530         break;
531     }
532     case KeyMap::K63:{
533         // Same as the K65 but without the Fn key
534         map = getMap(KeyMap::K70, layout);
535         QMutableHashIterator<QString, Key> i(map);
536         while(i.hasNext()){
537             i.next();
538             if(i.value().x >= K65_WIDTH)
539                 i.remove();
540         }
541         for(const Key* key = K63TopRow; key < K63TopRow +
    K63_TOP_COUNT; key++)
542             map[key->name] = *key;
543 
544         break;
545     }
546     case KeyMap::STRAFE:{
547         // The Strafe RGB maps are based on the K70 map minus the media keys
548         map = getMap(KeyMap::K70, layout);
549         //move light and lock right
550         map["light"].x=285 - K70_X_START;
551         //map["light"].hasLed=false;
552         map["lock"].x=297 - K70_X_START;
553         //map["lock"].hasLed=false;
554         // move everything right to make the space for the left sidelight
555         QMutableHashIterator<QString, Key> i(map);
556         while(i.hasNext()){
557             i.next();
558             i.value().x += KSTRAFE_X_START;
559         }
560         // Add Strafe lights and keys
561         map["lsidel"] = KStrafeKeys[0];
562         map["rsidel"] = KStrafeKeys[1];
563         map["logo"] = KStrafeKeys[2];
564         map["fn"] = KStrafeKeys[3];
565         map.remove("rwin");
566         // remove media controls
567         map.remove("mute");
568         map.remove("volup");
569         map.remove("voldn");
570         map.remove("stop");
```

```
571         map.remove("prev");
572         map.remove("play");
573         map.remove("next");
574 
575         if(KeyMap::isJP(layout))
576             patchJP106fn(map);
577 
578         // Done!
579         break;
580     }
581     case KeyMap::M65:{
582         // M65 isn't a keyboard; all mouse maps are unique.
583         for(const Key* key = M65Keys; key < M65Keys +
    KEYCOUNT_M65; key++){
584             // Keyboard keys are written from the center because that's where the LEDs are, but the mouse
     buttons are odd shapes so they're
585             // written from the upper left
586             Key translatedKey = *key;
587             translatedKey.x += translatedKey.width / 2;
588             translatedKey.y += translatedKey.height / 2;
589             map[key->name] = translatedKey;
590         }
591         // Mice also have no layout patches - no other changes necessary
592         break;
593     }
594     case KeyMap::SABRE:{
595         // Sabre mouse
596         for(const Key* key = SabreKeys; key < SabreKeys +
    KEYCOUNT_SABRE; key++){
597             // Like the M65, the keys are upper-left justified
598             Key translatedKey = *key;
599             translatedKey.x += translatedKey.width / 2;
600             translatedKey.y += translatedKey.height / 2;
601             map[key->name] = translatedKey;
602         }
603         break;
604     }
605     case KeyMap::SCIMITAR:{
606         // Scimitar mouse
607         for(const Key* key = ScimKeys; key < ScimKeys +
    KEYCOUNT_SCIM; key++){
608             Key translatedKey = *key;
609             translatedKey.x += translatedKey.width / 2;
610             translatedKey.y += translatedKey.height / 2;
611             map[key->name] = translatedKey;
612         }
613         break;
614     }
615     case KeyMap::HARPOON:{
616         // Harpoon mouse
617         for(const Key* key = HarpoonKeys; key < HarpoonKeys +
    KEYCOUNT_HARPOON; key++){
618             Key translatedKey = *key;
619             translatedKey.x += translatedKey.width / 2;
620             translatedKey.y += translatedKey.height / 2;
621             map[key->name] = translatedKey;
622         }
623         break;
624     }
625     case KeyMap::GLAIVE:{
626         // Glaive mouse
627         for(const Key* key = GlaiveKeys; key < GlaiveKeys +
    KEYCOUNT_GLAIVE; key++){
628             Key translatedKey = *key;
629             translatedKey.x += translatedKey.width / 2;
630             translatedKey.y += translatedKey.height / 2;
631             map[key->name] = translatedKey;
632         }
633         break;
634     }
635     default:;    // <- stop GCC from complaining
636     }
637     // Map is finished, return result
638     return map;
639 }
```

Here is the call graph for this function:



13.114.3.3 static void patchABNT2 (QHash< QString, Key > & *map*) [static]

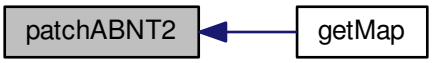
Definition at line 198 of file keymap.cpp.

References Key::width, and Key::x.

Referenced by getMap().

```
198                                                 {
199     map.remove("yen");
200     map.remove("henkan");
201     map.remove("muhenkan");
202     map.remove("katahira");
203     map.remove("bslash");
204
205     // Resize RShift
206     Key& rshift = map["rshift"];
207     rshift.width -= 12;
208     rshift.x += 6;
209 }
```

Here is the caller graph for this function:



13.114.3.4 static void patchANSI (QHash< QString, Key > & *map*) [static]

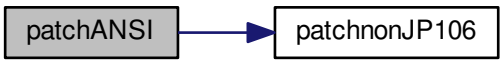
Definition at line 141 of file keymap.cpp.

References ANSI_ENTER_H, ANSI_ENTER_W, ANSI_ENTER_X, ANSI_ENTER_Y, ANSI_LSHIFT_W, ANSI_LSHIFT_X, Key::height, patchnonJP106(), Key::width, Key::x, and Key::y.

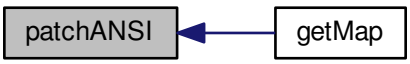
Referenced by getMap().

```
141                                                 {
142     patchnonJP106(map);
143     map.remove("bslash_iso");
144     map.remove("hash");
145     Key& enter = map["enter"];
146     enter.x = ANSI_ENTER_X;
147     enter.y = ANSI_ENTER_Y;
148     enter.width = ANSI_ENTER_W;
149     enter.height = ANSI_ENTER_H;
150     Key& lshift = map["lshift"];
151     lshift.x = ANSI_LSHIFT_X;
152     lshift.width = ANSI_LSHIFT_W;
153 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.114.3.5 static void patchISO (QHash< QString, Key > & *map*) [static]

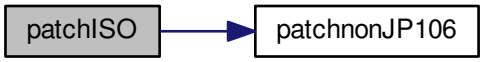
Definition at line 137 of file keymap.cpp.

References patchnonJP106().

Referenced by getMap().

```
137                                                {
138     patchnonJP106(map);
139     map.remove("bslash");
140 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.114.3.6 static void patchJP106 (QHash< QString, Key > & *map*) [static]

Definition at line 154 of file keymap.cpp.

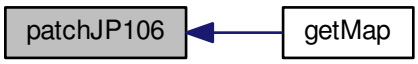
References ANSI_LSHIFT_W, ANSI_LSHIFT_X, Key::width, and Key::x.

Referenced by getMap().

```
154                                                {
155     // First apply the ISO patch
156     map.remove("bslash");
157
158     // Resize Backspace
159     Key& bspace = map["bspace"];
160     bspace.width -= 12;
161     bspace.x = 206;
162
163     // Resize RShift
164     Key& rshift = map["rshift"];
165     rshift.width -= 12;
166     rshift.x += 6;
167
168     // Resize Spacebar
169     Key& space = map["space"];
170     space.width -= 26;
171     space.x -= 1;
172
173     // Left shift
```

```
174     Key& lshift = map["lshift"];
175     lshift.x = ANSI_LSHIFT_X;
176     lshift.width = ANSI_LSHIFT_W;
177 
178     map.remove("bslash_iso");
179 
180     // Resize and move ralt to make space for the extra keys
181     Key& ralt = map["ralt"];
182     ralt.x += 11;
183     ralt.width += 2;
184 
185     // None of these layouts have rwin
186     map.remove("rwin");
187 }
```

Here is the caller graph for this function:



13.114.3.7 static void patchJP106fn (QHash< QString, Key > & *map*) [static]

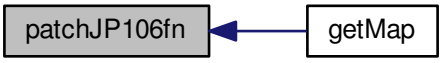
Definition at line 189 of file keymap.cpp.

References Key::width, and Key::x.

Referenced by getMap().

```
189                                                 {
190     if(map.contains("fn")){
191         map.remove("ralt");
192         Key& fn = map["fn"];
193         fn.width += 4;
194         fn.x -= 2;
195     }
196 }
```

Here is the caller graph for this function:



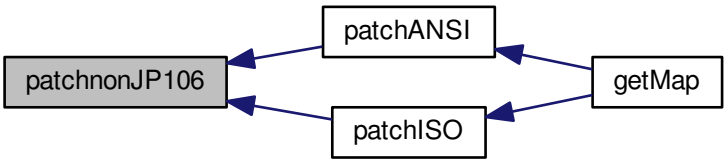
13.114.3.8 static void patchnonJP106 (QHash< QString, Key > & *map*) [static]

Definition at line 130 of file keymap.cpp.

Referenced by patchANSI(), and patchISO().

```
130                                                 {
131     map.remove("yen");
132     map.remove("henkan");
133     map.remove("muhenkan");
134     map.remove("katahira");
135     map.remove("ro");
136 }
```

Here is the caller graph for this function:



13.114.4 Variable Documentation

13.114.4.1 const Key GlaiveKeys[] [static]

Initial value:

```
= {
    {0, "Left Mouse", "mouse1", 17, 3, 14, 18, false, true}, {0, "Right Mouse", "mouse2", 37, 3, 14, 18,
      false, true}, {0, "Middle Mouse", "mouse3", 31, 9, 7, 7, false, true}, {0, "Front light", "front", 16, -5, 36,
       8, true, false },
    {0, "Wheel Up", "wheelup", 31, 5, 7, 5, false, true}, {0, "Wheel Down", "wheeldn", 31, 15, 7, 5, false,
       true}, {0, "Side Lights", "side", 22, 24, 7, 20, true, false},
    {0, "DPI Cycle", "dpiup", 31, 19, 6, 12, false, true}, {0, "Logo Light", "back", 24, 43, 20, 20, true,
      false},
    {0, "Forward", "mouse5", 15, 22, 5, 11, false, true}, {0, "Back", "mouse4", 15, 32, 5, 11, false, true}
    }
```

Definition at line 301 of file keymap.cpp.

13.114.4.2 const Key HarpoonKeys[] [static]

Initial value:

```
= {
    {0, "Left Mouse", "mouse1", 10, 5, 14, 26, false, true}, {0, "Right Mouse", "mouse2", 30, 5, 14, 26,
      false, true}, {0, "Middle Mouse", "mouse3", 25, 11, 6, 7, false, true},
    {0, "Wheel Up", "wheelup", 25, 7, 6, 5, false, true}, {0, "Wheel Down", "wheeldn", 25, 17, 6, 5, false,
       true},
    {0, "DPI Cycle", "dpiup", 25, 23, 6, 10, false, true}, {0, "Logo Light", "dpi", 17, 40, 20, 20, true,
      false},
    {0, "Forward", "mouse5", 3, 24, 5, 10, false, true}, {0, "Back", "mouse4", 3, 33, 5, 10, false, true}
    }
```

Definition at line 289 of file keymap.cpp.

13.114.4.3 const Key K63TopRow[] [static]

Initial value:

```
= {
    {0, "Stop", "stop",  38  - 37, 0, 12, 8, true, true}, {0, "Previous", "prev",  38  - 26, 0, 12, 8, true
      , true}, {0, "Play/Pause", "play",  38  - 15, 0, 12, 8, true, true}, {0, "Next", "next",  38  - 4, 0, 12, 8,
       true, true}, {0, "Brightness", "light", 170 -  38 , 0, 12, 12, true, true}, {0, "Windows Lock", "lock", 180
       -  38 , 0, 12, 12, true, true}, {0, "Mute", "mute", 222 -  38 , 0, 13, 8, true, true}, {0, "Volume Down", "
      voldn", 234 -  38 , 0, 13, 8, true, true}, {0, "Volume Up", "volup", 246 -  38 , 0, 13, 8, true, true}
}
```

Definition at line 245 of file keymap.cpp.

13.114.4.4 const Key K65TopRow[] [static]

Initial value:

```
= {
    {0, "Brightness", "light", 164 -  38 , 0, 12, 12, true, true}, {0, "Mute", "mute", 176 -  38 , 0, 12,
      12, true, true}, {0, "Volume Down", "voldn", 192 -  38 , 0, 14, 8, true, true}, {0, "Volume Up", "volup", 205
       -  38 , 0, 14, 8, true, true}, {0, "Windows Lock", "lock", 222 -  38 , 0, 12, 12, true, true}
}
```

Definition at line 240 of file keymap.cpp.

13.114.4.5 const Key K68TopRow[] [static]

Initial value:

```
= {
    {0, "Volume Down", "voldn", 285 -  38 , 0, 13, 8, true, true}, {0, "Volume Up", "volup", 297 -  38 , 0,
       13, 8, true, true},
}
```

Definition at line 235 of file keymap.cpp.

13.114.4.6 QHash<QString, Key> K95BaseMap [static]

Definition at line 342 of file keymap.cpp.

Referenced by getMap().

13.114.4.7 const Key K95Keys[] [static]

Initial value:

```
= {
    {0, 0, "mr", 38, 0,  12, 12 , true, true}, {0, 0, "m1", 50, 0,  12, 12 , true, true}, {0, 0, "m2", 62,
      0,  12, 12 , true, true}, {0, 0, "m3", 74, 0,  12, 12 , true, true}, {0, "Brightness", "light", 222, 0,  12,
       12 , true, true}, {0, "Windows Lock", "lock", 234, 0,  12, 12 , true, true}, {0, "Mute", "mute", 273, 0, 13
      , 8, true, true}, {0, "Volume Up", "volup", 290, -2, 18, 6, false, true}, {0, "Volume down", "voldn", 290, 2
      , 18, 6, false, true},
    {0, 0, "g1", 0, 14,  12, 12 , true, true}, {0, 0, "g2", 11, 14,  12, 12 , true, true}, {0, 0, "g3", 22,
       14,  12, 12 , true, true}, {0, "Esc", "esc", 38, 14,  12, 12 , true, true}, {0, 0, "f1", 58, 14,  12, 12 ,
      true, true}, {0, 0, "f2", 70, 14,  12, 12 , true, true}, {0, 0, "f3", 82, 14,  12, 12 , true, true}, {0, 0,
      "f4", 94, 14,  12, 12 , true, true}, {0, 0, "f5", 114, 14,  12, 12 , true, true}, {0, 0, "f6", 126, 14,  12,
       12 , true, true}, {0, 0, "f7", 138, 14,  12, 12 , true, true}, {0, 0, "f8", 150, 14,  12, 12 , true, true},
       {0, 0, "f9", 170, 14,  12, 12 , true, true}, {0, 0, "f10", 182, 14,  12, 12 , true, true}, {0, 0, "f11",
      194, 14,  12, 12 , true, true}, {0, 0, "f12", 206, 14,  12, 12 , true, true}, {0, "Print Screen\nSysRq", "
      prtscn", 222, 14,  12, 12 , true, true}, {0, "Scroll Lock", "scroll", 234, 14,  12, 12 , true, true}, {0, "Pause
      \nBreak", "pause", 246, 14,  12, 12 , true, true}, {0, "Stop", "stop", 262, 14, 12, 8, true, true}, {0, "
      Previous", "prev", 273, 14, 13, 8, true, true}, {0, "Play/Pause", "play", 285, 14, 13, 8, true, true}, {0, "
      Next", "next", 296, 14, 12, 8, true, true},
    {0, 0, "g4", 0, 25,  12, 12 , true, true}, {0, 0, "g5", 11, 25,  12, 12 , true, true}, {0, 0, "g6", 22,
       25,  12, 12 , true, true}, {0, "`", "grave", 38, 27,  12, 12 , true, true}, {0, 0, "1", 50, 27,  12, 12 ,
      true, true}, {0, 0, "2", 62, 27,  12, 12 , true, true}, {0, 0, "3", 74, 27,  12, 12 , true, true}, {0, 0, "4"
      , 86, 27,  12, 12 , true, true}, {0, 0, "5", 98, 27,  12, 12 , true, true}, {0, 0, "6", 110, 27,  12, 12 ,
      true, true}, {0, 0, "7", 122, 27,  12, 12 , true, true}, {0, 0, "8", 134, 27,  12, 12 , true, true}, {0, 0, "
      9", 146, 27,  12, 12 , true, true}, {0, 0, "0", 158, 27,  12, 12 , true, true}, {0, "-", "minus", 170, 27,
      12, 12 , true, true}, {0, "=", "equal", 182, 27,  12, 12 , true, true}, {0, "¥", "yen", 194, 27,  12, 12 ,
```

```
      true, true}, {0, "Backspace", "bspace", 200, 27, 24, 12, true, true}, {0, "Insert", "ins", 222, 27,  12, 12 ,
       true, true}, {0, "Home", "home", 234, 27,  12, 12 , true, true}, {0, "Page Up", "pgup", 246, 27,  12, 12 ,
      true, true}, {0, "Num Lock", "numlock", 261, 27,  12, 12 , true, true}, {0, "NumPad /", "numslash", 273, 27,
        12, 12 , true, true}, {0, "NumPad *", "numstar", 285, 27,  12, 12 , true, true}, {0, "NumPad -", "numminus
      ", 297, 27,  12, 12 , true, true},
    {0, 0, "g7", 0, 39,  12, 12 , true, true}, {0, 0, "g8", 11, 39,  12, 12 , true, true}, {0, 0, "g9", 22,
       39,  12, 12 , true, true}, {0, "Tab", "tab", 41, 39, 18, 12, true, true}, {0, 0, "q", 56, 39,  12, 12 ,
      true, true}, {0, 0, "w", 68, 39,  12, 12 , true, true}, {0, 0, "e", 80, 39,  12, 12 , true, true}, {0, 0, "r",
      92, 39,  12, 12 , true, true}, {0, 0, "t", 104, 39,  12, 12 , true, true}, {0, 0, "y", 116, 39,  12, 12 ,
      true, true}, {0, 0, "u", 128, 39,  12, 12 , true, true}, {0, 0, "i", 140, 39,  12, 12 , true, true}, {0, 0, "o
      ", 152, 39,  12, 12 , true, true}, {0, 0, "p", 164, 39,  12, 12 , true, true}, {0, "[", "lbrace", 176, 39,
      12, 12 , true, true}, {0, "]", "rbrace", 188, 39,  12, 12 , true, true}, {0, "\\", "bslash", 203, 39, 18, 12
      , true, true}, {0, "Enter", "enter", 203, 39, 18, 24, true, true}, {0, "Delete", "del", 222, 39,  12, 12 ,
      true, true}, {0, "End", "end", 234, 39,  12, 12 , true, true}, {0, "Page Down", "pgdn", 246, 39,  12, 12 ,
      true, true}, {0, "NumPad 7", "num7", 261, 39,  12, 12 , true, true}, {0, "NumPad 8", "num8", 273, 39,  12, 12
      , true, true}, {0, "NumPad 9", "num9", 285, 39,  12, 12 , true, true}, {0, "NumPad +", "numplus", 297, 45,
      12, 24, true, true},
    {0, 0, "g10", 0, 50,  12, 12 , true, true}, {0, 0, "g11", 11, 50,  12, 12 , true, true}, {0, 0, "g12",
      22, 50,  12, 12 , true, true}, {0, "Caps Lock", "caps", 42, 51, 20, 12, true, true}, {0, 0, "a", 59, 51,  12
      , 12 , true, true}, {0, 0, "s", 71, 51,  12, 12 , true, true}, {0, 0, "d", 83, 51,  12, 12 , true, true}, {0
      , 0, "f", 95, 51,  12, 12 , true, true}, {0, 0, "g", 107, 51,  12, 12 , true, true}, {0, 0, "h", 119, 51,
      12, 12 , true, true}, {0, 0, "j", 131, 51,  12, 12 , true, true}, {0, 0, "k", 143, 51,  12, 12 , true, true},
       {0, 0, "l", 155, 51,  12, 12 , true, true}, {0, ";", "colon", 167, 51,  12, 12 , true, true}, {0, "'", "
      quote", 179, 51,  12, 12 , true, true}, {0, "#", "hash", 191, 51,  12, 12 , true, true}, {0, "NumPad 4", "num4"
      , 261, 51,  12, 12 , true, true}, {0, "NumPad 5", "num5", 273, 51,  12, 12 , true, true}, {0, "NumPad 6", "
      num6", 285, 51,  12, 12 , true, true},
    {0, 0, "g13", 0, 64,  12, 12 , true, true}, {0, 0, "g14", 11, 64,  12, 12 , true, true}, {0, 0, "g15",
      22, 64,  12, 12 , true, true}, {0, "Left Shift", "lshift", 39, 63, 14, 12, true, true}, {"bslash", "\\", "
      bslash_iso", 53, 63,  12, 12 , true, true}, {0, 0, "z", 65, 63,  12, 12 , true, true}, {0, 0, "x", 77, 63,  12
      , 12 , true, true}, {0, 0, "c", 89, 63,  12, 12 , true, true}, {0, 0, "v", 101, 63,  12, 12 , true, true}, {
      0, 0, "b", 113, 63,  12, 12 , true, true}, {0, 0, "n", 125, 63,  12, 12 , true, true}, {0, 0, "m", 137, 63,
       12, 12 , true, true}, {0, ",", "comma", 149, 63,  12, 12 , true, true}, {0, ".", "dot", 161, 63,  12, 12 ,
      true, true}, {0, "/", "slash", 173, 63,  12, 12 , true, true}, {0, "_", "ro", 185, 63,  12, 12 , true, true}
      , {0, "Right Shift", "rshift", 196, 63, 32, 12, true, true}, {0, "Up", "up", 234, 63,  12, 12 , true, true},
       {0, "NumPad 1", "num1", 261, 63,  12, 12 , true, true}, {0, "NumPad 2", "num2", 273, 63,  12, 12 , true,
      true}, {0, "NumPad 3", "num3", 285, 63,  12, 12 , true, true}, {0, "NumPad Enter", "numenter", 297, 69, 12, 24
      , true, true},
    {0, 0, "g16", 0, 75,  12, 12 , true, true}, {0, 0, "g17", 11, 75,  12, 12 , true, true}, {0, 0, "g18",
      22, 75,  12, 12 , true, true}, {0, "Left Ctrl", "lctrl", 40, 75, 16, 12, true, true}, {0, "Left Windows", "
      lwin", 54, 75,  12, 12 , true, true}, {0, "Left Alt", "lalt", 67, 75, 14, 12, true, true}, {0, "", "
      muhenkan", 80, 75,  12, 12 , true, true}, {0, "Space", "space", 116, 75, 84, 12, true, true}, {0, "", "henkan",
      150, 75,  12, 12 , true, true}, {0, "  ", "katahira", 162, 75,  12, 12 , true, true}, {0, "Right
       Alt", "ralt", 165, 75, 14, 12, true, true}, {0, "Right Windows", "rwin", 178, 75,  12, 12 , true, true}, {0
      , "Menu", "rmenu", 190, 75,  12, 12 , true, true}, {0, "Right Ctrl", "rctrl", 204, 75, 16, 12, true, true},
      {0, "Left", "left", 222, 75,  12, 12 , true, true}, {0, "Down", "down", 234, 75,  12, 12 , true, true}, {0,
      "Right", "right", 246, 75,  12, 12 , true, true}, {0, "NumPad 0", "num0", 267, 75, 24, 12, true, true}, {0,
      "NumPad .", "numdot", 285, 75,  12, 12 , true, true}
}
```

Definition at line 12 of file keymap.cpp.

13.114.4.8 const Key K95PLbar[] [static]

Initial value:

```
= {
    {0, 0, "topbar1", 4, -3,  17, 6 , true, false}, {0, 0, "topbar2", 19, -3,  17, 6 , true, false}, {0, 0,
       "topbar3", 34, -3,  17, 6 , true, false}, {0, 0, "topbar4", 49, -3,  17, 6 , true, false}, {0, 0, "topbar5"
      , 64, -3,  17, 6 , true, false}, {0, 0, "topbar6", 79, -3,  17, 6 , true, false},
    {0, 0, "topbar7", 94, -3,  17, 6 , true, false}, {0, 0, "topbar8", 109, -3,  17, 6 , true, false}, {0,
      0, "topbar9", 124, -3,  17, 6 , true, false}, {0, 0, "topbar10", 139, -3,  17, 6 , true, false}, {0, 0, "
      topbar11", 154, -3,  17, 6 , true, false}, {0, 0, "topbar12", 169, -3,  17, 6 , true, false},
    {0, 0, "topbar13", 184, -3,  17, 6 , true, false}, {0, 0, "topbar14", 199, -3,  17, 6 , true, false}, {
      0, 0, "topbar15", 214, -3,  17, 6 , true, false}, {0, 0, "topbar16", 229, -3,  17, 6 , true, false}, {0, 0,
      "topbar17", 244, -3,  17, 6 , true, false}, {0, 0, "topbar18", 259, -3,  17, 6 , true, false},
    {0, 0, "topbar19", 274, -3,  17, 6 , true, false},
}
```

Definition at line 330 of file keymap.cpp.

13.114.4.9 const Key KStrafeKeys[] [static]

Initial value:

```
= {
    {0, "Sidelight", "lsidel", 0,   76  /2,  12 ,   76  , true, false},
```

```
    {0, "Sidelight", "rsidel",  ( ( 298  -  38 )  + ( 12  * 2)) ,   76  /2,  12 ,   76  , true, false},
    {0, "Logo", "logo",  12 , 0,  12, 12 , true, false},
    {0, "Function", "fn", 152, 75,  12, 12 , true, true}
}
```

Definition at line 255 of file keymap.cpp.

13.114.4.10 const Key M65Keys[] [static]

Initial value:

```
= {
    {0, "Left Mouse", "mouse1", 8, 0, 14, 32, false, true}, {0, "Right Mouse", "mouse2", 30, 0, 14, 32,
      false, true}, {0, "Middle Mouse", "mouse3", 22, 8, 8, 7, false, true},
    {0, "Wheel Up", "wheelup", 22, 4, 8, 5, false, true}, {0, "Wheel Down", "wheeldn", 22, 14, 8, 5, false,
       true}, {0, "Wheel Light", "front", 22, 15, 8, 8, true, false},
    {0, "DPI Up", "dpiup", 22, 19, 8, 6, false, true}, {0, "DPI Light", "dpi", 22, 24, 8, 8, true, false},
      {0, "DPI Down", "dpidn", 22, 31, 8, 6, false, true},
    {0, "Forward", "mouse5", 5, 24, 5, 9, false, true}, {0, "Back", "mouse4", 5, 33, 5, 10, false, true}, {
      0, "Sniper", "sniper", 0, 25, 5, 15, false, true},
    {0, "Logo", "back", 14, 55, 24, 12, true, false}
}
```

Definition at line 263 of file keymap.cpp.

13.114.4.11 const KeyPatch patchDE[] [static]

Initial value:

```
= {
    {"caret", "^", "grave"}, {"ss", "ß", "minus"}, {"grave", "´", "equal"},
    {"z", "Z", "y"}, {"ue", "Ü", "lbrace"}, {"plus", "+", "rbrace"},
    {"oe", "Ö", "colon"}, {"ae", "Ä", "quote"},
    {"angle", "<", "bslash_iso"}, {"y", "Y", "z"}, {"minus", "-", "slash"}
}
```

Definition at line 57 of file keymap.cpp.

13.114.4.12 const KeyPatch patchDK[] [static]

Initial value:

```
= {
    {0, "§", "grave"}, {0, "+", "minus"}, {0, "´", "equal"},
    {0, "Å", "lbrace"}, {0, "¨", "rbrace"},
    {0, "Æ", "colon"}, {0, "Ø", "quote"}, {0, "’", "hash"},
    {0, "<", "bslash_iso"}, {0, "-", "slash"},
}
```

Definition at line 38 of file keymap.cpp.

13.114.4.13 const KeyPatch patchDvorak[] [static]

Initial value:

```
= {
    {0, "[", "minus"}, {0, "]", "equal"},
    {0, "’", "q"}, {0, ",", "w"}, {0, ".", "e"}, {0, "P", "r"}, {0, "Y", "t"}, {0, "F", "y"}, {0, "G", "u"}
      , {0, "C", "i"}, {0, "R", "o"}, {0, "L", "p"}, {0, "/", "lbrace"}, {0, "=", "rbrace"},
    {0, "O", "s"}, {0, "E", "d"}, {0, "U", "f"}, {0, "I", "g"}, {0, "D", "h"}, {0, "H", "j"}, {0, "T", "k"}
      , {0, "N", "l"}, {0, "S", "colon"}, {0, "-", "quote"},
    {0, ";", "z"}, {0, "Q", "x"}, {0, "J", "c"}, {0, "K", "v"}, {0, "X", "b"}, {0, "B", "n"}, {0, "W", "
      comma"}, {0, "V", "dot"}, {0, "Z", "slash"},
}
```

Definition at line 111 of file keymap.cpp.

13.114.4.14 const KeyPatch patchES[] [static]

Initial value:

```
= {
    {"oa", "°", "grave"}, {"quote", "’", "minus"}, {"lexclam", "¡", "equal"},
    {"grave", "`", "lbrace"}, {"plus", "+", "rbrace"},
    {"nn", "Ñ", "colon"}, {"accent", "´", "quote"}, {"cc", "Ç", "hash"},
    {"angle", "<", "bslash_iso"}, {"minus", "-", "slash"},
}
```

Definition at line 92 of file keymap.cpp.

13.114.4.15 const KeyPatch patchEU[] [static]

Initial value:

```
= {
    {0, "\\ (R)", "hash"},
    {0, "\\ (L)", "bslash_iso"},
}
```

Definition at line 45 of file keymap.cpp.

13.114.4.16 const KeyPatch patchFR[] [static]

Initial value:

```
= {
    {"sup2", "²", "grave"}, {0, "&", "1"}, {0, "É", "2"}, {0, "\"", "3"}, {0, "’", "4"}, {0, "(", "5"}, {0,
       "-", "6"}, {0, "È", "7"}, {0, "_", "8"}, {0, "Ç", "9"}, {0, "À", "0"}, {"rparen", ")", "minus"},
    {"a", "A", "q"}, {"z", "Z", "w"}, {"caret", "^", "lbrace"}, {"dollar", "$", "rbrace"},
    {"q", "Q", "a"}, {"m", "M", "colon"}, {"percent", "Ù", "quote"}, {"star", "*", "hash"},
    {"angle", "<", "bslash_iso"}, {"w", "W", "z"}, {"comma", ",", "m"}, {"semicolon", ";", "comma"}, {"
      colon", ":", "dot"}, {"exclam", "!", "slash"},
}
```

Definition at line 50 of file keymap.cpp.

13.114.4.17 const KeyPatch patchIT[] [static]

Initial value:

```
= {
    {0, "\\", "grave"}, {0, "’", "minus"}, {0, "ì", "equal"},
    {0, "è", "lbrace"}, {0, "+", "rbrace"},
    {0, "ò", "colon"}, {0, "à", "quote"}, {0, "ù", "hash"},
    {0, "<", "bslash_iso"}, {0, "-", "slash"},
}
```

Definition at line 64 of file keymap.cpp.

13.114.4.18 const KeyPatch patchJP[] [static]

Initial value:

```
= {

}
```

Definition at line 106 of file keymap.cpp.

13.114.4.19 const KeyPatch patchMX[] [static]

Initial value:

```
= {
    {0, "|", "grave"}, {0, "´", "minus"}, {0, "¿", "equal"},
    {0, "´", "lbrace"}, {0, "+", "rbrace"},
    {0, "Ñ", "colon"}, {0, "{", "quote"}, {0, "}", "hash"},
    {0, "<", "bslash_iso"}, {0, "-", "slash"},
}
```

Definition at line 78 of file keymap.cpp.

13.114.4.20 const KeyPatch patchNO[] [static]

Initial value:

```
= {
    {0, "§", "grave"}, {0, "+", "minus"}, {0, "´", "equal"},
    {0, "Å", "lbrace"}, {0, "¨", "rbrace"},
    {0, "Ø", "colon"}, {0, "Æ", "quote"}, {0, "'", "hash"},
    {0, "<", "bslash_iso"}, {0, "-", "slash"},
}
```

Definition at line 71 of file keymap.cpp.

13.114.4.21 const KeyPatch patchPTBR[] [static]

Initial value:

```
= {
    {"quote", "'", "grave"},
    {"accent", "´", "lbrace"}, {"lbrace", "[", "rbrace"},
    {"cc", "Ç", "colon"}, {"tilde", "~", "quote"}, {"rbrace", "]", "hash"},
    {"colon", ";", "slash"}, {"slash", "/", "ro"}, {"numcomma", ",", "numdot"},
}
```

Definition at line 85 of file keymap.cpp.

13.114.4.22 const KeyPatch patchSE[] [static]

Initial value:

```
= {
    {"section", "§", "grave"}, {"plus", "+", "minus"}, {"grave", "´", "equal"},
    {"aa", "Å", "lbrace"}, {"umlaut", "¨", "rbrace"},
    {"oe", "Ö", "colon"}, {"ae", "Ä", "quote"}, {"quote", "'", "hash"},
    {"angle", "<", "bslash_iso"}, {"minus", "-", "slash"},
}
```

Definition at line 99 of file keymap.cpp.

13.114.4.23 const Key SabreKeys[] [static]

Initial value:

```
= {
    {0, "Left Mouse", "mouse1", 8, 0, 14, 32, false, true}, {0, "Right Mouse", "mouse2", 30, 0, 14, 32,
      false, true}, {0, "Middle Mouse", "mouse3", 22, 9, 8, 7, false, true}, {0, "Front light", "front", 8, -2, 14, 8
      , true, false },
    {0, "Wheel Up", "wheelup", 22, 5, 8, 5, false, true}, {0, "Wheel Down", "wheeldn", 22, 15, 8, 5, false,
       true}, {0, "Wheel Light", "wheel", 22, 5, 8, 15, true, false}, {0, "Extra button", "thumb1", 22, 20, 8, 18,
       false, true},
    {0, "DPI Up", "dpiup", 5, 3, 5, 7, false, true}, {0, "DPI Down", "dpidn", 5, 10, 5, 7, false, true}, {0
```

```
      , "DPI Light", "dpi", 5, 4, 5, 12, true, false},
    {0, "Forward", "mouse5", 5, 24, 5, 9, false, true}, {0, "Back", "mouse4", 5, 33, 5, 10, false, true},
    {0, "Logo", "back", 14, 50, 24, 12, true, false}
}
```

Definition at line 276 of file keymap.cpp.

13.114.4.24 const Key ScimKeys[] [static]

Initial value:

```
= {
    {0, "Left Mouse", "mouse1", 8, 0, 14, 32, false, true}, {0, "Right Mouse", "mouse2", 30, 0, 12, 32,
      false, true}, {0, "Middle Mouse", "mouse3", 22, 9, 8, 6, false, true}, {0, "Front light", "front", 30, 0, 12, 8
      , true, false },
    {0, "Wheel Up", "wheelup", 22, 3, 8, 6, false, true}, {0, "Wheel Down", "wheeldn", 22, 14, 8, 6, false,
       true}, {0, "Wheel Light", "wheel", 22, 3, 8, 17, true, false},
    {0, "DPI Up", "dpiup", 22, 19, 8, 9, false, true}, {0, "DPI Light", "dpi", 1, 12, 8, 4, true, false}, {
      0, "DPI Down", "dpidn", 22, 28, 8, 9, false, true},
    {0, "Thumb light", "thumb", 0, 21, 10, 24, true, false},
    {0, "1", "thumb1", -13, 18, 7, 7, false, true}, {0, "2", "thumb2", -6, 18, 7, 7, false, true}, {0, "3",
       "thumb3", 1, 18, 7, 7, false, true},
    {0, "4", "thumb4", -13, 25, 7, 7, false, true}, {0, "5", "thumb5", -6, 25, 7, 7, false, true}, {0, "6",
       "thumb6", 1, 25, 7, 7, false, true},
    {0, "7", "thumb7", -13, 32, 7, 7, false, true}, {0, "8", "thumb8", -6, 32, 7, 7, false, true}, {0, "9",
       "thumb9", 1, 32, 7, 7, false, true},
    {0, "10", "thumb10", -13, 39, 7, 7, false, true}, {0, "11", "thumb11", -6, 39, 7, 7, false, true}, {0,
      "12", "thumb12", 1, 39, 7, 7, false, true},
    {0, "Logo", "back", 14, 50, 24, 16, true, false}
}
```

Definition at line 313 of file keymap.cpp.

13.114.4.25 QHash<QString, Key> standardMaps[KeyMap::_MODEL_MAX][KeyMap::_LAYOUT_MAX] [static]

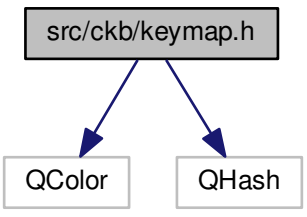
Definition at line 343 of file keymap.cpp.

Referenced by getMap().

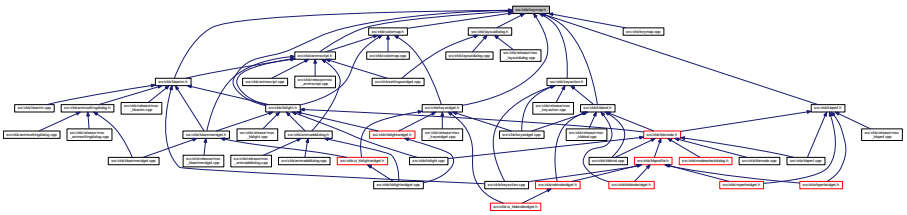
13.115 src/ckb/keymap.h File Reference

```
#include <QColor>
#include <QHash>
```

Include dependency graph for keymap.h:



This graph shows which files directly or indirectly include this file:



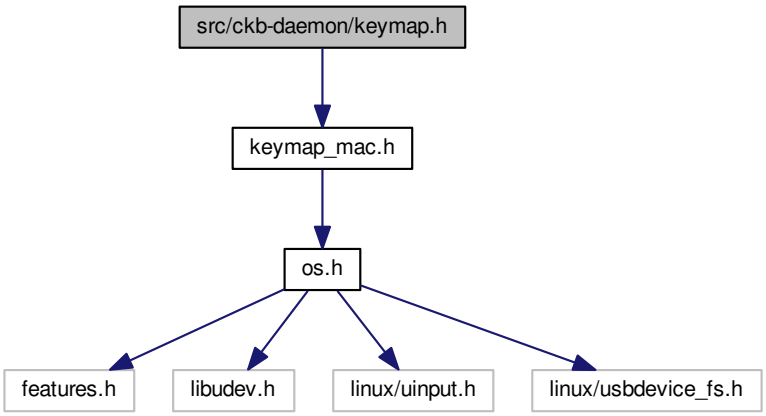
Data Structures

- struct Key
- class KeyMap

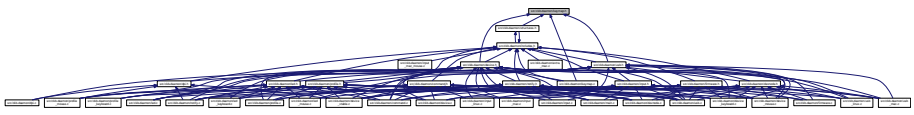
13.116 src/ckb-daemon/keymap.h File Reference

```
#include "keymap_mac.h"
```

Include dependency graph for keymap.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct key

Macros

- #define KEY_NONE -1
- #define KEY_CORSAIR -2
- #define KEY_UNBOUND -3
- #define BTN_WHEELUP 0x1f01
- #define BTN_WHEELDOWN 0x1f02
- #define KEY_BACKSLASH_ISO KEY_BACKSLASH
- #define N_KEYS_HW 152
- #define N_KEYBYTES_HW ((N_KEYS_HW + 7) / 8)
- #define N_KEY_ZONES 22
- #define N_KEYS_EXTRA 12
- #define N_BUTTONS_HW 20
- #define N_BUTTONS_EXTENDED 25
- #define MOUSE_BUTTON_FIRST (N_KEYS_HW + N_KEY_ZONES + N_KEYS_EXTRA)
- #define MOUSE_EXTRA_FIRST (MOUSE_BUTTON_FIRST + N_BUTTONS_HW)
- #define N_KEYS_INPUT (MOUSE_BUTTON_FIRST + N_BUTTONS_EXTENDED)
- #define N_KEYBYTES_INPUT ((N_KEYS_INPUT + 7) / 8)
- #define LED_MOUSE N_KEYS_HW
- #define N_MOUSE_ZONES 6
- #define N_MOUSE_ZONES_EXTENDED 12
- #define LED_DPI (LED_MOUSE + 2)
- #define N_KEYS_EXTENDED (N_KEYS_INPUT + N_MOUSE_ZONES_EXTENDED)
- #define N_KEYBYTES_EXTENDED ((N_KEYS_EXTENDED + 7) / 8)
- #define SCAN_SILENT 0x8000
- #define SCAN_KBD 0
- #define SCAN_MOUSE 0x1000

Functions

- void hid_kb_translate (unsigned char *kbinput, int endpoint, int length, const unsigned char *urbinput)
- void hid_mouse_translate (unsigned char *kbinput, short *xaxis, short *yaxis, int endpoint, int length, const unsigned char *urbinput, ushort fwversion)
- void corsair_kbcopy (unsigned char *kbinput, int endpoint, const unsigned char *urbinput)
- void corsair_mousecopy (unsigned char *kbinput, int endpoint, const unsigned char *urbinput)

Variables

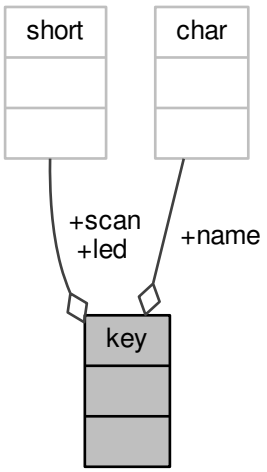
- const key keymap [(((152+22+12)+25)+12)]

13.116.1 Data Structure Documentation

13.116.1.1 struct key

Definition at line 49 of file keymap.h.

Collaboration diagram for key:



Data Fields

short	led	
const char *	name	
short	scan	

13.116.2 Macro Definition Documentation

13.116.2.1 #define BTN_WHEELDOWN 0x1f02

Definition at line 13 of file keymap.h.

Referenced by os_keypress().

13.116.2.2 #define BTN_WHEELUP 0x1f01

Definition at line 12 of file keymap.h.

Referenced by os_keypress().

13.116.2.3 #define KEY_BACKSLASH_ISO KEY_BACKSLASH

Definition at line 20 of file keymap.h.

13.116.2.4 #define KEY_CORSAIR -2

Definition at line 8 of file keymap.h.

13.116.2.5 #define KEY_NONE -1

Definition at line 7 of file keymap.h.

13.116.2.6 #define KEY_UNBOUND -3

Definition at line 9 of file keymap.h.

Referenced by cmd_unbind().

13.116.2.7 #define LED_DPI (LED_MOUSE + 2)

Definition at line 43 of file keymap.h.

Referenced by loadrgb_mouse(), and savergb_mouse().

13.116.2.8 #define LED_MOUSE N_KEYS_HW

Definition at line 39 of file keymap.h.

Referenced by isblack(), loaddpi(), loadrgb_mouse(), rgbcmp(), savedpi(), savergb_mouse(), and updatergb_mouse().

13.116.2.9 #define MOUSE_BUTTON_FIRST (N_KEYS_HW + N_KEY_ZONES + N_KEYS_EXTRA)

Definition at line 33 of file keymap.h.

Referenced by corsair_mousecopy(), and hid_mouse_translate().

13.116.2.10 #define MOUSE_EXTRA_FIRST (MOUSE_BUTTON_FIRST + N_BUTTONS_HW)

Definition at line 34 of file keymap.h.

Referenced by hid_mouse_translate().

13.116.2.11 #define N_BUTTONS_EXTENDED 25

Definition at line 32 of file keymap.h.

13.116.2.12 #define N_BUTTONS_HW 20

Definition at line 31 of file keymap.h.

Referenced by corsair_mousecopy().

13.116.2.13 #define N_KEY_ZONES 22

Definition at line 27 of file keymap.h.

13.116.2.14 #define N_KEYBYTES_EXTENDED ((N_KEYS_EXTENDED + 7) / 8)

Definition at line 46 of file keymap.h.

13.116.2.15 #define N_KEYBYTES_HW ((N_KEYS_HW + 7) / 8)

Definition at line 25 of file keymap.h.

Referenced by corsair_kbcopy().

13.116.2.16 #define N_KEYBYTES_INPUT ((N_KEYS_INPUT + 7) / 8)

Definition at line 37 of file keymap.h.

Referenced by _cmd_macro(), inputupdate_keys(), and macromask().

13.116.2.17 #define N_KEYS_EXTENDED (N_KEYS_INPUT + N_MOUSE_ZONES_EXTENDED)

Definition at line 45 of file keymap.h.

Referenced by printrgb(), and readcmd().

13.116.2.18 #define N_KEYS_EXTRA 12

Definition at line 29 of file keymap.h.

13.116.2.19 #define N_KEYS_HW 152

Definition at line 24 of file keymap.h.

Referenced by loadrgb_kb(), makergb_512(), rgbcmp(), and setactive_kb().

13.116.2.20 #define N_KEYS_INPUT (MOUSE_BUTTON_FIRST + N_BUTTONS_EXTENDED)

Definition at line 36 of file keymap.h.

Referenced by _cmd_get(), _cmd_macro(), cmd_bind(), cmd_notify(), cmd_rebind(), cmd_unbind(), initbind(), and inputupdate_keys().

13.116.2.21 #define N_MOUSE_ZONES 6

Definition at line 40 of file keymap.h.

Referenced by isblack(), loaddpi(), rgbcmp(), savedpi(), and updatergb_mouse().

13.116.2.22 #define N_MOUSE_ZONES_EXTENDED 12

Definition at line 41 of file keymap.h.

13.116.2.23 #define SCAN_KBD 0

Definition at line 57 of file keymap.h.

13.116.2.24 #define SCAN_MOUSE 0x1000

Definition at line 58 of file keymap.h.

Referenced by os_keypress().

13.116.2.25 #define SCAN_SILENT 0x8000

Definition at line 56 of file keymap.h.

Referenced by inputupdate_keys().

13.116.3 Function Documentation

13.116.3.1 void corsair_kbcopy (unsigned char * *kbinput,* int *endpoint,* const unsigned char * *urbinput*)

Definition at line 420 of file keymap.c.

References N_KEYBYTES_HW.

Referenced by os_inputmain().

```
420                                                                                                 {
421     if(endpoint == 2 || endpoint == -2){
422         if(urbinput[0] != 3)
423             return;
424         urbinput++;
425     }
426     memcpy(kbinput, urbinput, N_KEYBYTES_HW);
427 }
```

Here is the caller graph for this function:

13.116.3.2 void corsair_mousecopy (unsigned char * *kbinput,* int *endpoint,* const unsigned char * *urbinput*)

Definition at line 429 of file keymap.c.

References BUTTON_HID_COUNT, CLEAR_KEYBIT, MOUSE_BUTTON_FIRST, N_BUTTONS_HW, and SET_KEYBIT.

Referenced by os_inputmain().

```
429                                                                                                 {
430     if(endpoint == 2 || endpoint == -2){
431         if(urbinput[0] != 3)
432             return;
433         urbinput++;
434     }
435     for(int bit = BUTTON_HID_COUNT; bit < N_BUTTONS_HW; bit++){
436         int byte = bit / 8;
437         uchar test = 1 << (bit % 8);
438         if(urbinput[byte] & test)
439             SET_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
440         else
441             CLEAR_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
442     }
443 }
```

Here is the caller graph for this function:

13.116.3.3 void hid_kb_translate (unsigned char ∗ *kbinput,* int *endpoint,* int *length,* const unsigned char ∗ *urbinput*)

Definition at line 246 of file keymap.c.

References ckb_warn, CLEAR_KEYBIT, and SET_KEYBIT.

Referenced by os_inputmain().

```
246 {
247     if(length < 1)
248         return;
249     // LUT for HID -> Corsair scancodes (-1 for no scan code, -2 for currently unsupported)
250     // Modified from Linux drivers/hid/usbhid/usbkbd.c, key codes replaced with keymap array indices and K95 keys added
251     // Make sure the indices match the keyindex as passed to nprintkey() in notify.c
252     static const short hid_codes[256] = {
253         -1,  -1,  -1,  -1,  37,  54,  52,  39,  27,  40,  41,  42,  32,  43,  44,  45,
254         56,  55,  33,  34,  25,  28,  38,  29,  31,  53,  26,  51,  30,  50,  13,  14,
255         15,  16,  17,  18,  19,  20,  21,  22,  82,   0,  86,  24,  64,  23,  84,  35,
256         79,  80,  81,  46,  47,  12,  57,  58,  59,  36,   1,   2,   3,   4,   5,   6,
257          7,   8,   9,  10,  11,  72,  73,  74,  75,  76,  77,  78,  87,  88,  89,  95,
258         93,  94,  92, 102, 103, 104, 105, 106, 107, 115, 116, 117, 112, 113, 114, 108,
259        109, 110, 118, 119,  49,  69,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,
260         -2,  -2,  -2,  -2,  -2,  -2,  -2,  -2,  98,  -2,  -2,  -2,  -2,  -2,  -2,  97,
261        130, 131,  -1,  -1,  -1,  -2,  -1,  83,  66,  85, 145, 144,  -2,  -1,  -1,  -1,
262         -2,  -2,  -2,  -2,  -2,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
263         -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
264         -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,
265         -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -3,  -1,  -1,  -1,  // <- -3 = non-RGB program key
266        120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 136, 137, 138, 139, 140, 141,
267         60,  48,  62,  61,  91,  90,  67,  68, 142, 143,  99, 101,  -2, 130, 131,  97,
268         -2, 133, 134, 135,  -2,  96,  -2, 132,  -2,  -2,  71,  71,  71,  71,  -1,  -1,
269     };
270     switch(endpoint){
271     case 1:
272     case -1:
273         // EP 1: 6KRO input (RGB and non-RGB)
274         // Clear previous input
275         for(int i = 0; i < 256; i++){
276             if(hid_codes[i] >= 0)
277                 CLEAR_KEYBIT(kbinput, hid_codes[i]);
278         }
279         // Set new input
280         for(int i = 0; i < 8; i++){
281             if((urbinput[0] >> i) & 1)
282                 SET_KEYBIT(kbinput, hid_codes[i + 224]);
283         }
284         for(int i = 2; i < length; i++){
285             if(urbinput[i] > 3){
286                 int scan = hid_codes[urbinput[i]];
287                 if(scan >= 0)
288                     SET_KEYBIT(kbinput, scan);
289                 else
290                     ckb_warn("Got unknown key press %d on EP 1\n", urbinput[i]);
291             }
292         }
293         break;
294     case -2:
295         // EP 2 RGB: NKRO input
296         if(urbinput[0] == 1){
297             // Type 1: standard key
298             if(length != 21)
299                 return;
300             for(int bit = 0; bit < 8; bit++){
301                 if((urbinput[1] >> bit) & 1)
302                     SET_KEYBIT(kbinput, hid_codes[bit + 224]);
303                 else
304                     CLEAR_KEYBIT(kbinput, hid_codes[bit + 224]);
305             }
306             for(int byte = 0; byte < 19; byte++){
307                 char input = urbinput[byte + 2];
308                 for(int bit = 0; bit < 8; bit++){
309                     int keybit = byte * 8 + bit;
310                     int scan = hid_codes[keybit];
311                     if((input >> bit) & 1){
312                         if(scan >= 0)
313                             SET_KEYBIT(kbinput, hid_codes[keybit]);
314                         else
315                             ckb_warn("Got unknown key press %d on EP 2\n", keybit);
316                     } else if(scan >= 0)
317                         CLEAR_KEYBIT(kbinput, hid_codes[keybit]);
318                 }
319             }
320             break;
```

```
321         } else if (urbinput[0] == 2)
322             ; // Type 2: media key (implicitly falls through)
323         else
324             break;  // No other known types
325         /* FALLTHRU */
326     case 2:
327         // EP 2 Non-RGB: media keys
328         CLEAR_KEYBIT(kbinput, 97);          // mute
329         CLEAR_KEYBIT(kbinput, 98);          // stop
330         CLEAR_KEYBIT(kbinput, 99);          // prev
331         CLEAR_KEYBIT(kbinput, 100);         // play
332         CLEAR_KEYBIT(kbinput, 101);         // next
333         CLEAR_KEYBIT(kbinput, 130);         // volup
334         CLEAR_KEYBIT(kbinput, 131);         // voldn
335         for(int i = 0; i < length; i++){
336             switch(urbinput[i]){
337             case 181:
338                 SET_KEYBIT(kbinput, 101);   // next
339                 break;
340             case 182:
341                 SET_KEYBIT(kbinput, 99);    // prev
342                 break;
343             case 183:
344                 SET_KEYBIT(kbinput, 98);    // stop
345                 break;
346             case 205:
347                 SET_KEYBIT(kbinput, 100);   // play
348                 break;
349             case 226:
350                 SET_KEYBIT(kbinput, 97);    // mute
351                 break;
352             case 233:
353                 SET_KEYBIT(kbinput, 130);   // volup
354                 break;
355             case 234:
356                 SET_KEYBIT(kbinput, 131);   // voldn
357                 break;
358             }
359         }
360         break;
361     case 3:
362         // EP 3 non-RGB: NKRO input
363         if(length != 15)
364             return;
365         for(int bit = 0; bit < 8; bit++){
366             if((urbinput[0] >> bit) & 1)
367                 SET_KEYBIT(kbinput, hid_codes[bit + 224]);
368             else
369                 CLEAR_KEYBIT(kbinput, hid_codes[bit + 224]);
370         }
371         for(int byte = 0; byte < 14; byte++){
372             char input = urbinput[byte + 1];
373             for(int bit = 0; bit < 8; bit++){
374                 int keybit = byte * 8 + bit;
375                 int scan = hid_codes[keybit];
376                 if((input >> bit) & 1){
377                     if(scan >= 0)
378                         SET_KEYBIT(kbinput, hid_codes[keybit]);
379                     else
380                         ckb_warn("Got unknown key press %d on EP 3\n", keybit);
381                 } else if(scan >= 0)
382                     CLEAR_KEYBIT(kbinput, hid_codes[keybit]);
383             }
384         }
385         break;
386     }
387 }
```

Here is the caller graph for this function:

13.116.3.4 void hid_mouse_translate (unsigned char * *kbinput*, short * *xaxis*, short * *yaxis*, int *endpoint*, int *length*, const unsigned char * *urbinput*, ushort *fwversion*)

Definition at line 391 of file keymap.c.

References BUTTON_HID_COUNT, CLEAR_KEYBIT, MOUSE_BUTTON_FIRST, MOUSE_EXTRA_FIRST, and S-ET_KEYBIT.

Referenced by os_inputmain().

```
391
                                                {
392     //The HID Input Endpoint on FWv3 is 64 bytes, so we can't check for length.
393     if((endpoint != 2 && endpoint != -2) || (fwversion < 0x300 && length < 10))
394         return;
395     // EP 2: mouse input
396     if(urbinput[0] != 1)
397         return;
398     // Byte 1 = mouse buttons (bitfield)
399     for(int bit = 0; bit < BUTTON_HID_COUNT; bit++){
400         if(urbinput[1] & (1 << bit))
401             SET_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
402         else
403             CLEAR_KEYBIT(kbinput, MOUSE_BUTTON_FIRST + bit);
404     }
405     // Bytes 5 - 8: movement
406     *xaxis += *(short*)(urbinput + 5);
407     *yaxis += *(short*)(urbinput + 7);
408     // Byte 9: wheel
409     char wheel = urbinput[9];
410     if(wheel > 0)
411         SET_KEYBIT(kbinput, MOUSE_EXTRA_FIRST);         // wheelup
412     else
413         CLEAR_KEYBIT(kbinput, MOUSE_EXTRA_FIRST);
414     if(wheel < 0)
415         SET_KEYBIT(kbinput, MOUSE_EXTRA_FIRST + 1);     // wheeldn
416     else
417         CLEAR_KEYBIT(kbinput, MOUSE_EXTRA_FIRST + 1);
418 }
```

Here is the caller graph for this function:

13.116.4 Variable Documentation

13.116.4.1 const key keymap[(((152+22+12)+25)+12)]

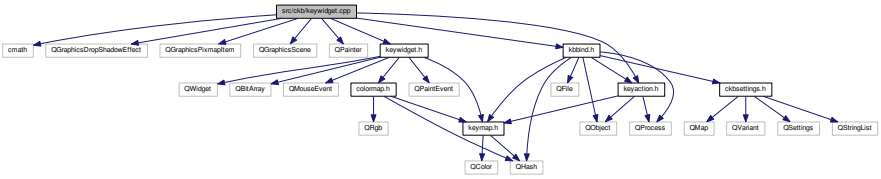
Definition at line 5 of file keymap.c.

Referenced by _cmd_get(), _cmd_macro(), cmd_bind(), cmd_rebind(), cmd_rgb(), initbind(), inputupdate_keys(), nprintkey(), printrgb(), readcmd(), and setactive_kb().

13.117 src/ckb/keywidget.cpp File Reference

```
#include <cmath>
#include <QGraphicsDropShadowEffect>
#include <QGraphicsPixmapItem>
#include <QGraphicsScene>
#include <QPainter>
#include "keywidget.h"
#include "keyaction.h"
#include "kbbind.h"
```

Include dependency graph for keywidget.cpp:



Functions

- QRgb monoRgb (float r, float g, float b)

Variables

- static const int KEY_SIZE = 12
- static QImage * m65Overlay = 0
- static QImage * sabOverlay = 0
- static QImage * scimOverlay = 0
- static QImage * harpOverlay = 0
- static QImage * glaiveOverlay = 0

13.117.1 Function Documentation

13.117.1.1 QRgb monoRgb (float *r*, float *g*, float *b*)

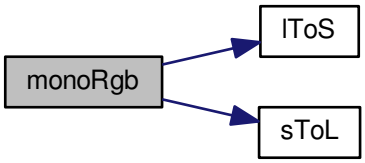
Definition at line 284 of file kblight.cpp.

References lToS(), and sToL().

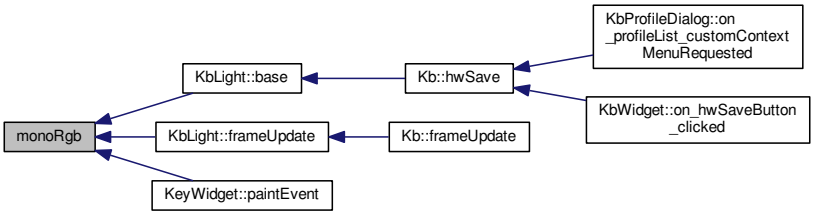
Referenced by KbLight::base(), KbLight::frameUpdate(), and KeyWidget::paintEvent().

```
284                                     {
285     // It's important to use a linear colorspace for this, otherwise the colors will appear inconsistent
286     // Note that although we could use linear space for alpha blending or the animation blending functions,
        we don't.
287     // The reason for this is that photo manipulation programs don't do it either, so even though the
        result would technically be more correct,
288     // it would look wrong to most people.
289     r = sToL(r);
290     g = sToL(g);
291     b = sToL(b);
292     int value = round(lToS((r + g + b) / 3.f));
293     return qRgb(value, value, value);
294 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.117.2 Variable Documentation

13.117.2.1 QImage * glaiveOverlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

13.117.2.2 QImage * harpOverlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

13.117.2.3 const int KEY_SIZE = 12 [static]

Definition at line 10 of file keywidget.cpp.

Referenced by KeyWidget::drawInfo(), and KeyWidget::map().

13.117.2.4 QImage* m65Overlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

13.117.2.5 QImage * sabOverlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

13.117.2.6 QImage * scimOverlay = 0 [static]

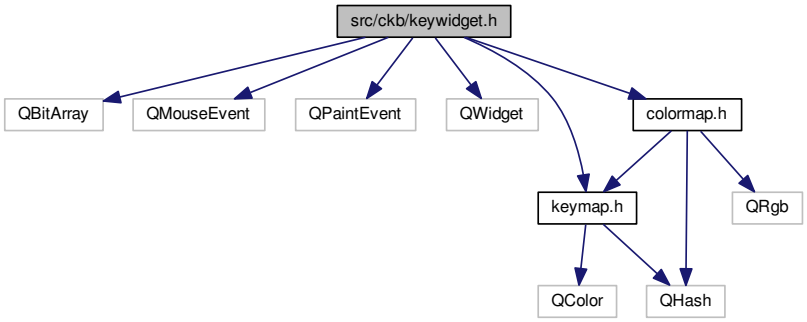
Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

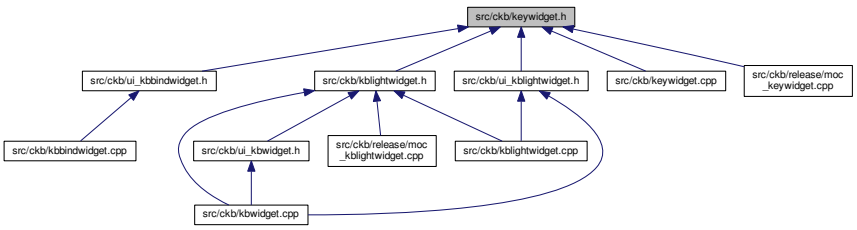
13.118 src/ckb/keywidget.h File Reference

```
#include <QBitArray>
#include <QMouseEvent>
#include <QPaintEvent>
#include <QWidget>
#include "keymap.h"
#include "colormap.h"
```

Include dependency graph for keywidget.h:



This graph shows which files directly or indirectly include this file:



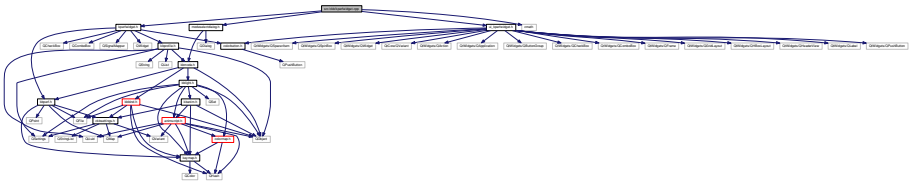
Data Structures

- class KeyWidget

13.119 src/ckb/kperfwidget.cpp File Reference

```
#include "kperfwidget.h"
#include "ui_kperfwidget.h"
#include "modeselectdialog.h"
#include <cmath>
```

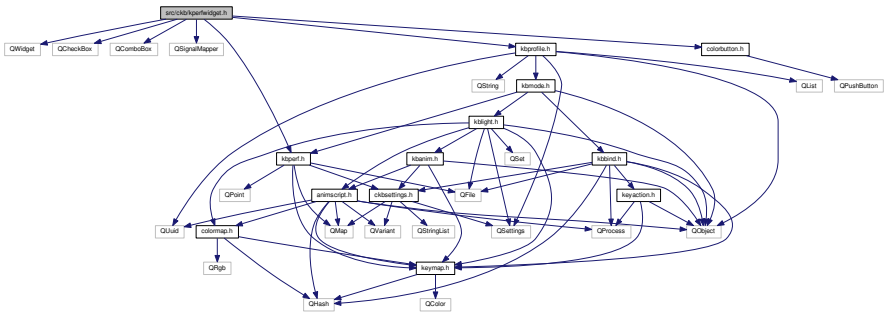
Include dependency graph for kperfwidget.cpp:



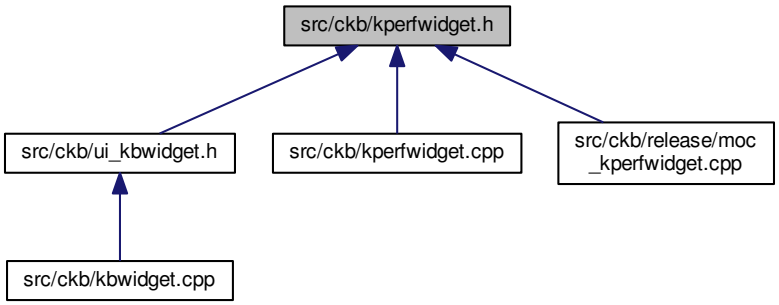
13.120 src/ckb/kperfwidget.h File Reference

```
#include <QWidget>
#include <QCheckBox>
#include <QComboBox>
#include <QSignalMapper>
#include "kbperf.h"
#include "kbprofile.h"
#include "colorbutton.h"
```

Include dependency graph for kperfwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class KPerfWidget
- struct KPerfWidget::IndicatorUi

Namespaces

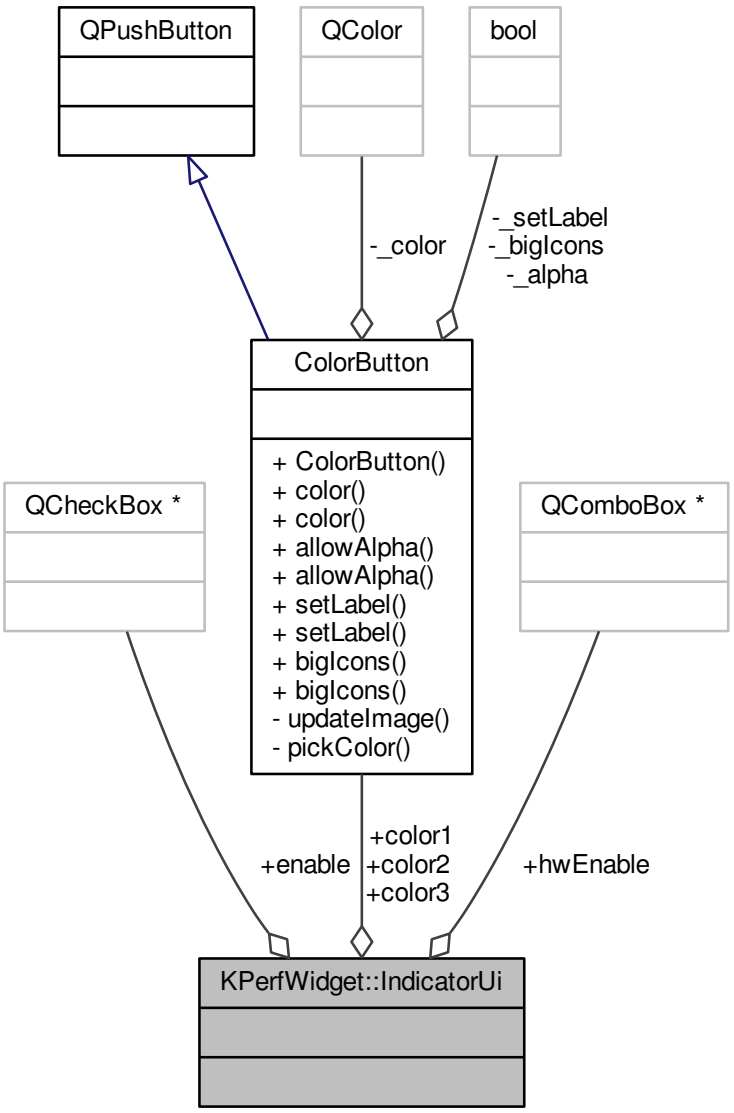
- Ui

13.120.1 Data Structure Documentation

13.120.1.1 struct KPerfWidget::IndicatorUi

Definition at line 54 of file kperfwidget.h.

Collaboration diagram for KPerfWidget::IndicatorUi:



Data Fields

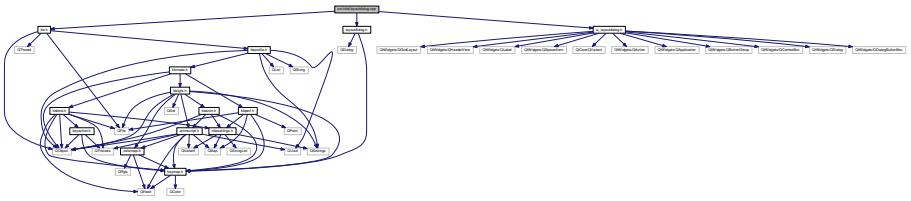
ColorButton *	color1	
ColorButton *	color2	
ColorButton *	color3	
QCheckBox *	enable	
QComboBox *	hwEnable	

13.121 src/ckb/layoutdialog.cpp File Reference

```
#include "layoutdialog.h"
```

```
#include "kb.h"
#include "ui_layoutdialog.h"
```

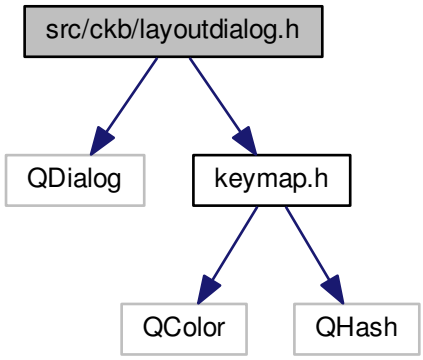
Include dependency graph for layoutdialog.cpp:



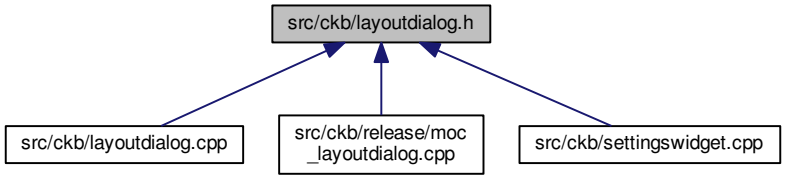
13.122 src/ckb/layoutdialog.h File Reference

```
#include <QDialog>
#include "keymap.h"
```

Include dependency graph for layoutdialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class LayoutDialog

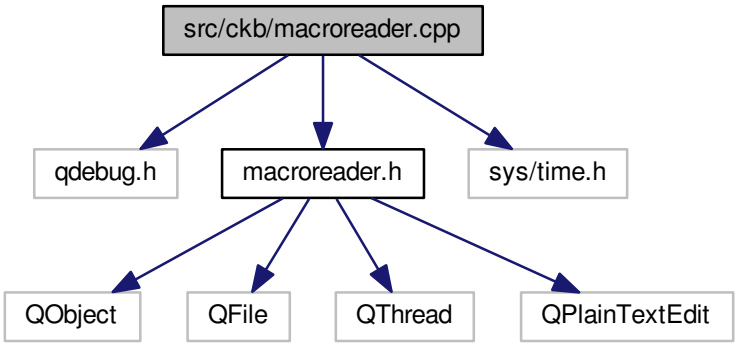
Namespaces

- Ui

13.123 src/ckb/macroreader.cpp File Reference

```
#include <qdebug.h>
#include "macroreader.h"
#include <sys/time.h>
```

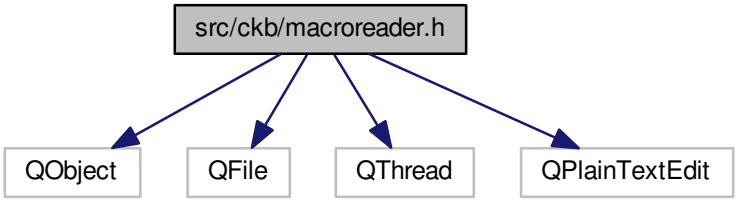
Include dependency graph for macroreader.cpp:



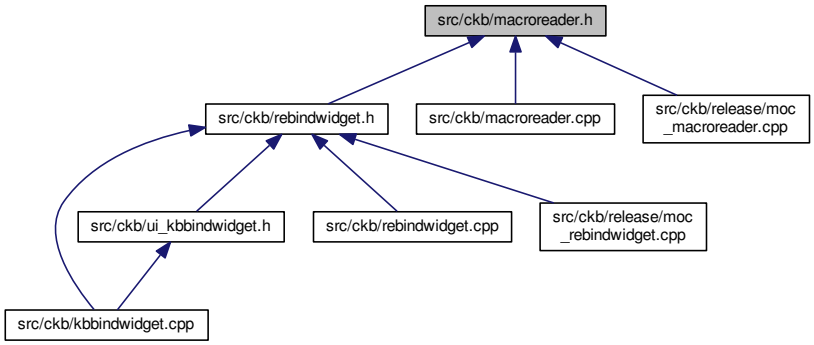
13.124 src/ckb/macroreader.h File Reference

```
#include <QObject>
#include <QFile>
#include <QThread>
#include <QPlainTextEdit>
```

Include dependency graph for macroreader.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class MacroReaderThread

  The MacroReaderThread class is responsible for reading Macro Key Values. It is created as a separate thread (worker thread) for reading macro commands from an fresh opened notify channel. Standard notify channel for macro definitions is number 2.

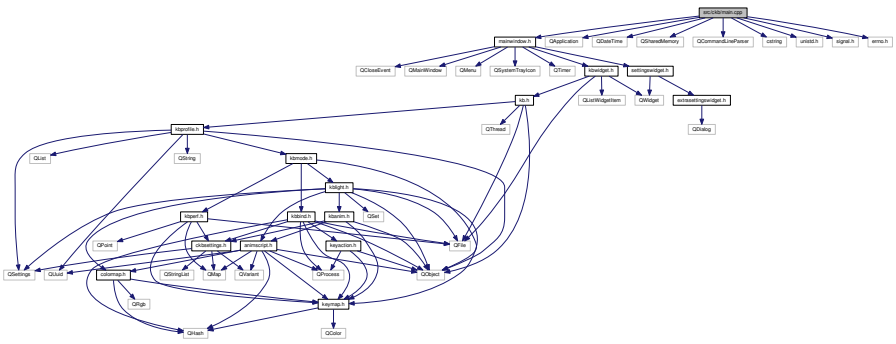
- class MacroReader

  The MacroReader class creates a worker thread object. It does a connect do delayed deletion of thread local variables in the case the worker thread terminates.

13.125 src/ckb/main.cpp File Reference

```
#include "mainwindow.h"
#include <QApplication>
#include <QDateTime>
#include <QSharedMemory>
#include <QCommandLineParser>
#include <cstring>
#include <unistd.h>
#include <signal.h>
#include <errno.h>
```

Include dependency graph for main.cpp:



Macros

- #define SHMEM_SIZE 128 ∗ 1024
- #define SHMEM_SIZE_V015 16

Enumerations

- enum CommandLineParseResults {
  CommandLineOK, CommandLineError, CommandLineVersionRequested, CommandLineHelpRequested,
  CommandLineClose, CommandLineBackground }

Functions

- QSharedMemory appShare ("ckb")
- CommandLineParseResults parseCommandLine (QCommandLineParser &parser, QString ∗errorMessage)

  parseCommandLine - Setup options and parse command line arguments.
- static bool pidActive (const QStringList &lines)
- static bool isRunning (const char ∗command)
- int main (int argc, char ∗argv[])

13.125.1 Macro Definition Documentation

13.125.1.1 #define SHMEM_SIZE 128 ∗ 1024

Definition at line 20 of file main.cpp.

Referenced by isRunning().

13.125.1.2 #define SHMEM_SIZE_V015 16

Definition at line 21 of file main.cpp.

Referenced by isRunning().

13.125.2 Enumeration Type Documentation

13.125.2.1 enum CommandLineParseResults

Enumerator

CommandLineOK

CommandLineError

CommandLineVersionRequested

CommandLineHelpRequested

CommandLineClose

CommandLineBackground

Definition at line 24 of file main.cpp.

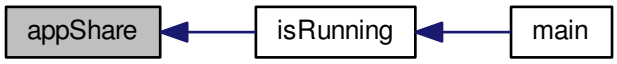
```
24                              {
25     CommandLineOK,
26     CommandLineError,
27     CommandLineVersionRequested,
28     CommandLineHelpRequested,
29     CommandLineClose,
30     CommandLineBackground
31 };
```

13.125.3 Function Documentation

13.125.3.1 QSharedMemory appShare ("ckb")

Referenced by isRunning().

Here is the caller graph for this function:



13.125.3.2 static bool isRunning (const char * *command*) [static]

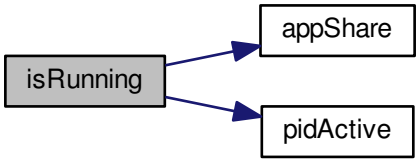
Definition at line 109 of file main.cpp.

References appShare(), pidActive(), SHMEM_SIZE, and SHMEM_SIZE_V015.

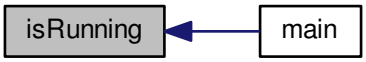
Referenced by main().

```
109                                         {
110     // Try to create shared memory (if successful, application was not already running)
111     if(!appShare.create(SHMEM_SIZE) || !appShare.lock()){
112         // Lock existing shared memory
113         if(!appShare.attach() || !appShare.lock())
114             return true;
115         bool running = false;
116         if(appShare.size() == SHMEM_SIZE_V015){
117             // Old shmem - no PID listed so assume the process is running, and print the command directly
        to the buffer
118             if(command){
119                 void* data = appShare.data();
120                 snprintf((char*)data, SHMEM_SIZE_V015, "%s", command);
121             }
122             running = true;
123         } else {
124             // New shmem. Scan the contents of the shared memory for a PID
125             QStringList lines = QString((const char*)appShare.constData()).split("\n");
126             if(pidActive(lines)){
127                 running = true;
128                 // Append the command
129                 if(command){
130                     lines.append(QString("Option ") + command);
131                     QByteArray newMem = lines.join("\n").left(SHMEM_SIZE).toLatin1();
132                     // Copy the NUL byte as well as the string
133                     memcpy(appShare.data(), newMem.constData(), newMem.length() + 1);
134                 }
135             }
136         }
137         if(running){
138             appShare.unlock();
139             return true;
140         }
141     }
142     // Not already running. Initialize the shared memory with our PID
143     snprintf((char*)appShare.data(), appShare.size(), "PID %ld", (long)getpid());
144     appShare.unlock();
145     return false;
146 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.125.3.3 int main (int *argc*, char * *argv[]*)

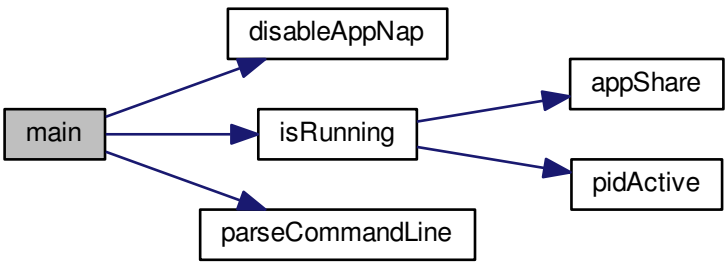
Definition at line 148 of file main.cpp.

References CommandLineBackground, CommandLineClose, CommandLineError, CommandLineHelpRequested, CommandLineOK, CommandLineVersionRequested, disableAppNap(), isRunning(), and parseCommandLine().

```
148                                {
149     // Setup main application
150     QApplication a(argc, argv);
151 
152     // Setup names and versions
153     QCoreApplication::setOrganizationName("ckb");
154     QCoreApplication::setApplicationVersion(CKB_VERSION_STR);
155     QCoreApplication::setApplicationName("ckb");
156 
157     // Setup argument parser
158     QCommandLineParser parser;
159     QString errorMessage;
160     parser.setApplicationDescription("Open Source Corsair Input Device Driver for Linux and OSX.");
161     bool background = 0;
162 
163     // Although the daemon runs as root, the GUI needn't and shouldn't be, as it has the potential to
      corrupt settings data.
164     if(getuid() == 0){
165         printf("The ckb GUI should not be run as root.\n");
166         return 0;
167     }
168 
169     // Seed the RNG for UsbIds
170     qsrand(QDateTime::currentMSecsSinceEpoch());
171 #ifdef Q_OS_MACX
172     disableAppNap();
173 
174     FILE *fp = fopen("/tmp/ckb", "w");
175     fprintf(fp, "%d", getpid());
176     fclose(fp);
177 #endif
178 #if QT_VERSION >= QT_VERSION_CHECK(5, 3, 0)
```

```
179     // Enable HiDPI support
180     qApp->setAttribute(Qt::AA_UseHighDpiPixmaps);
181 #endif
182
183     // Parse arguments
184     switch (parseCommandLine(parser, &errorMessage)) {
185     case CommandLineOK:
186         // If launched with no argument
187         break;
188     case CommandLineError:
189         fputs(qPrintable(errorMessage), stderr);
190         fputs("\n\n", stderr);
191         fputs(qPrintable(parser.helpText()), stderr);
192         return 1;
193     case CommandLineVersionRequested:
194         // If launched with --version, print version info and then quit
195         printf("%s %s\n", qPrintable(QCoreApplication::applicationName()),
196                 qPrintable(QCoreApplication::applicationVersion()));
197         return 0;
198     case CommandLineHelpRequested:
199         // If launched with --help, print help and then quit
200         parser.showHelp();
201         return 0;
202     case CommandLineClose:
203         // If launched with --close, kill existing app
204         if (isRunning("Close"))
205             printf("Asking existing instance to close.\n");
206         else
207             printf("ckb is not running.\n");
208         return 0;
209     case CommandLineBackground:
210         // If launched with --background, launch in background
211         background = 1;
212         break;
213     }
214
215     // Launch in background if requested, or if re-launching a previous session
216     if(qApp->isSessionRestored())
217             background = 1;
218     if(isRunning(background ? 0 : "Open")){
219         printf("ckb is already running. Exiting.\n");
220         return 0;
221     }
222     MainWindow w;
223     if(!background)
224         w.show();
225
226     return a.exec();
227 }
```

Here is the call graph for this function:



13.125.3.4 CommandLineParseResults parseCommandLine (QCommandLineParser & *parser*, QString * *errorMessage*)

Parameters

parser	parser instance to use for parse the arguments
errorMessage	argument parse error message

Returns

integer, representing the requested argument

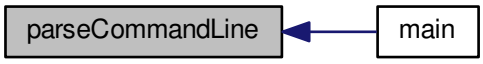
Definition at line 41 of file main.cpp.

References CommandLineBackground, CommandLineClose, CommandLineError, CommandLineHelpRequested, CommandLineOK, and CommandLineVersionRequested.

Referenced by main().

```
41                                                                                                  {
42     // setup parser to interpret -abc as -a -b -c
43     parser.setSingleDashWordOptionMode(QCommandLineParser::ParseAsCompactedShortOptions);
44
45     /* add command line options */
46     // add -v, --version
47     const QCommandLineOption versionOption = parser.addVersionOption();
48     // add -h, --help (/? on windows)
49     const QCommandLineOption helpOption = parser.addHelpOption();
50     // add -b, --background
51     const QCommandLineOption backgroundOption(QStringList() << "b" << "background",
52                                               "Starts in background, without displaying the main window.");
53     parser.addOption(backgroundOption);
54     // add -c, --close
55     const QCommandLineOption closeOption(QStringList() << "c" << "close",
56                                          "Causes already running instance (if any) to exit.");
57     parser.addOption(closeOption);
58
59     /* parse arguments */
60     if (!parser.parse(QCoreApplication::arguments())) {
61         // set error, if there already are some
62         *errorMessage = parser.errorText();
63         return CommandLineError;
64     }
65
66     /* return requested operation/setup */
67     if (parser.isSet(versionOption)) {
68         // print version and exit
69         return CommandLineVersionRequested;
70     }
71
72     if (parser.isSet(helpOption)) {
73         // print help and exit
74         return CommandLineHelpRequested;
75     }
76
77     if (parser.isSet(backgroundOption)) {
78         // open application in background
79         return CommandLineBackground;
80     }
81
82     if (parser.isSet(closeOption)) {
83         // close already running application instances, if any
84         return CommandLineClose;
85     }
86
87     /* no explicit argument was passed */
88     return CommandLineOK;
89 };
```

Here is the caller graph for this function:



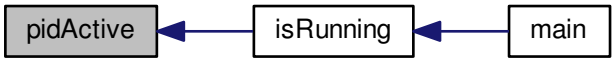
13.125.3.5 static bool pidActive (const QStringList & *lines*) [static]

Definition at line 92 of file main.cpp.

Referenced by isRunning().

```
92                                                {
93     foreach(const QString& line, lines){
94         if(line.startsWith("PID ")){
95             bool ok;
96             pid_t pid;
97             // Valid PID found?
98             if((pid = line.split(" ")[1].toLong(&ok)) > 0 && ok){
99                 // kill -0 does nothing to the application, but checks if it's running
100                 return (kill(pid, 0) == 0 || errno != ESRCH);
101             }
102         }
103     }
104     // If the PID wasn't listed in the shmem, assume it is running
105     return true;
106 }
```

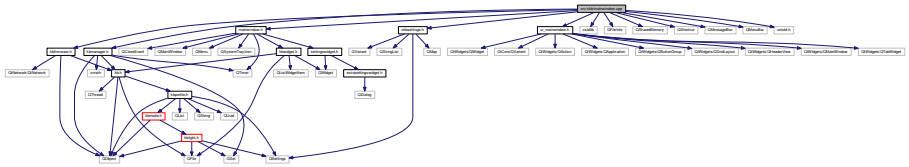
Here is the caller graph for this function:



13.126 src/ckb/mainwindow.cpp File Reference

```
#include "ckbsettings.h"
#include "kbmanager.h"
#include "kbfirmware.h"
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <cstdlib>
#include <QFileInfo>
#include <QSharedMemory>
#include <QShortcut>
#include <QMessageBox>
#include <QMenuBar>
#include <unistd.h>
```

Include dependency graph for mainwindow.cpp:



Variables

- QSharedMemory appShare
- QString devpath

  Device path base ("/dev/input/ckb" or "/var/run/ckb")
- static const QString configLabel = "Settings"
- QString daemonDialogText

13.126.1 Variable Documentation

13.126.1.1 QSharedMemory appShare

Referenced by MainWindow::timerTick().

13.126.1.2 const QString configLabel = "Settings" [static]

Definition at line 17 of file mainwindow.cpp.

Referenced by MainWindow::MainWindow().

13.126.1.3 QString daemonDialogText

Initial value:

```
= QObject::tr("Start it once with:") +
    "<blockquote><code>sudo systemctl start ckb-daemon</code></blockquote>" +
    QObject::tr("Enable it for every boot:") +
    "<blockquote><code>sudo systemctl enable ckb-daemon</code></blockquote>"
```

Definition at line 20 of file mainwindow.cpp.

Referenced by MainWindow::MainWindow().

13.126.1.4 QString devpath

Definition at line 4 of file kbmanager.cpp.

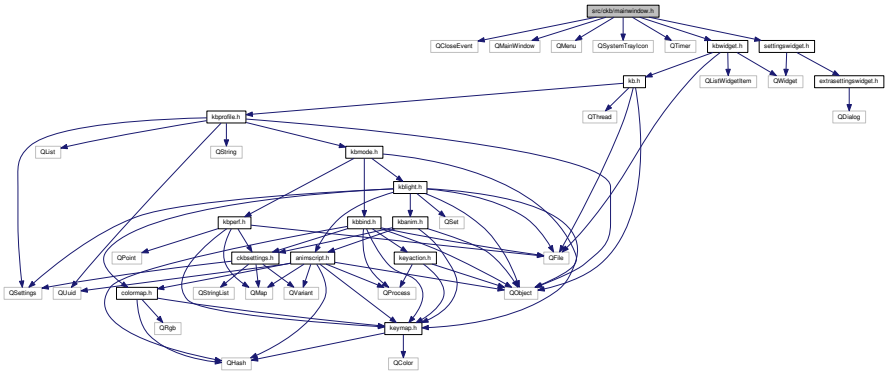
Referenced by MainWindow::MainWindow().

13.127 src/ckb/mainwindow.h File Reference

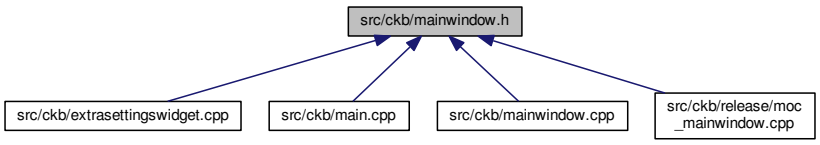
```
#include <QCloseEvent>
```

```
#include <QMainWindow>
#include <QMenu>
#include <QSystemTrayIcon>
#include <QTimer>
#include "kbwidget.h"
#include "settingswidget.h"
```

Include dependency graph for mainwindow.h:



This graph shows which files directly or indirectly include this file:



Data Structures

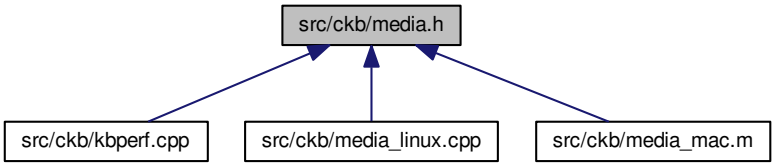
- class MainWindow

Namespaces

- Ui

13.128 src/ckb/media.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define EXTERN_C extern
- #define ENUM_C(name) typedef enum
- #define ENUM_END_C(name) name

Enumerations

- enum muteState { UNKNOWN = -1, UNMUTED, MUTED }

Functions

- muteState getMuteState ()

13.128.1 Macro Definition Documentation

13.128.1.1 #define ENUM_C(*name*) typedef enum

Definition at line 13 of file media.h.

13.128.1.2 #define ENUM_END_C(*name*) name

Definition at line 14 of file media.h.

13.128.1.3 #define EXTERN_C extern

Definition at line 12 of file media.h.

13.128.2 Enumeration Type Documentation

13.128.2.1 enum muteState

Enumerator

UNKNOWN

UNMUTED

MUTED

Definition at line 19 of file media.h.

```
19                  {
20     UNKNOWN = -1,
21     UNMUTED,
22     MUTED
23 } ENUM_END_C(muteState);
```

13.128.3 Function Documentation

13.128.3.1 muteState getMuteState ( )

Definition at line 102 of file media_linux.cpp.

References lastKnown, MUTED, UNKNOWN, and UNMUTED.

Referenced by KbPerf::applyIndicators().

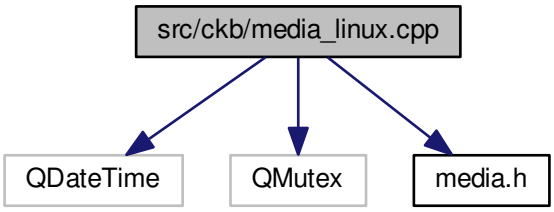
```
102                         {
103 #ifdef USE_LIBPULSE
104     static pa_threaded_mainloop* mainLoop = nullptr;
105
106     QMutexLocker locker(&mutex);
107
108     //Setup main loop thread used for communicating with Pulse Audio. All
109     //communication is done asynchronously.
110     if(mainLoop == nullptr){
111         mainLoop = pa_threaded_mainloop_new();
112         if(mainLoop == nullptr)
113             return lastKnown;
114
115         if(pa_threaded_mainloop_start(mainLoop) != 0){
116             pa_threaded_mainloop_free(mainLoop);
117             mainLoop = nullptr;
118             return lastKnown;
119         }
120     }
121
122     //Connect to the local Pulse Audio server. It's usually running but a
123     //reconnect is attempted periodically whenever the connection fails or is
124     //terminated.
125     if(paContext == nullptr && QDateTime::currentMSecsSinceEpoch() >= reconnectTime){
126         pa_threaded_mainloop_lock(mainLoop);
127         {
128             pa_mainloop_api* api = pa_threaded_mainloop_get_api(mainLoop);
129             Q_ASSERT(api != nullptr);
130
131             paContext = pa_context_new(api, "QPulse");
132             Q_ASSERT(paContext != nullptr);
133             pa_context_set_state_callback(paContext, &ContextStateCallback, nullptr);
134             pa_context_connect(paContext, nullptr, PA_CONTEXT_NOFAIL, nullptr);
135         }
136         pa_threaded_mainloop_unlock(mainLoop);
137     }
138 #endif
139
140     return lastKnown;
141 }
```

Here is the caller graph for this function:

13.129 src/ckb/media_linux.cpp File Reference

```
#include <QDateTime>
#include <QMutex>
#include "media.h"
```

Include dependency graph for media_linux.cpp:



Functions

- muteState getMuteState ()

Variables

- static muteState lastKnown = UNKNOWN

13.129.1 Function Documentation

13.129.1.1 muteState getMuteState ( )

Definition at line 102 of file media_linux.cpp.

References lastKnown.

```
102                         {
103 #ifdef USE_LIBPULSE
104     static pa_threaded_mainloop* mainLoop = nullptr;
105
106     QMutexLocker locker(&mutex);
107
108     //Setup main loop thread used for communicating with Pulse Audio. All
109     //communication is done asynchronously.
110     if(mainLoop == nullptr){
111         mainLoop = pa_threaded_mainloop_new();
112         if(mainLoop == nullptr)
113             return lastKnown;
114
115         if(pa_threaded_mainloop_start(mainLoop) != 0){
116             pa_threaded_mainloop_free(mainLoop);
117             mainLoop = nullptr;
118             return lastKnown;
119         }
120     }
121
122     //Connect to the local Pulse Audio server. It's usually running but a
123     //reconnect is attempted periodically whenever the connection fails or is
124     //terminated.
125     if(paContext == nullptr && QDateTime::currentMSecsSinceEpoch() >= reconnectTime){
126         pa_threaded_mainloop_lock(mainLoop);
127         {
128             pa_mainloop_api* api = pa_threaded_mainloop_get_api(mainLoop);
```

```
129             Q_ASSERT(api != nullptr);
130
131             paContext = pa_context_new(api, "QPulse");
132             Q_ASSERT(paContext != nullptr);
133             pa_context_set_state_callback(paContext, &ContextStateCallback, nullptr);
134             pa_context_connect(paContext, nullptr, PA_CONTEXT_NOFAIL, nullptr);
135         }
136         pa_threaded_mainloop_unlock(mainLoop);
137     }
138 #endif
139
140     return lastKnown;
141 }
```

13.129.2 Variable Documentation

13.129.2.1 muteState lastKnown = UNKNOWN [static]

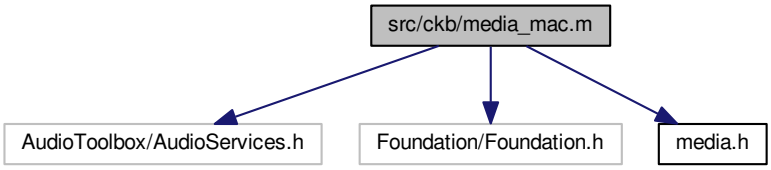
Definition at line 13 of file media_linux.cpp.

Referenced by getMuteState().

13.130 src/ckb/media_mac.m File Reference

```
#import <AudioToolbox/AudioServices.h>
#import <Foundation/Foundation.h>
#include "media.h"
```

Include dependency graph for media_mac.m:



Functions

- muteState getMuteState ()
- void disableAppNap ()

13.130.1 Function Documentation

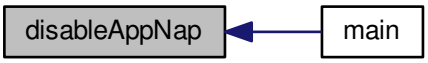
13.130.1.1 void disableAppNap ( )

Definition at line 34 of file media_mac.m.

Referenced by main().

```
34                  {
35     [[NSProcessInfo processInfo] beginActivityWithOptions:NSActivityUserInitiatedAllowingIdleSystemSleep
       reason:@"Keyboard animation"];
36 }
```

Here is the caller graph for this function:



13.130.1.2 muteState getMuteState ( )

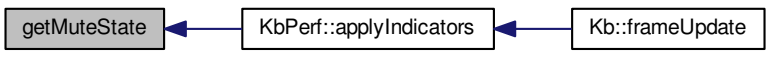
Definition at line 5 of file media_mac.m.

References MUTED, UNKNOWN, and UNMUTED.

Referenced by KbPerf::applyIndicators().

```
5                       {
6     // Get the system's default audio device
7     AudioObjectPropertyAddress propertyAddress = {
8         kAudioHardwarePropertyDefaultOutputDevice,
9         kAudioObjectPropertyScopeGlobal,
10         kAudioObjectPropertyElementMaster
11     };
12     AudioDeviceID deviceID = 0;
13     UInt32 dataSize = sizeof(deviceID);
14     if(AudioHardwareServiceGetPropertyData(kAudioObjectSystemObject, &propertyAddress, 0, NULL, &dataSize,
       &deviceID)
15             != kAudioHardwareNoError){
16         return UNKNOWN;
17     }
18     // Grab the mute property
19     AudioObjectPropertyAddress propertyAddress2 = {
20         kAudioDevicePropertyMute,
21         kAudioDevicePropertyScopeOutput,
22         kAudioObjectPropertyElementMaster
23     };
24     if(!AudioHardwareServiceHasProperty(deviceID, &propertyAddress2))
25         return UNKNOWN;
26     bool state = 0;
27     dataSize = sizeof(state);
28     if(AudioHardwareServiceGetPropertyData(deviceID, &propertyAddress2, 0, NULL, &dataSize, &state)
29             != kAudioHardwareNoError)
30         return UNKNOWN;
31     return state ? MUTED : UNMUTED;
32 }
```

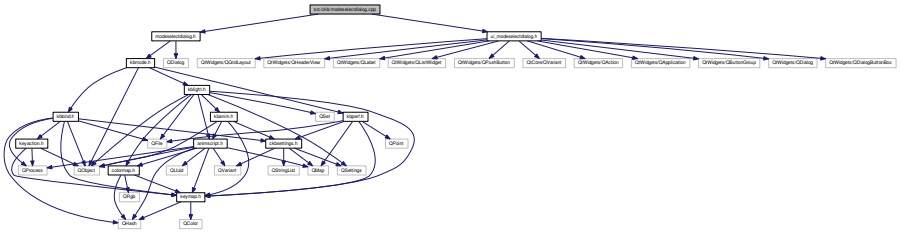
Here is the caller graph for this function:



13.131 src/ckb/modeselectdialog.cpp File Reference

```
#include "modeselectdialog.h"
#include "ui_modeselectdialog.h"
```

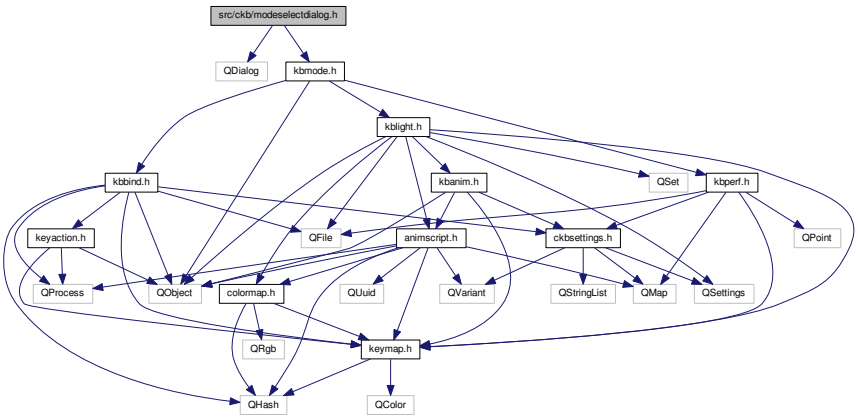
Include dependency graph for modeselectdialog.cpp:



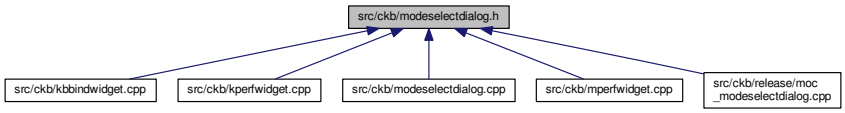
13.132 src/ckb/modeselectdialog.h File Reference

```
#include <QDialog>
#include "kbmode.h"
```

Include dependency graph for modeselectdialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class ModeSelectDialog

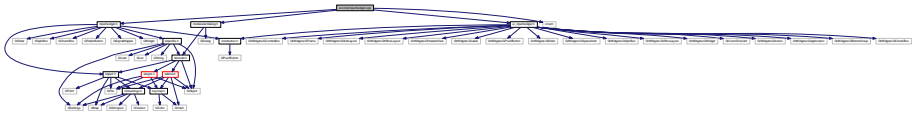
Namespaces

- Ui

13.133 src/ckb/mperfwidget.cpp File Reference

```
#include "mperfwidget.h"
#include "ui_mperfwidget.h"
#include "modeselectdialog.h"
#include <cmath>
```

Include dependency graph for mperfwidget.cpp:



Macros

- #define SET_START if(isSetting) return; isSetting = true
- #define SET_END isSetting = false

Functions

- int dpiExp (int value, int min, int max)
- int dpiLog (int value, int min, int max)
- int dpiRound (int value)

Variables

- static const QString xyLinkPath = "UI/DPI/UnlinkXY"
- static const double DPI_BASE = (double)KbPerf::DPI_MAX / (double)KbPerf::DPI_MIN

13.133.1 Macro Definition Documentation

13.133.1.1 #define SET_END isSetting = false

Definition at line 126 of file mperfwidget.cpp.

Referenced by MPerfWidget::boxXChanged(), MPerfWidget::boxYChanged(), MPerfWidget::sliderXMoved(), and MPerfWidget::sliderYMoved().

13.133.1.2 #define SET_START if(isSetting) return; isSetting = true

Definition at line 125 of file mperfwidget.cpp.

Referenced by MPerfWidget::boxXChanged(), MPerfWidget::boxYChanged(), MPerfWidget::sliderXMoved(), and MPerfWidget::sliderYMoved().

13.133.2 Function Documentation

13.133.2.1 int dpiExp (int *value,* int *min,* int *max*) [inline]

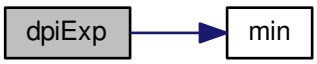
Definition at line 105 of file mperfwidget.cpp.

References DPI_BASE, KbPerf::DPI_MIN, and min().

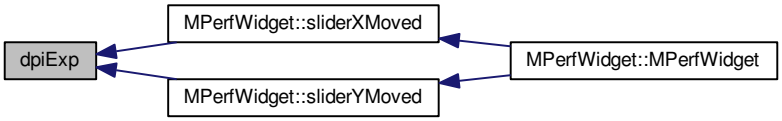
Referenced by MPerfWidget::sliderXMoved(), and MPerfWidget::sliderYMoved().

```
105                                                 {
106     double val = (value - min) / (double)(max - min);
107     val = pow(DPI_BASE, val);
108     return round(KbPerf::DPI_MIN * val);
109 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.133.2.2 int dpiLog (int *value*, int *min*, int *max*) [inline]

Definition at line 111 of file mperfwidget.cpp.

References DPI_BASE, and KbPerf::DPI_MIN.

Referenced by MPerfWidget::boxXChanged(), and MPerfWidget::boxYChanged().

```
111                                                 {
112     double val = value / (double)KbPerf::DPI_MIN;
113     val = log(val) / log(DPI_BASE);
114     return round(val * (max - min) + min);
115 }
```

Here is the caller graph for this function:

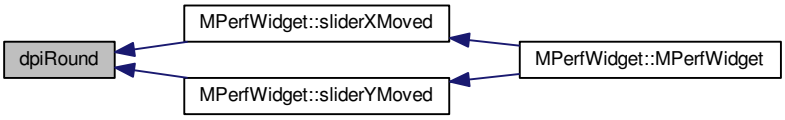
13.133.2.3 int dpiRound (int *value*) [inline]

Definition at line 117 of file mperfwidget.cpp.

Referenced by MPerfWidget::sliderXMoved(), and MPerfWidget::sliderYMoved().

```
117                                {
118     if(value > 1000)
119         return round(value / 100.) * 100;
120     else
121         return round(value / 50.) * 50;
122 }
```

Here is the caller graph for this function:



13.133.3 Variable Documentation

13.133.3.1 const double DPI_BASE = (double)KbPerf::DPI_MAX / (double)KbPerf::DPI_MIN [static]

Definition at line 103 of file mperfwidget.cpp.

Referenced by dpiExp(), and dpiLog().

13.133.3.2 const QString xyLinkPath = "UI/DPI/UnlinkXY" [static]

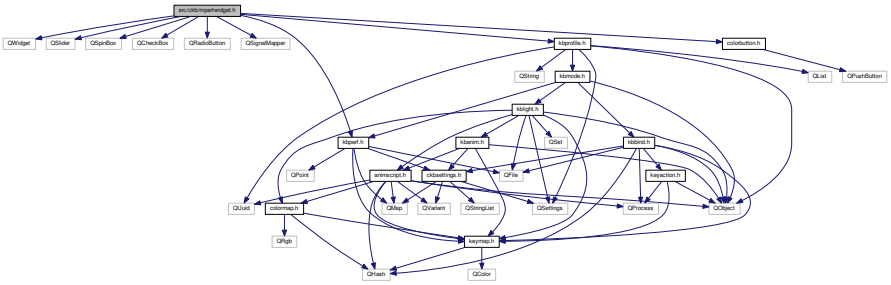
Definition at line 6 of file mperfwidget.cpp.

Referenced by MPerfWidget::on_xyBox_clicked().

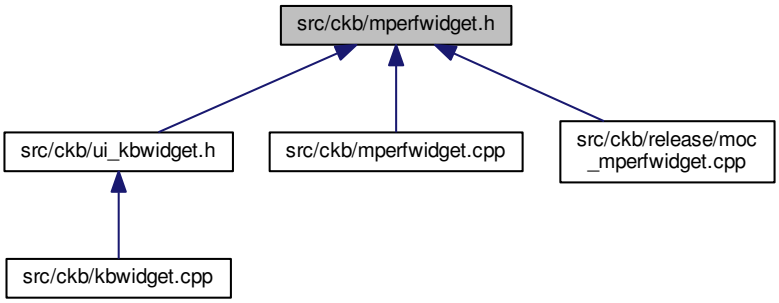
13.134 src/ckb/mperfwidget.h File Reference

```
#include <QWidget>
#include <QSlider>
#include <QSpinBox>
#include <QCheckBox>
#include <QRadioButton>
#include <QSignalMapper>
#include "kbperf.h"
#include "kbprofile.h"
#include "colorbutton.h"
```

Include dependency graph for mperfwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class MPerfWidget
- struct MPerfWidget::DpiUi

Namespaces

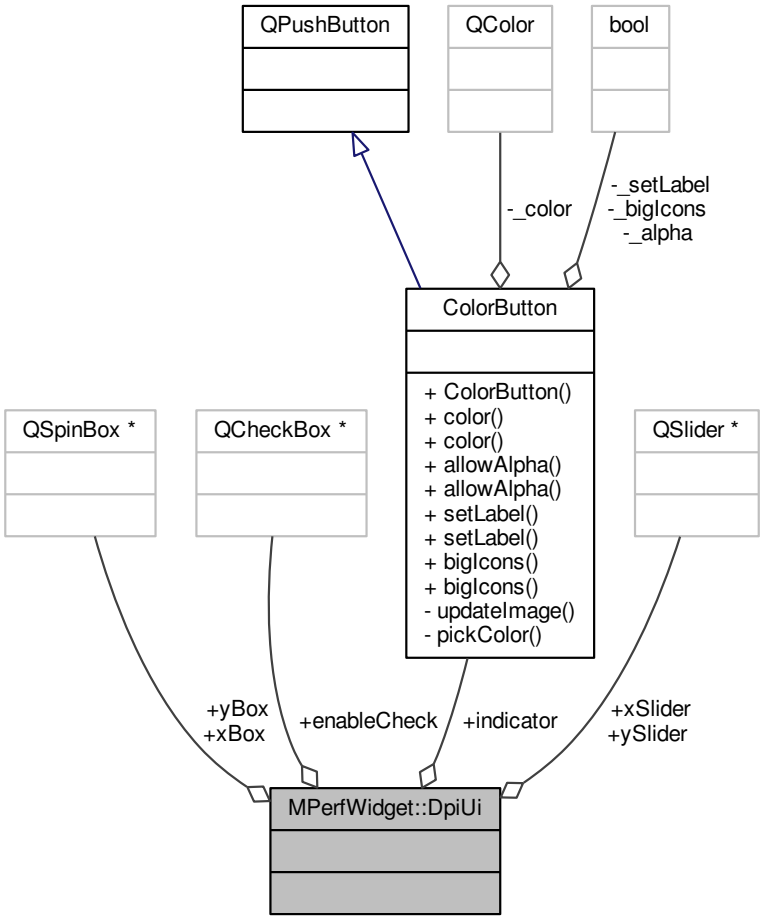
- Ui

13.134.1 Data Structure Documentation

13.134.1.1 struct MPerfWidget::DpiUi

Definition at line 37 of file mperfwidget.h.

Collaboration diagram for MPerfWidget::DpiUi:



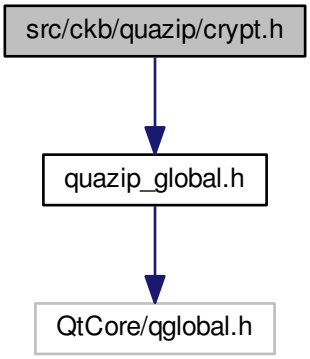
Data Fields

QCheckBox *	enableCheck	
ColorButton *	indicator	
QSpinBox *	xBox	
QSlider *	xSlider	
QSpinBox *	yBox	
QSlider *	ySlider	

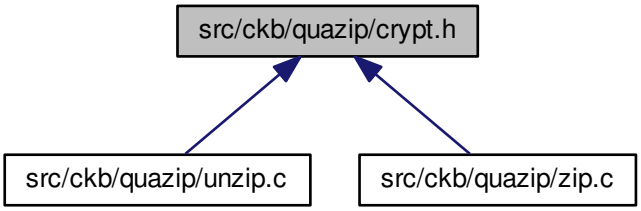
13.135 src/ckb/quazip/crypt.h File Reference

```
#include "quazip_global.h"
```

Include dependency graph for crypt.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define CRC32(c, b) ((*(pcrc_32_tab+(((int)(c) ^ (b)) & 0xff))) ^ ((c) >> 8))
- #define zdecode(pkeys, pcrc_32_tab, c) (update_keys(pkeys,pcrc_32_tab,c ^= decrypt_byte(pkeys,pcrc_32_tab)))
- #define zencode(pkeys, pcrc_32_tab, c, t) (t=decrypt_byte(pkeys,pcrc_32_tab), update_keys(pkeys,pcrc_32_tab,c), t^(c))

Functions

- static int decrypt_byte (unsigned long *pkeys, const z_crc_t FAR *pcrc_32_tab)
- static int update_keys (unsigned long *pkeys, const z_crc_t FAR *pcrc_32_tab, int c)
- static void init_keys (const char *passwd, unsigned long *pkeys, const z_crc_t FAR *pcrc_32_tab)

13.135.1 Macro Definition Documentation

13.135.1.1 #define CRC32(c, b) ((*(pcrc_32_tab+(((int)(c) ^ (b)) & 0xff))) ^ ((c) >> 8))

Definition at line 32 of file crypt.h.

Referenced by update_keys().

13.135.1.2 #define zdecode(pkeys, pcrc_32_tab, c) (update_keys(pkeys,pcrc_32_tab,c ^= decrypt_byte(pkeys,pcrc_32_tab)))

Definition at line 79 of file crypt.h.

Referenced by unzOpenCurrentFile3(), and unzReadCurrentFile().

13.135.1.3 #define zencode(pkeys, pcrc_32_tab, c, t) (t=decrypt_byte(pkeys,pcrc_32_tab), update_keys(pkeys,pcrc_32_tab,c), t^(c))

Definition at line 82 of file crypt.h.

Referenced by zip64FlushWriteBuffer().

13.135.2 Function Documentation

13.135.2.1 static int decrypt_byte (unsigned long * pkeys, const z_crc_t FAR * pcrc_32_tab) [static]

Definition at line 37 of file crypt.h.

```
38 {
39     //(void) pcrc_32_tab; /* avoid "unused parameter" warning */
40     unsigned temp;  /* POTENTIAL BUG:  temp*(temp^1) may overflow in an
41                      * unpredictable manner on 16-bit systems; not a problem
42                      * with any known compiler so far, though */
43
44     temp = ((unsigned)(*(pkeys+2)) & 0xffff) | 2;
45     return (int)(((temp * (temp ^ 1)) >> 8) & 0xff);
46 }
```

13.135.2.2 static void init_keys (const char * passwd, unsigned long * pkeys, const z_crc_t FAR * pcrc_32_tab) [static]

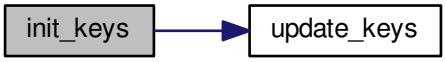
Definition at line 68 of file crypt.h.

References update_keys().

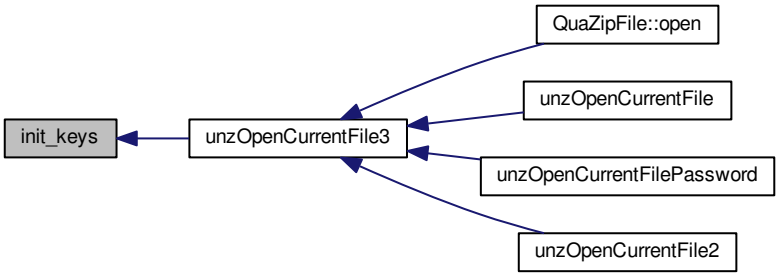
Referenced by unzOpenCurrentFile3().

```
69 {
70     *(pkeys+0) = 305419896L;
71     *(pkeys+1) = 591751049L;
72     *(pkeys+2) = 878082192L;
73     while (*passwd != '\0') {
74         update_keys(pkeys,pcrc_32_tab,(int)*passwd);
75         passwd++;
76     }
77 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.135.2.3 static int update_keys (unsigned long * *pkeys,* const z_crc_t FAR * *pcrc_32_tab,* int *c*) [static]

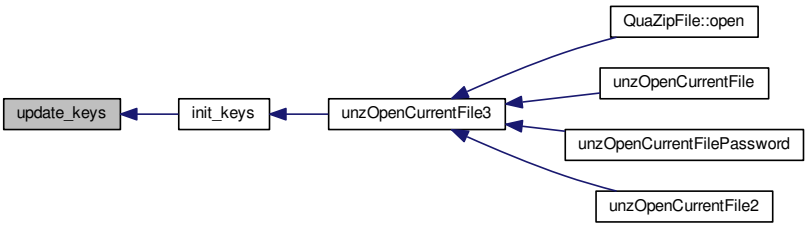
Definition at line 51 of file crypt.h.

References CRC32.

Referenced by init_keys().

```
52 {
53     (*(pkeys+0)) = CRC32((*(pkeys+0)), c);
54     (*(pkeys+1)) += (*(pkeys+0)) & 0xff;
55     (*(pkeys+1)) = (*(pkeys+1)) * 134775813L + 1;
56     {
57       register int keyshift = (int)((*(pkeys+1)) >> 24);
58       (*(pkeys+2)) = CRC32((*(pkeys+2)), keyshift);
59     }
60     return c;
61 }
```

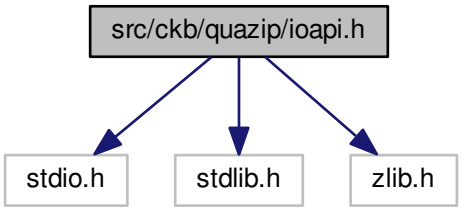
Here is the caller graph for this function:



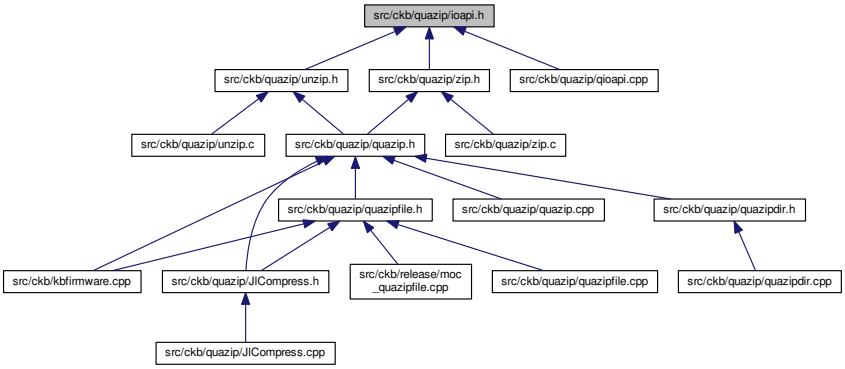
13.136 src/ckb/quazip/ioapi.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "zlib.h"
```

Include dependency graph for ioapi.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct zlib_filefunc_def_s
- struct zlib_filefunc64_def_s
- struct zlib_filefunc64_32_def_s

Macros

- #define __USE_FILE_OFFSET64
- #define __USE_LARGEFILE64
- #define _LARGEFILE64_SOURCE
- #define _FILE_OFFSET_BIT 64
- #define OF _Z_OF
- #define ZLIB_FILEFUNC_SEEK_CUR (1)
- #define ZLIB_FILEFUNC_SEEK_END (2)
- #define ZLIB_FILEFUNC_SEEK_SET (0)
- #define ZLIB_FILEFUNC_MODE_READ (1)
- #define ZLIB_FILEFUNC_MODE_WRITE (2)
- #define ZLIB_FILEFUNC_MODE_READWRITEFILTER (3)
- #define ZLIB_FILEFUNC_MODE_EXISTING (4)
- #define ZLIB_FILEFUNC_MODE_CREATE (8)
- #define ZCALLBACK
- #define ZREAD64(filefunc, filestream, buf, size) ((*((filefunc).zfile_func64.zread_file)) ((filefunc).zfile_-func64.opaque,filestream,buf,size))
- #define ZWRITE64(filefunc, filestream, buf, size) ((*((filefunc).zfile_func64.zwrite_file)) ((filefunc).zfile_-func64.opaque,filestream,buf,size))
- #define ZCLOSE64(filefunc, filestream) ((*((filefunc).zfile_func64.zclose_file)) ((filefunc).zfile_func64.-opaque,filestream))
- #define ZFAKECLOSE64(filefunc, filestream) ((*((filefunc).zfile_func64.zfakeclose_file)) ((filefunc).zfile_-func64.opaque,filestream))
- #define ZERROR64(filefunc, filestream) ((*((filefunc).zfile_func64.zerror_file)) ((filefunc).zfile_func64.-opaque,filestream))
- #define ZOPEN64(filefunc, filename, mode) (call_zopen64((&(filefunc)),(filename),(mode)))
- #define ZTELL64(filefunc, filestream) (call_ztell64((&(filefunc)),(filestream)))
- #define ZSEEK64(filefunc, filestream, pos, mode) (call_zseek64((&(filefunc)),(filestream),(pos),(mode)))

Typedefs

- typedef unsigned long long int ZPOS64_T
- typedef voidpf open_file_func _Z_OF ((voidpf opaque, voidpf file, int mode))

  Addition for GDAL : START.
- typedef struct zlib_filefunc_def_s zlib_filefunc_def
- typedef struct zlib_filefunc64_def_s zlib_filefunc64_def
- typedef struct zlib_filefunc64_32_def_s zlib_filefunc64_32_def

Functions

- void fill_qiodevice64_filefunc _Z_OF ((zlib_filefunc64_def *pzlib_filefunc_def))
- void fill_qiodevice_filefunc _Z_OF ((zlib_filefunc_def *pzlib_filefunc_def))
- voidpf call_zopen64 _Z_OF ((const zlib_filefunc64_32_def *pfilefunc, voidpf file, int mode))
- int call_zseek64 _Z_OF ((const zlib_filefunc64_32_def *pfilefunc, voidpf filestream, ZPOS64_T offset, int origin))

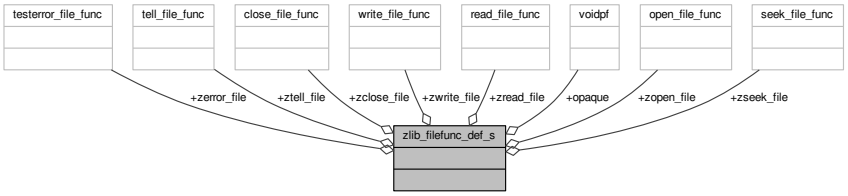
- ZPOS64_T call_ztell64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream))
- void fill_zlib_filefunc64_32_def_from_filefunc32 (zlib_filefunc64_32_def ∗p_filefunc64_32, const zlib_filefunc_def ∗p_filefunc32)

13.136.1 Data Structure Documentation

13.136.1.1 struct zlib_filefunc_def_s

Definition at line 143 of file ioapi.h.

Collaboration diagram for zlib_filefunc_def_s:



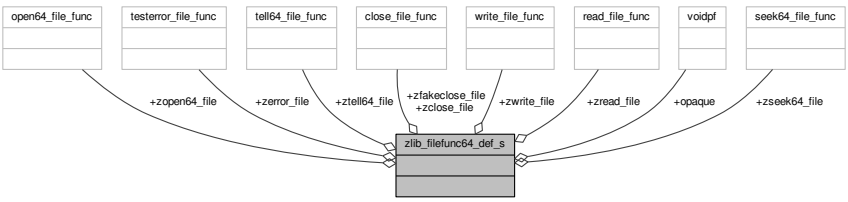
Data Fields

voidpf	opaque	
close_file_func	zclose_file	
testerror_file_func	zerror_file	
open_file_func	zopen_file	
read_file_func	zread_file	
seek_file_func	zseek_file	
tell_file_func	ztell_file	
write_file_func	zwrite_file	

13.136.1.2 struct zlib_filefunc64_def_s

Definition at line 159 of file ioapi.h.

Collaboration diagram for zlib_filefunc64_def_s:



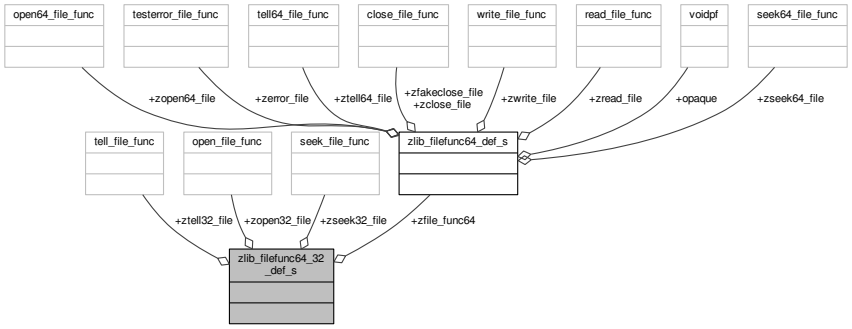
Data Fields

voidpf	opaque	
close_file_func	zclose_file	
testerror_file_-func	zerror_file	
close_file_func	zfakeclose_file	
open64_file_-func	zopen64_file	
read_file_func	zread_file	
seek64_file_func	zseek64_file	
tell64_file_func	ztell64_file	
write_file_func	zwrite_file	

13.136.1.3 struct zlib_filefunc64_32_def_s

Definition at line 176 of file ioapi.h.

Collaboration diagram for zlib_filefunc64_32_def_s:



Data Fields

zlib_filefunc64_-def	zfile_func64	
open_file_func	zopen32_file	
seek_file_func	zseek32_file	
tell_file_func	ztell32_file	

13.136.2 Macro Definition Documentation

13.136.2.1 #define __USE_FILE_OFFSET64

Definition at line 32 of file ioapi.h.

13.136.2.2 #define __USE_LARGEFILE64

Definition at line 35 of file ioapi.h.

13.136.2.3 #define _FILE_OFFSET_BIT 64

Definition at line 41 of file ioapi.h.

13.136.2.4 #define _LARGEFILE64_SOURCE

Definition at line 38 of file ioapi.h.

13.136.2.5 #define OF _Z_OF

Definition at line 106 of file ioapi.h.

13.136.2.6 #define ZCALLBACK

Definition at line 125 of file ioapi.h.

13.136.2.7 #define ZCLOSE64(*filefunc, filestream*) ((∗((filefunc).zfile_func64.zclose_file)) ((filefunc).zfile_func64.opaque,filestream))

Definition at line 189 of file ioapi.h.

Referenced by LoadCentralDirectoryRecord(), unzClose(), unzOpenInternal(), zipClose(), and zipOpen3().

13.136.2.8 #define ZERROR64(*filefunc, filestream*) ((∗((filefunc).zfile_func64.zerror_file)) ((filefunc).zfile_func64.opaque,filestream))

Definition at line 191 of file ioapi.h.

Referenced by unz64local_getByte(), and zip64local_getByte().

13.136.2.9 #define ZFAKECLOSE64(*filefunc, filestream*) ((∗((filefunc).zfile_func64.zfakeclose_file)) ((filefunc).zfile_func64.opaque,filestream))

Definition at line 190 of file ioapi.h.

Referenced by LoadCentralDirectoryRecord(), unzClose(), unzOpenInternal(), zipClose(), and zipOpen3().

13.136.2.10 #define ZLIB_FILEFUNC_MODE_CREATE (8)

Definition at line 118 of file ioapi.h.

Referenced by qiodevice_open_file_func(), and zipOpen3().

13.136.2.11 #define ZLIB_FILEFUNC_MODE_EXISTING (4)

Definition at line 117 of file ioapi.h.

Referenced by qiodevice_open_file_func(), unzOpenInternal(), and zipOpen3().

13.136.2.12 #define ZLIB_FILEFUNC_MODE_READ (1)

Definition at line 113 of file ioapi.h.

Referenced by qiodevice_open_file_func(), unzOpenInternal(), and zipOpen3().

13.136.2.13 #define ZLIB_FILEFUNC_MODE_READWRITEFILTER (3)

Definition at line 115 of file ioapi.h.

Referenced by qiodevice_open_file_func().

13.136.2.14 #define ZLIB_FILEFUNC_MODE_WRITE (2)

Definition at line 114 of file ioapi.h.

Referenced by zipOpen3().

13.136.2.15 #define ZLIB_FILEFUNC_SEEK_CUR (1)

Definition at line 109 of file ioapi.h.

Referenced by qiodevice64_seek_file_func(), qiodevice_seek_file_func(), and unz64local_GetCurrentFileInfoInternal().

13.136.2.16 #define ZLIB_FILEFUNC_SEEK_END (2)

Definition at line 110 of file ioapi.h.

Referenced by qiodevice64_seek_file_func(), qiodevice_seek_file_func(), unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), zip64local_SearchCentralDir(), and zip64local_SearchCentralDir64().

13.136.2.17 #define ZLIB_FILEFUNC_SEEK_SET (0)

Definition at line 111 of file ioapi.h.

Referenced by LoadCentralDirectoryRecord(), qiodevice64_seek_file_func(), qiodevice_seek_file_func(), unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), unzGetGlobalComment(), unzGetLocalExtrafield(), unzOpenInternal(), unzReadCurrentFile(), zip64local_SearchCentralDir(), zip64local_SearchCentralDir64(), and zipCloseFileInZipRaw64().

13.136.2.18 #define ZOPEN64(*filefunc, filename, mode*) (call_zopen64((&(filefunc)),(filename),(mode)))

Definition at line 199 of file ioapi.h.

Referenced by unzOpenInternal(), and zipOpen3().

13.136.2.19 #define ZREAD64(*filefunc, filestream, buf, size*) ((∗((filefunc).zfile_func64.zread_file)) ((filefunc).zfile_func64.opaque,filestream,buf,size))

Definition at line 185 of file ioapi.h.

Referenced by LoadCentralDirectoryRecord(), unz64local_getByte(), unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), unzGetGlobalComment(), unzGetLocalExtrafield(), unzOpenCurrentFile3(), unzReadCurrentFile(), zip64local_getByte(), zip64local_SearchCentralDir(), and zip64local_SearchCentralDir64().

13.136.2.20 #define ZSEEK64(*filefunc, filestream, pos, mode*) (call_zseek64((&(filefunc)),(filestream),(pos),(mode)))

Definition at line 201 of file ioapi.h.

Referenced by LoadCentralDirectoryRecord(), unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), unzGetGlobalComment(), unzGetLocalExtrafield(), unzOpenCurrentFile3(), unzOpenInternal(), unzReadCurrentFile(), zip64local_SearchCentralDir(), zip64local_SearchCentralDir64(), zipCloseFileInZipRaw64(), and zipOpen3().

13.136.2.21 #define ZTELL64(*filefunc, filestream*) (call_ztell64((&(filefunc)),(filestream)))

Definition at line 200 of file ioapi.h.

Referenced by unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), Write_LocalFileHeader(), zip64local_SearchCentralDir(), zip64local_SearchCentralDir64(), zipClose(), zipCloseFileInZipRaw64(), zipOpen3(), and zipOpenNewFileInZip4_64().

13.136.2.22 #define ZWRITE64(*filefunc, filestream, buf, size*) ((∗((filefunc).zfile_func64.zwrite_file)) ((filefunc).zfile_func64.opaque,filestream,buf,size))

Definition at line 186 of file ioapi.h.

Referenced by Write_GlobalComment(), Write_LocalFileHeader(), zip64FlushWriteBuffer(), zip64local_putValue(), zipClose(), and zipOpenNewFileInZip4_64().

13.136.3 Typedef Documentation

13.136.3.1 zipFile ZEXPORT zipOpen64 _Z_OF ((voidpf opaque, voidpf file, int mode))

Addition for GDAL : END.

Definition at line 132 of file ioapi.h.

13.136.3.2 typedef struct zlib_filefunc64_32_def_s zlib_filefunc64_32_def

13.136.3.3 typedef struct zlib_filefunc64_def_s zlib_filefunc64_def

13.136.3.4 typedef struct zlib_filefunc_def_s zlib_filefunc_def

13.136.3.5 typedef unsigned long long int ZPOS64_T

Definition at line 94 of file ioapi.h.

13.136.4 Function Documentation

13.136.4.1 void fill_qiodevice64_filefunc _Z_OF ((zlib_filefunc64_def ∗pzlib_filefunc_def))

13.136.4.2 void fill_qiodevice_filefunc _Z_OF ((zlib_filefunc_def ∗pzlib_filefunc_def))

13.136.4.3 voidpf call_zopen64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf file, int mode))

13.136.4.4 int call_zseek64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream, ZPOS64_T offset, int origin))

13.136.4.5 ZPOS64_T call_ztell64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream))

13.136.4.6 void fill_zlib_filefunc64_32_def_from_filefunc32 (zlib_filefunc64_32_def * *p_filefunc64_32,* const zlib_filefunc_def * *p_filefunc32*)

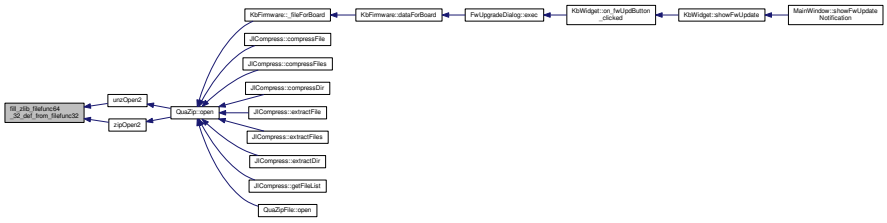
Definition at line 346 of file qioapi.cpp.

References zlib_filefunc_def_s::opaque, zlib_filefunc64_def_s::opaque, zlib_filefunc_def_s::zclose_file, zlib_filefunc64_def_s::zclose_file, zlib_filefunc_def_s::zerror_file, zlib_filefunc64_def_s::zerror_file, zlib_filefunc64_def_s::zfakeclose_file, zlib_filefunc64_32_def_s::zfile_func64, zlib_filefunc64_32_def_s::zopen32_file, zlib_filefunc64_def_s::zopen64_file, zlib_filefunc_def_s::zopen_file, zlib_filefunc_def_s::zread_file, zlib_filefunc64_def_s::zread_file, zlib_filefunc64_32_def_s::zseek32_file, zlib_filefunc64_def_s::zseek64_file, zlib_filefunc_def_s::zseek_file, zlib_filefunc64_32_def_s::ztell32_file, zlib_filefunc64_def_s::ztell64_file, zlib_filefunc_def_s::ztell_file, zlib_filefunc_def_s::zwrite_file, and zlib_filefunc64_def_s::zwrite_file.

Referenced by unzOpen2(), and zipOpen2().

```
347 {
348     p_filefunc64_32->zfile_func64.zopen64_file = NULL;
349     p_filefunc64_32->zopen32_file = p_filefunc32->zopen_file;
350     p_filefunc64_32->zfile_func64.zerror_file = p_filefunc32->
    zerror_file;
351     p_filefunc64_32->zfile_func64.zread_file = p_filefunc32->
    zread_file;
352     p_filefunc64_32->zfile_func64.zwrite_file = p_filefunc32->
    zwrite_file;
353     p_filefunc64_32->zfile_func64.ztell64_file = NULL;
354     p_filefunc64_32->zfile_func64.zseek64_file = NULL;
355     p_filefunc64_32->zfile_func64.zclose_file = p_filefunc32->
    zclose_file;
356     p_filefunc64_32->zfile_func64.zerror_file = p_filefunc32->
    zerror_file;
357     p_filefunc64_32->zfile_func64.opaque = p_filefunc32->
    opaque;
358     p_filefunc64_32->zfile_func64.zfakeclose_file = NULL;
359     p_filefunc64_32->zseek32_file = p_filefunc32->zseek_file;
360     p_filefunc64_32->ztell32_file = p_filefunc32->ztell_file;
361 }
```

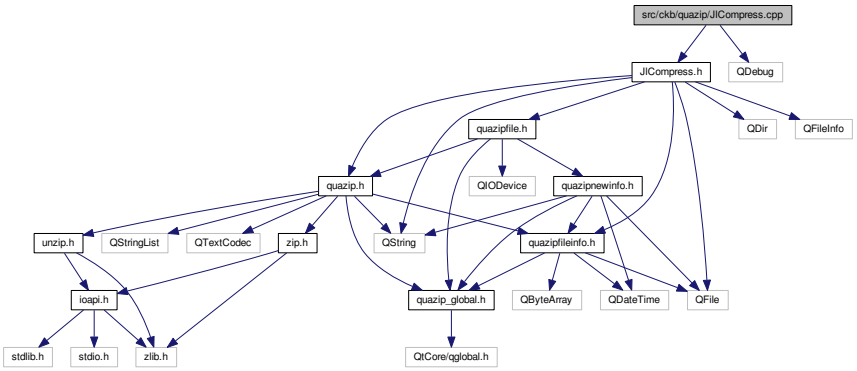
Here is the caller graph for this function:



13.137 src/ckb/quazip/JlCompress.cpp File Reference

```
#include "JlCompress.h"
#include <QDebug>
```

Include dependency graph for JlCompress.cpp:



Functions

- static bool copyData (QIODevice &inFile, QIODevice &outFile)

13.137.1 Function Documentation

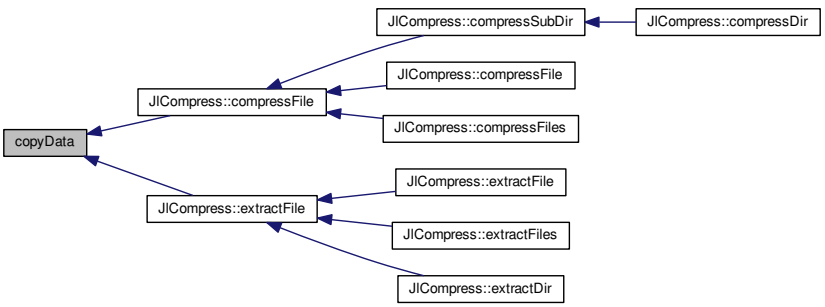
13.137.1.1 static bool copyData (QIODevice & *inFile*, QIODevice & *outFile*) [static]

Definition at line 29 of file JlCompress.cpp.

Referenced by JlCompress::compressFile(), and JlCompress::extractFile().

```
30 {
31     while (!inFile.atEnd()) {
32         char buf[4096];
33         qint64 readLen = inFile.read(buf, 4096);
34         if (readLen <= 0)
35             return false;
36         if (outFile.write(buf, readLen) != readLen)
37             return false;
38     }
39     return true;
40 }
```

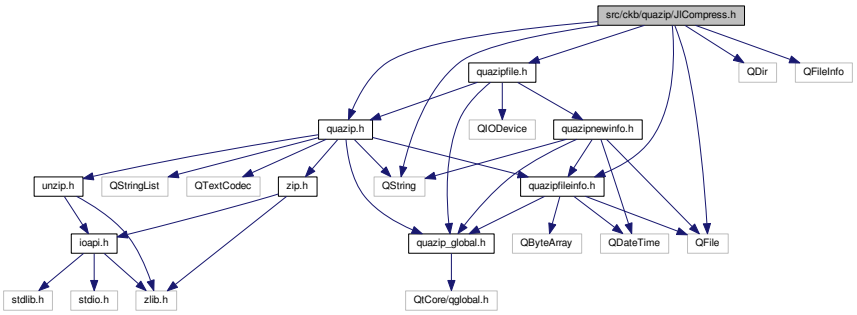
Here is the caller graph for this function:



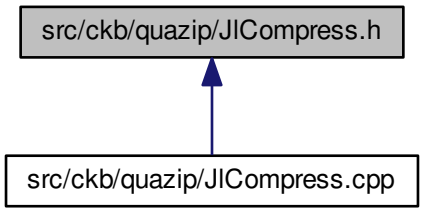
13.138 src/ckb/quazip/JlCompress.h File Reference

```
#include "quazip.h"
#include "quazipfile.h"
#include "quazipfileinfo.h"
#include <QString>
#include <QDir>
#include <QFileInfo>
#include <QFile>
```

Include dependency graph for JlCompress.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class JlCompress

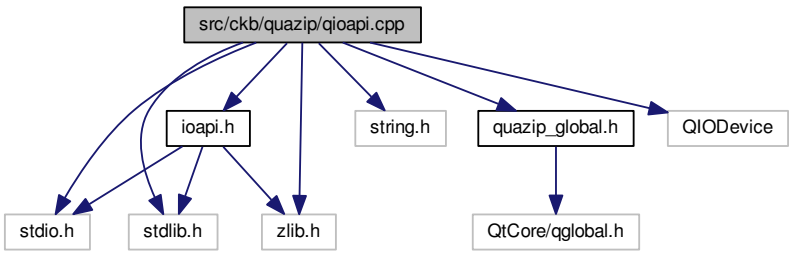
  Utility class for typical operations.

13.139 src/ckb/quazip/qioapi.cpp File Reference

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>
#include "zlib.h"
#include "ioapi.h"
#include "quazip_global.h"
#include <QIODevice>
```

Include dependency graph for qioapi.cpp:



Data Structures

- struct QIODevice_descriptor

Macros

- #define SEEK_CUR 1
- #define SEEK_END 2
- #define SEEK_SET 0

Functions

- voidpf call_zopen64 (const zlib_filefunc64_32_def ∗pfilefunc, voidpf file, int mode)
- int call_zseek64 (const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream, ZPOS64_T offset, int origin)
- ZPOS64_T call_ztell64 (const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream)
- voidpf qiodevice_open_file_func (voidpf opaque, voidpf file, int mode)
- uLong qiodevice_read_file_func (voidpf opaque, voidpf stream, void ∗buf, uLong size)
- uLong qiodevice_write_file_func (voidpf opaque, voidpf stream, const void ∗buf, uLong size)
- uLong qiodevice_tell_file_func (voidpf opaque, voidpf stream)
- ZPOS64_T qiodevice64_tell_file_func (voidpf opaque, voidpf stream)
- int qiodevice_seek_file_func (voidpf, voidpf stream, uLong offset, int origin)
- int qiodevice64_seek_file_func (voidpf, voidpf stream, ZPOS64_T offset, int origin)
- int qiodevice_close_file_func (voidpf opaque, voidpf stream)
- int qiodevice_fakeclose_file_func (voidpf opaque, voidpf)
- int qiodevice_error_file_func (voidpf, voidpf)
- void fill_qiodevice_filefunc (zlib_filefunc_def ∗pzlib_filefunc_def)
- void fill_qiodevice64_filefunc (zlib_filefunc64_def ∗pzlib_filefunc_def)
- void fill_zlib_filefunc64_32_def_from_filefunc32 (zlib_filefunc64_32_def ∗p_filefunc64_32, const zlib_filefunc_def ∗p_filefunc32)

13.139.1 Macro Definition Documentation

13.139.1.1 #define SEEK_CUR 1

Definition at line 29 of file qioapi.cpp.

13.139.1.2 #define SEEK_END 2

Definition at line 33 of file qioapi.cpp.

13.139.1.3 #define SEEK_SET 0

Definition at line 37 of file qioapi.cpp.

Referenced by unzOpenCurrentFile3().

13.139.2 Function Documentation

13.139.2.1 voidpf call_zopen64 (const zlib_filefunc64_32_def * *pfilefunc,* voidpf *file,* int *mode*)

Definition at line 40 of file qioapi.cpp.

References zlib_filefunc64_def_s::opaque, zlib_filefunc64_32_def_s::zfile_func64, zlib_filefunc64_32_def_s::zopen32_file, and zlib_filefunc64_def_s::zopen64_file.

```
41 {
42     if (pfilefunc->zfile_func64.zopen64_file != NULL)
43         return (*(pfilefunc->zfile_func64.zopen64_file)) (pfilefunc->
      zfile_func64.opaque,file,mode);
44     else
45     {
46         return (*(pfilefunc->zopen32_file))(pfilefunc->zfile_func64.
      opaque,file,mode);
47     }
48 }
```

13.139.2.2 int call_zseek64 (const zlib_filefunc64_32_def * *pfilefunc,* voidpf *filestream,* ZPOS64_T *offset,* int *origin*)

Definition at line 50 of file qioapi.cpp.

References zlib_filefunc64_def_s::opaque, zlib_filefunc64_32_def_s::zfile_func64, zlib_filefunc64_32_def_s::zseek32_file, and zlib_filefunc64_def_s::zseek64_file.

```
51 {
52     if (pfilefunc->zfile_func64.zseek64_file != NULL)
53         return (*(pfilefunc->zfile_func64.zseek64_file)) (pfilefunc->
      zfile_func64.opaque,filestream,offset,origin);
54     else
55     {
56         uLong offsetTruncated = (uLong)offset;
57         if (offsetTruncated != offset)
58             return -1;
59         else
60             return (*(pfilefunc->zseek32_file))(pfilefunc->
      zfile_func64.opaque,filestream,offsetTruncated,origin);
61     }
62 }
```

13.139.2.3 ZPOS64_T call_ztell64 (const zlib_filefunc64_32_def * *pfilefunc,* voidpf *filestream*)

Definition at line 64 of file qioapi.cpp.

References zlib_filefunc64_def_s::opaque, zlib_filefunc64_32_def_s::zfile_func64, zlib_filefunc64_def_s::zseek64_file, zlib_filefunc64_32_def_s::ztell32_file, and zlib_filefunc64_def_s::ztell64_file.

```
65 {
66     if (pfilefunc->zfile_func64.zseek64_file != NULL)
67         return (*(pfilefunc->zfile_func64.ztell64_file)) (pfilefunc->
       zfile_func64.opaque,filestream);
68     else
69     {
70         uLong tell_uLong = (*(pfilefunc->ztell32_file))(pfilefunc->
       zfile_func64.opaque,filestream);
71         if ((tell_uLong) == ((uLong)-1))
72             return (ZPOS64_T)-1;
73         else
74             return tell_uLong;
75     }
76 }
```

13.139.2.4 void fill_qiodevice64_filefunc (zlib_filefunc64_def * *pzlib_filefunc_def*)

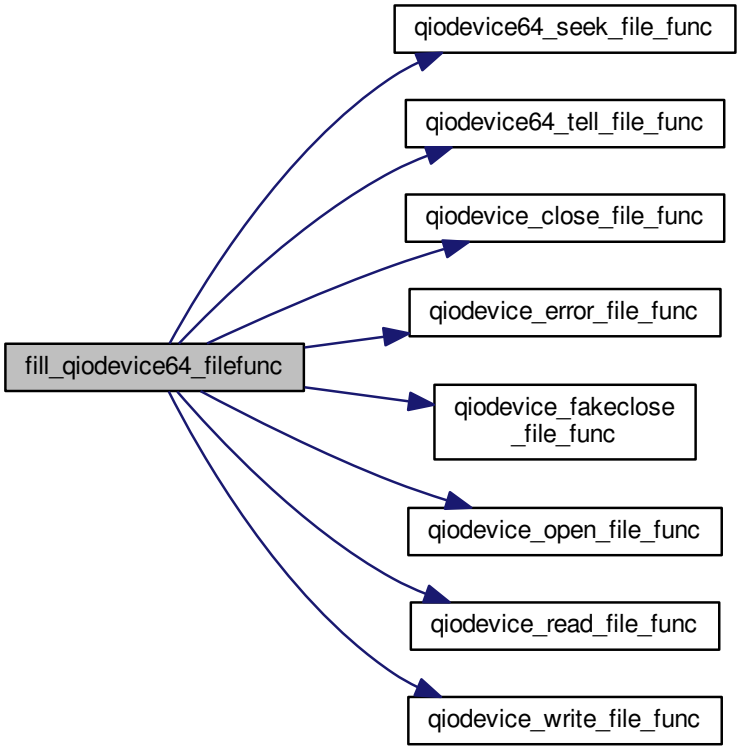
Definition at line 331 of file qioapi.cpp.

References zlib_filefunc64_def_s::opaque, qiodevice64_seek_file_func(), qiodevice64_tell_file_func(), qiodevice_close_file_func(), qiodevice_error_file_func(), qiodevice_fakeclose_file_func(), qiodevice_open_file_func(), qiodevice_read_file_func(), qiodevice_write_file_func(), zlib_filefunc64_def_s::zclose_file, zlib_filefunc64_def_s::zerror_file, zlib_filefunc64_def_s::zfakeclose_file, zlib_filefunc64_def_s::zopen64_file, zlib_filefunc64_def_s::zread_file, zlib_filefunc64_def_s::zseek64_file, zlib_filefunc64_def_s::ztell64_file, and zlib_filefunc64_def_s::zwrite_file.

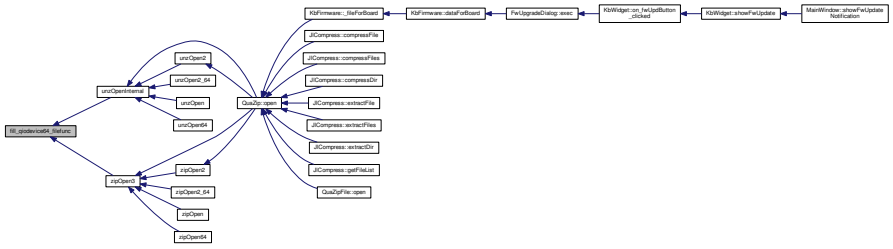
Referenced by unzOpenInternal(), and zipOpen3().

```
333 {
334     // Open functions are the same for Qt.
335     pzlib_filefunc_def->zopen64_file = qiodevice_open_file_func;
336     pzlib_filefunc_def->zread_file = qiodevice_read_file_func;
337     pzlib_filefunc_def->zwrite_file = qiodevice_write_file_func;
338     pzlib_filefunc_def->ztell64_file = qiodevice64_tell_file_func;
339     pzlib_filefunc_def->zseek64_file = qiodevice64_seek_file_func;
340     pzlib_filefunc_def->zclose_file = qiodevice_close_file_func;
341     pzlib_filefunc_def->zerror_file = qiodevice_error_file_func;
342     pzlib_filefunc_def->opaque = new QIODevice_descriptor;
343     pzlib_filefunc_def->zfakeclose_file =
      qiodevice_fakeclose_file_func;
344 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.139.2.5 void fill_qiodevice_filefunc (zlib_filefunc_def * *pzlib_filefunc_def*)

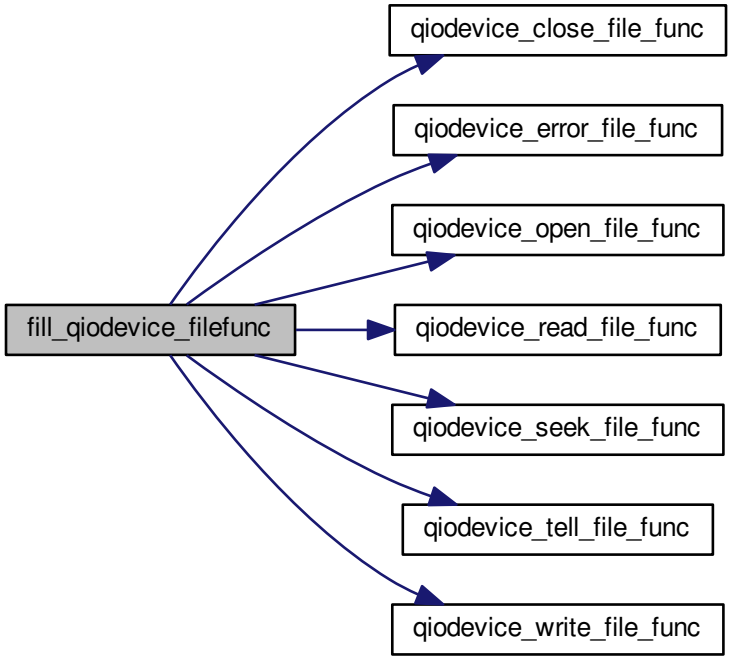
Definition at line 318 of file qioapi.cpp.

References zlib_filefunc_def_s::opaque, qiodevice_close_file_func(), qiodevice_error_file_func(), qiodevice_open_file_func(), qiodevice_read_file_func(), qiodevice_seek_file_func(), qiodevice_tell_file_func(), qiodevice_write_file_func(), zlib_filefunc_def_s::zclose_file, zlib_filefunc_def_s::zerror_file, zlib_filefunc_def_s::zopen_file, zlib_filefunc_def_s::zread_file, zlib_filefunc_def_s::zseek_file, zlib_filefunc_def_s::ztell_file, and zlib_filefunc_def_s::zwrite_file.

```
320 {
```

```
321     pzlib_filefunc_def->zopen_file = qiodevice_open_file_func;
322     pzlib_filefunc_def->zread_file = qiodevice_read_file_func;
323     pzlib_filefunc_def->zwrite_file = qiodevice_write_file_func;
324     pzlib_filefunc_def->ztell_file = qiodevice_tell_file_func;
325     pzlib_filefunc_def->zseek_file = qiodevice_seek_file_func;
326     pzlib_filefunc_def->zclose_file = qiodevice_close_file_func;
327     pzlib_filefunc_def->zerror_file = qiodevice_error_file_func;
328     pzlib_filefunc_def->opaque = new QIODevice_descriptor;
329 }
```

Here is the call graph for this function:



13.139.2.6 void fill_zlib_filefunc64_32_def_from_filefunc32 (zlib_filefunc64_32_def * *p_filefunc64_32*, const zlib_filefunc_def * *p_filefunc32*)

Definition at line 346 of file qioapi.cpp.

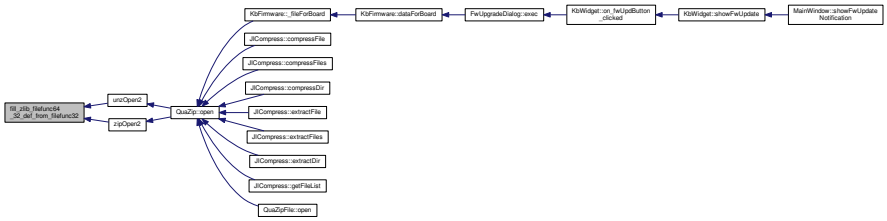
References zlib_filefunc_def_s::opaque, zlib_filefunc64_def_s::opaque, zlib_filefunc_def_s::zclose_file, zlib_filefunc64_def_s::zclose_file, zlib_filefunc_def_s::zerror_file, zlib_filefunc64_def_s::zerror_file, zlib_filefunc64_def_s::zfakeclose_file, zlib_filefunc64_32_def_s::zfile_func64, zlib_filefunc64_32_def_s::zopen32_file, zlib_filefunc64_def_s::zopen64_file, zlib_filefunc_def_s::zopen_file, zlib_filefunc_def_s::zread_file, zlib_filefunc64_def_s::zread_file, zlib_filefunc64_32_def_s::zseek32_file, zlib_filefunc64_def_s::zseek64_file, zlib_filefunc_def_s::zseek_file, zlib_filefunc64_32_def_s::ztell32_file, zlib_filefunc64_def_s::ztell64_file, zlib_filefunc_def_s::ztell_file, zlib_filefunc_def_s::zwrite_file, and zlib_filefunc64_def_s::zwrite_file.

Referenced by unzOpen2(), and zipOpen2().

```
347 {
348     p_filefunc64_32->zfile_func64.zopen64_file = NULL;
349     p_filefunc64_32->zopen32_file = p_filefunc32->zopen_file;
350     p_filefunc64_32->zfile_func64.zerror_file = p_filefunc32->
      zerror_file;
351     p_filefunc64_32->zfile_func64.zread_file = p_filefunc32->
```

```
    zread_file;
352   p_filefunc64_32->zfile_func64.zwrite_file = p_filefunc32->
    zwrite_file;
353   p_filefunc64_32->zfile_func64.ztell64_file = NULL;
354   p_filefunc64_32->zfile_func64.zseek64_file = NULL;
355   p_filefunc64_32->zfile_func64.zclose_file = p_filefunc32->
    zclose_file;
356   p_filefunc64_32->zfile_func64.zerror_file = p_filefunc32->
    zerror_file;
357   p_filefunc64_32->zfile_func64.opaque = p_filefunc32->
    opaque;
358   p_filefunc64_32->zfile_func64.zfakeclose_file = NULL;
359   p_filefunc64_32->zseek32_file = p_filefunc32->zseek_file;
360   p_filefunc64_32->ztell32_file = p_filefunc32->ztell_file;
361 }
```

Here is the caller graph for this function:



13.139.2.7 int qiodevice64_seek_file_func (voidpf , voidpf *stream*, ZPOS64_T *offset*, int *origin*)

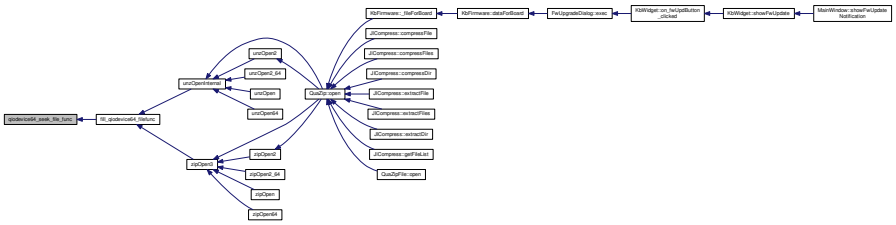
Definition at line 244 of file qioapi.cpp.

References ZLIB_FILEFUNC_SEEK_CUR, ZLIB_FILEFUNC_SEEK_END, and ZLIB_FILEFUNC_SEEK_SET.

Referenced by fill_qiodevice64_filefunc().

```
249 {
250     QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
251     if (iodevice->isSequential()) {
252         if (origin == ZLIB_FILEFUNC_SEEK_END
253                 && offset == 0) {
254             // sequential devices are always at end (needed in mdAppend)
255             return 0;
256         } else {
257             qWarning("qiodevice_seek_file_func() called for sequential device");
258             return -1;
259         }
260     }
261     qint64 qiodevice_seek_result=0;
262     int ret;
263     switch (origin)
264     {
265     case ZLIB_FILEFUNC_SEEK_CUR :
266         qiodevice_seek_result = ((QIODevice*)stream)->pos() + offset;
267         break;
268     case ZLIB_FILEFUNC_SEEK_END :
269         qiodevice_seek_result = ((QIODevice*)stream)->size() - offset;
270         break;
271     case ZLIB_FILEFUNC_SEEK_SET :
272         qiodevice_seek_result = offset;
273         break;
274     default:
275         return -1;
276     }
277     ret = !iodevice->seek(qiodevice_seek_result);
278     return ret;
279 }
```

Here is the caller graph for this function:



13.139.2.8 ZPOS64_T qiodevice64_tell_file_func (voidpf *opaque*, voidpf *stream*)

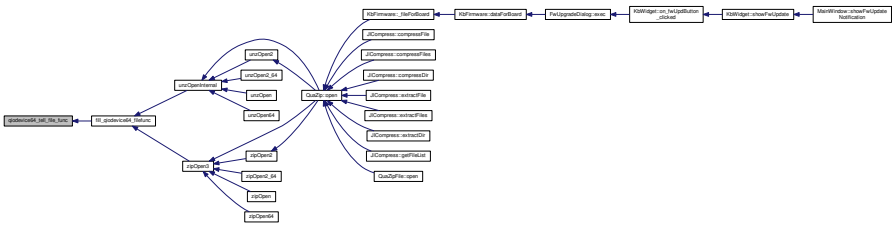
Definition at line 192 of file qioapi.cpp.

References QIODevice_descriptor::pos.

Referenced by fill_qiodevice64_filefunc().

```
195 {
196     QIODevice_descriptor *d = reinterpret_cast<
    QIODevice_descriptor*>(opaque);
197     QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
198     qint64 ret;
199     if (iodevice->isSequential()) {
200         ret = d->pos;
201     } else {
202         ret = iodevice->pos();
203     }
204     return static_cast<ZPOS64_T>(ret);
205 }
```

Here is the caller graph for this function:



13.139.2.9 int qiodevice_close_file_func (voidpf *opaque*, voidpf *stream*)

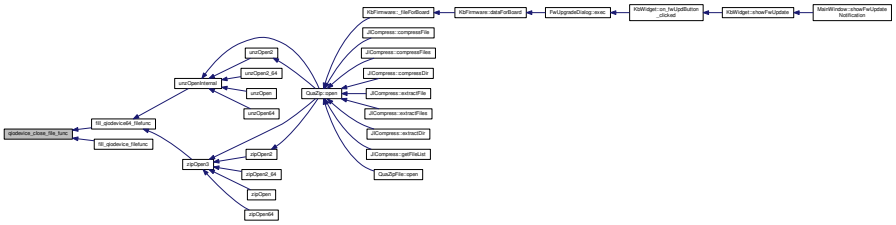
Definition at line 281 of file qioapi.cpp.

Referenced by fill_qiodevice64_filefunc(), and fill_qiodevice_filefunc().

```
284 {
285     QIODevice_descriptor *d = reinterpret_cast<
    QIODevice_descriptor*>(opaque);
286     delete d;
287     QIODevice *device = reinterpret_cast<QIODevice*>(stream);
288 #ifdef QUAZIP_QSAVEFILE_BUG_WORKAROUND
289     // QSaveFile terribly breaks the is-a idiom:
290     // it IS a QIODevice, but it is NOT compatible with it: close() is private
291     QSaveFile *file = qobject_cast<QSaveFile*>(device);
292     if (file != NULL) {
293         // We have to call the ugly commit() instead:
294         return file->commit() ? 0 : -1;
295     }
```

```
296 #endif
297     device->close();
298     return 0;
299 }
```

Here is the caller graph for this function:



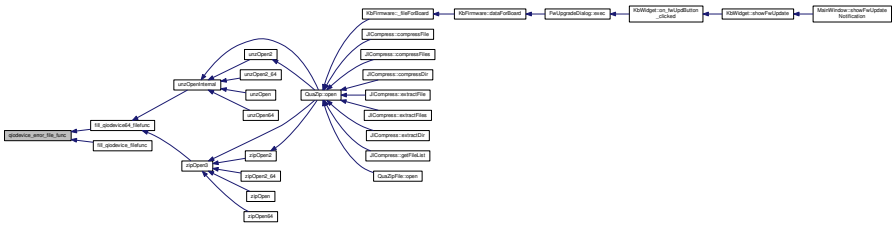
13.139.2.10 int qiodevice_error_file_func (voidpf , voidpf)

Definition at line 310 of file qioapi.cpp.

Referenced by fill_qiodevice64_filefunc(), and fill_qiodevice_filefunc().

```
313 {
314     // can't check for error due to the QIODevice API limitation
315     return 0;
316 }
```

Here is the caller graph for this function:



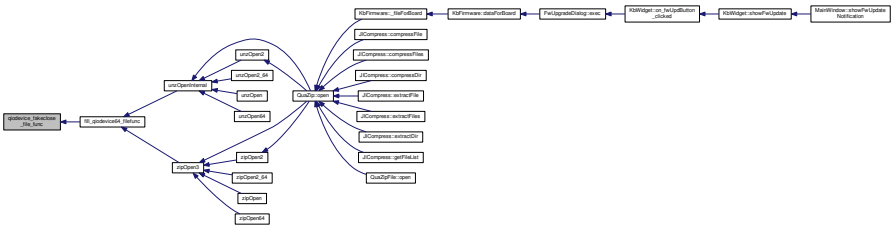
13.139.2.11 int qiodevice_fakeclose_file_func (voidpf *opaque,* voidpf)

Definition at line 301 of file qioapi.cpp.

Referenced by fill_qiodevice64_filefunc().

```
304 {
305     QIODevice_descriptor *d = reinterpret_cast<
    QIODevice_descriptor*>(opaque);
306     delete d;
307     return 0;
308 }
```

Here is the caller graph for this function:



13.139.2.12 voidpf qiodevice_open_file_func (voidpf *opaque,* voidpf *file,* int *mode*)

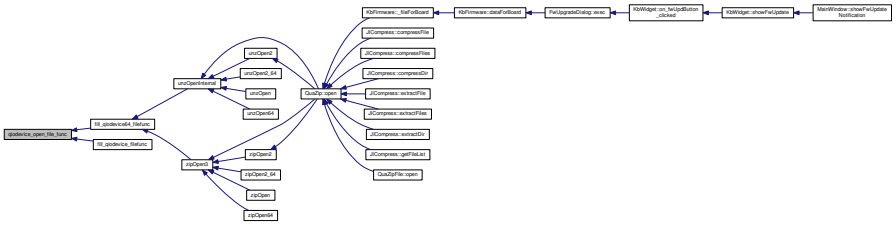
Definition at line 86 of file qioapi.cpp.

References QIODevice_descriptor::pos, ZLIB_FILEFUNC_MODE_CREATE, ZLIB_FILEFUNC_MODE_EXISTING, ZLIB_FILEFUNC_MODE_READ, and ZLIB_FILEFUNC_MODE_READWRITEFILTER.

Referenced by fill_qiodevice64_filefunc(), and fill_qiodevice_filefunc().

```
90 {
91     QIODevice_descriptor *d = reinterpret_cast<
      QIODevice_descriptor*>(opaque);
92     QIODevice *iodevice = reinterpret_cast<QIODevice*>(file);
93     QIODevice::OpenMode desiredMode;
94     if ((mode & ZLIB_FILEFUNC_MODE_READWRITEFILTER)==
      ZLIB_FILEFUNC_MODE_READ)
95         desiredMode = QIODevice::ReadOnly;
96     else if (mode & ZLIB_FILEFUNC_MODE_EXISTING)
97         desiredMode = QIODevice::ReadWrite;
98     else if (mode & ZLIB_FILEFUNC_MODE_CREATE)
99         desiredMode = QIODevice::WriteOnly;
100     if (iodevice->isOpen()) {
101         if ((iodevice->openMode() & desiredMode) == desiredMode) {
102             if (desiredMode != QIODevice::WriteOnly
103                     && iodevice->isSequential()) {
104                 // We can use sequential devices only for writing.
105                 delete d;
106                 return NULL;
107             } else {
108                 if ((desiredMode & QIODevice::WriteOnly) != 0) {
109                     // open for writing, need to seek existing device
110                     if (!iodevice->isSequential()) {
111                         iodevice->seek(0);
112                     } else {
113                         d->pos = iodevice->pos();
114                     }
115                 }
116             }
117             return iodevice;
118         } else {
119             delete d;
120             return NULL;
121         }
122     }
123     iodevice->open(desiredMode);
124     if (iodevice->isOpen()) {
125         if (desiredMode != QIODevice::WriteOnly && iodevice->isSequential()) {
126             // We can use sequential devices only for writing.
127             iodevice->close();
128             delete d;
129             return NULL;
130         } else {
131             return iodevice;
132         }
133     } else {
134         delete d;
135         return NULL;
136     }
137 }
```

Here is the caller graph for this function:



13.139.2.13 uLong qiodevice_read_file_func (voidpf *opaque*, voidpf *stream*, void * *buf*, uLong *size*)

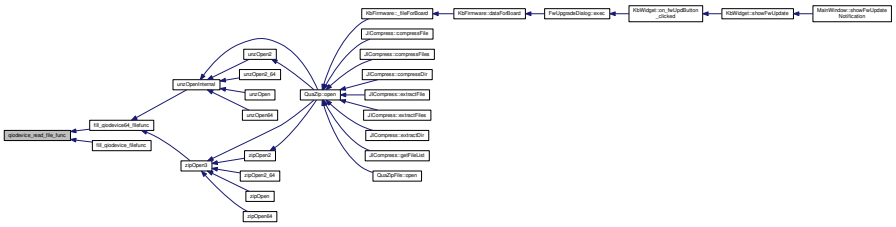
Definition at line 140 of file qioapi.cpp.

References QIODevice_descriptor::pos.

Referenced by fill_qiodevice64_filefunc(), and fill_qiodevice_filefunc().

```
145 {
146     QIODevice_descriptor *d = reinterpret_cast<
    QIODevice_descriptor*>(opaque);
147     QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
148     qint64 ret64 = iodevice->read((char*)buf,size);
149     uLong ret;
150     ret = (uLong) ret64;
151     if (ret64 != -1) {
152         d->pos += ret64;
153     }
154     return ret;
155 }
```

Here is the caller graph for this function:



13.139.2.14 int qiodevice_seek_file_func (voidpf , voidpf *stream*, uLong *offset*, int *origin*)

Definition at line 207 of file qioapi.cpp.

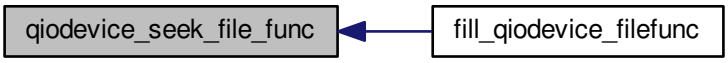
References ZLIB_FILEFUNC_SEEK_CUR, ZLIB_FILEFUNC_SEEK_END, and ZLIB_FILEFUNC_SEEK_SET.

Referenced by fill_qiodevice_filefunc().

```
212 {
213     QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
214     if (iodevice->isSequential()) {
215         if (origin == ZLIB_FILEFUNC_SEEK_END
216                 && offset == 0) {
217             // sequential devices are always at end (needed in mdAppend)
218             return 0;
219         } else {
220             qWarning("qiodevice_seek_file_func() called for sequential device");
221             return -1;
```

```
222         }
223     }
224     uLong qiodevice_seek_result=0;
225     int ret;
226     switch (origin)
227     {
228     case ZLIB_FILEFUNC_SEEK_CUR :
229         qiodevice_seek_result = ((QIODevice*)stream)->pos() + offset;
230         break;
231     case ZLIB_FILEFUNC_SEEK_END :
232         qiodevice_seek_result = ((QIODevice*)stream)->size() - offset;
233         break;
234     case ZLIB_FILEFUNC_SEEK_SET :
235         qiodevice_seek_result = offset;
236         break;
237     default:
238         return -1;
239     }
240     ret = !iodevice->seek(qiodevice_seek_result);
241     return ret;
242 }
```

Here is the caller graph for this function:



13.139.2.15 uLong qiodevice_tell_file_func (voidpf *opaque*, voidpf *stream*)

Definition at line 175 of file qioapi.cpp.

References QIODevice_descriptor::pos.

Referenced by fill_qiodevice_filefunc().

```
178 {
179     QIODevice_descriptor *d = reinterpret_cast<
      QIODevice_descriptor*>(opaque);
180     QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
181     uLong ret;
182     qint64 ret64;
183     if (iodevice->isSequential()) {
184         ret64 = d->pos;
185     } else {
186         ret64 = iodevice->pos();
187     }
188     ret = static_cast<uLong>(ret64);
189     return ret;
190 }
```

Here is the caller graph for this function:

13.139.2.16 uLong qiodevice_write_file_func (voidpf *opaque*, voidpf *stream*, const void * *buf*, uLong *size*)

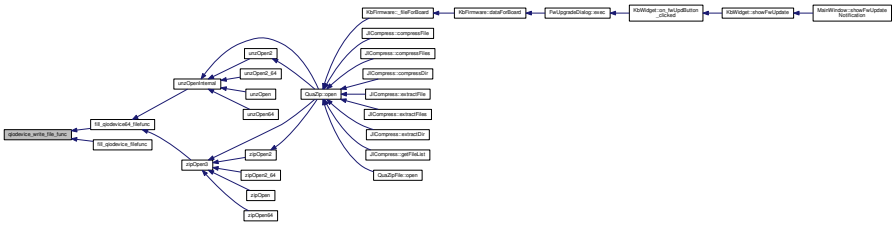
Definition at line 158 of file qioapi.cpp.

References QIODevice_descriptor::pos.

Referenced by fill_qiodevice64_filefunc(), and fill_qiodevice_filefunc().

```
163 {
164     QIODevice_descriptor *d = reinterpret_cast<
      QIODevice_descriptor*>(opaque);
165     QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
166     uLong ret;
167     qint64 ret64 = iodevice->write((char*)buf,size);
168     if (ret64 != -1) {
169         d->pos += ret64;
170     }
171     ret = (uLong) ret64;
172     return ret;
173 }
```

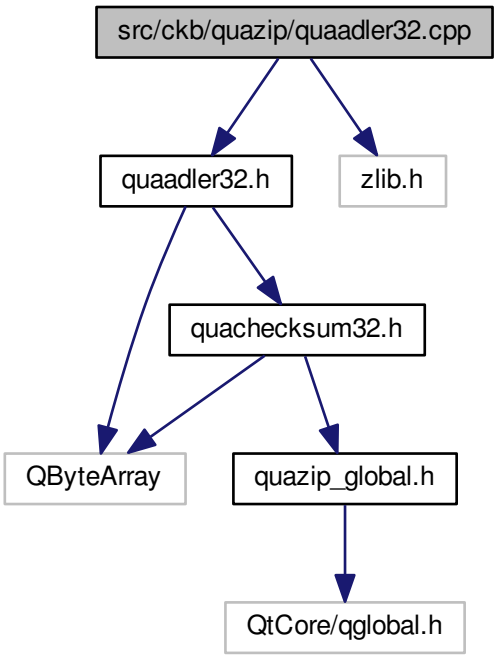
Here is the caller graph for this function:



13.140 src/ckb/quazip/quaadler32.cpp File Reference

```
#include "quaadler32.h"
#include "zlib.h"
```

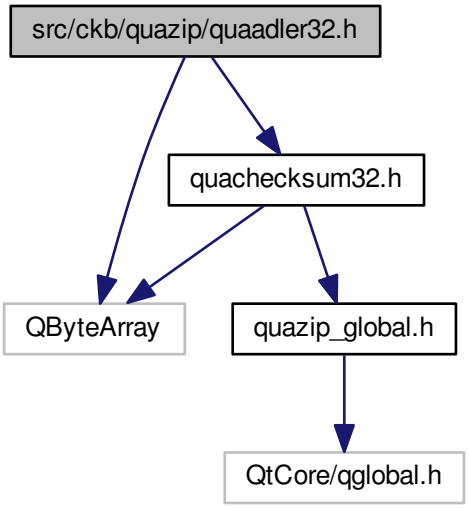
Include dependency graph for quaadler32.cpp:



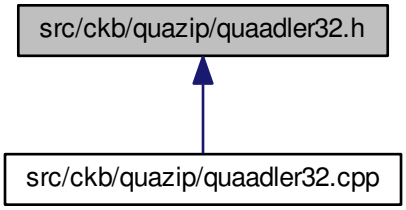
13.141 src/ckb/quazip/quaadler32.h File Reference

```
#include <QByteArray>
#include "quachecksum32.h"
```

Include dependency graph for quaadler32.h:



This graph shows which files directly or indirectly include this file:



Data Structures

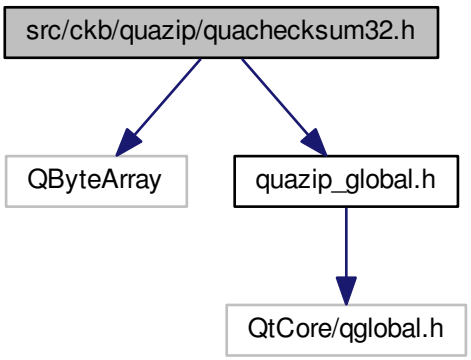
- class QuaAdler32

  Adler32 checksum.

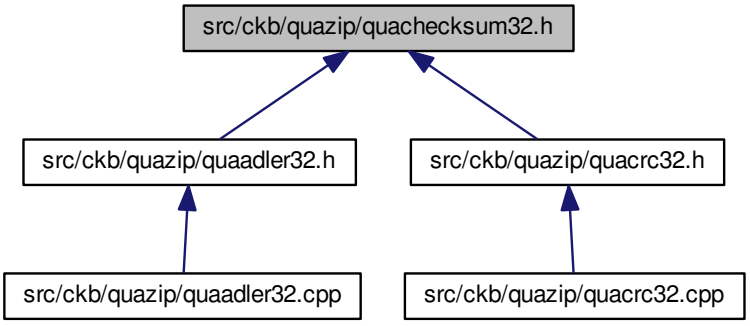
13.142 src/ckb/quazip/quachecksum32.h File Reference

```
#include <QByteArray>
#include "quazip_global.h"
```

Include dependency graph for quachecksum32.h:



This graph shows which files directly or indirectly include this file:



Data Structures

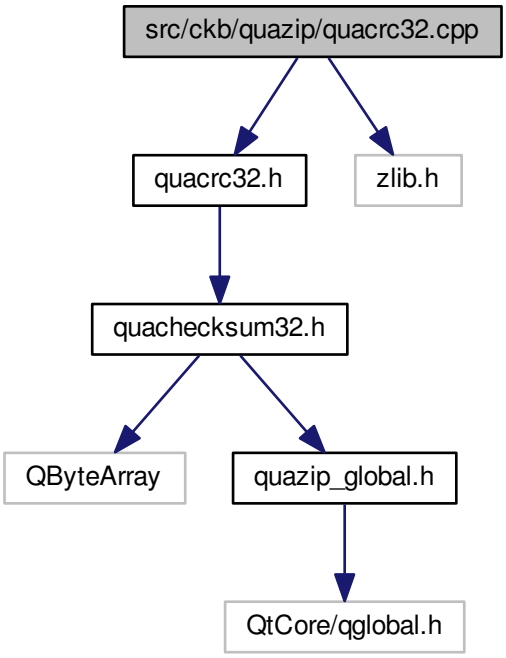
- class QuaChecksum32

  Checksum interface.

13.143 src/ckb/quazip/quacrc32.cpp File Reference

```
#include "quacrc32.h"
#include "zlib.h"
```

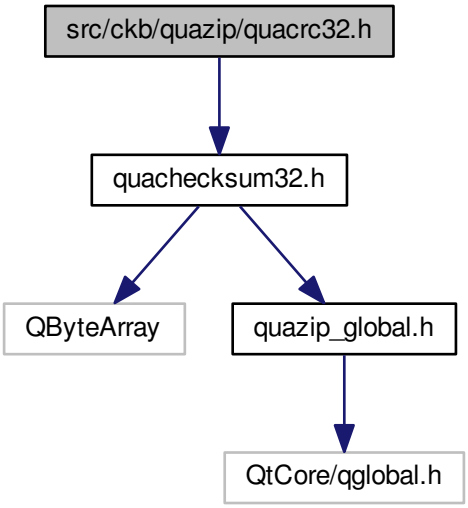
Include dependency graph for quacrc32.cpp:



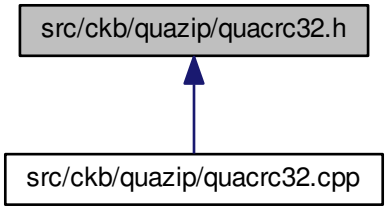
13.144 src/ckb/quazip/quacrc32.h File Reference

```
#include "quachecksum32.h"
```

Include dependency graph for quacrc32.h:



This graph shows which files directly or indirectly include this file:



Data Structures

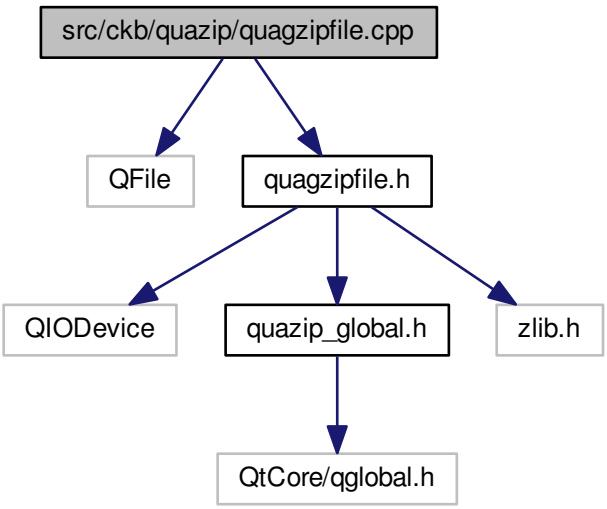
- class QuaCrc32

  CRC32 checksum.

13.145 src/ckb/quazip/quagzipfile.cpp File Reference

```
#include <QFile>
#include "quagzipfile.h"
```

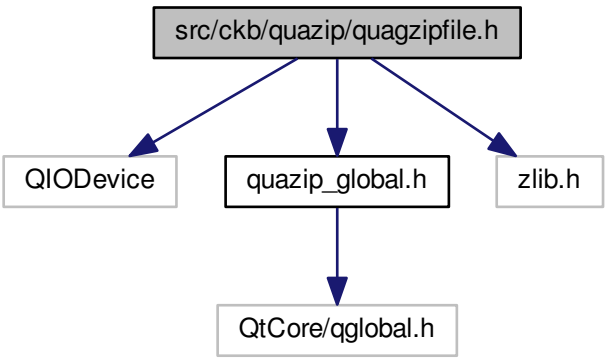
Include dependency graph for quagzipfile.cpp:



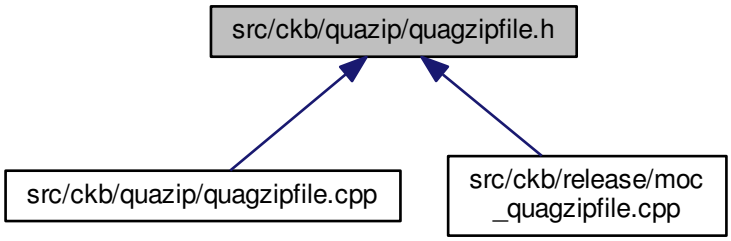
13.146 src/ckb/quazip/quagzipfile.h File Reference

```
#include <QIODevice>
#include "quazip_global.h"
#include <zlib.h>
```

Include dependency graph for quagzipfile.h:



This graph shows which files directly or indirectly include this file:



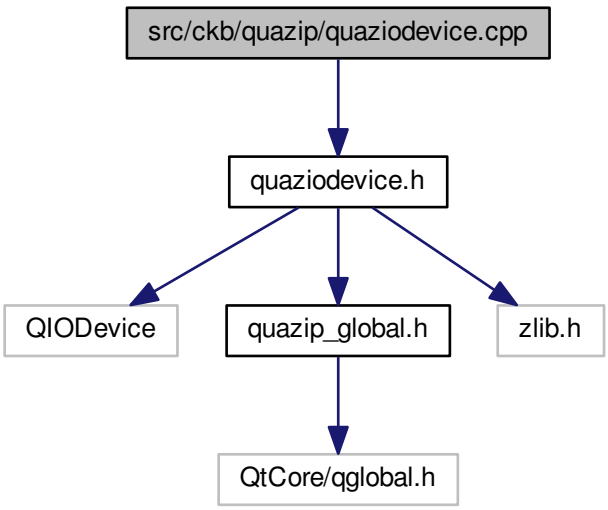
Data Structures

- class QuaGzipFile
  GZIP file.

13.147 src/ckb/quazip/quaziodevice.cpp File Reference

```
#include "quaziodevice.h"
```

Include dependency graph for quaziodevice.cpp:



Macros

- #define QUAZIO_INBUFSIZE 4096

- #define QUAZIO_OUTBUFSIZE 4096

13.147.1 Macro Definition Documentation

13.147.1.1 #define QUAZIO_INBUFSIZE 4096

Definition at line 27 of file quaziodevice.cpp.

Referenced by QuaZIODevice::readData().

13.147.1.2 #define QUAZIO_OUTBUFSIZE 4096

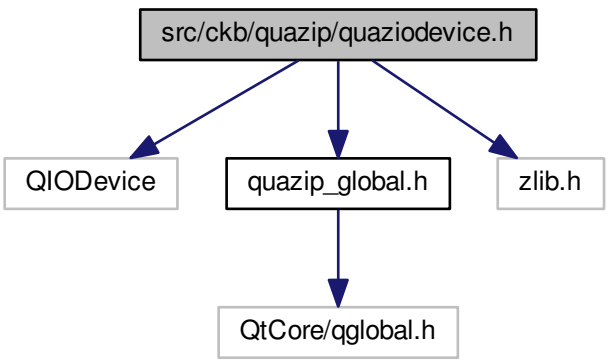
Definition at line 28 of file quaziodevice.cpp.

Referenced by QuaZIODevice::flush(), and QuaZIODevice::writeData().

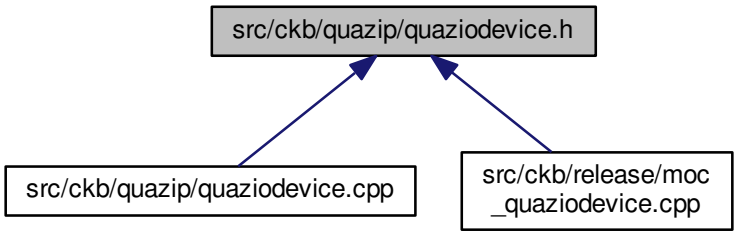
13.148 src/ckb/quazip/quaziodevice.h File Reference

```
#include <QIODevice>
#include "quazip_global.h"
#include <zlib.h>
```

Include dependency graph for quaziodevice.h:



This graph shows which files directly or indirectly include this file:



Data Structures

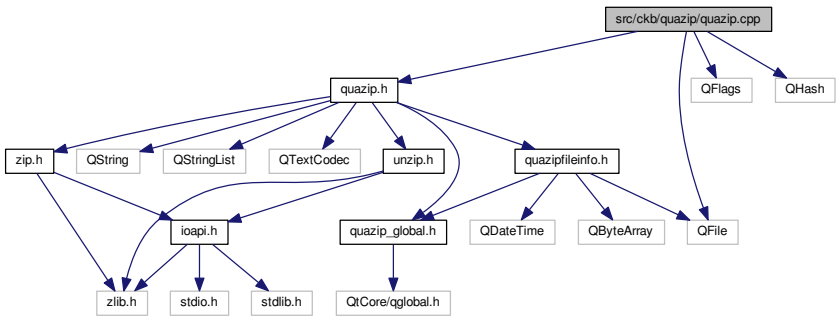
- class QuaZIODevice

  A class to compress/decompress QIODevice.

13.149 src/ckb/quazip/quazip.cpp File Reference

```
#include <QFile>
#include <QFlags>
#include <QHash>
#include "quazip.h"
```

Include dependency graph for quazip.cpp:



Data Structures

- class QuaZipPrivate

  All the internal stuff for the QuaZip class.

Functions

- template<typename TFileInfo >
  TFileInfo QuaZip_getFileInfo (QuaZip *zip, bool *ok)

- template<>
  QuaZipFileInfo QuaZip_getFileInfo (QuaZip *zip, bool *ok)
- template<>
  QuaZipFileInfo64 QuaZip_getFileInfo (QuaZip *zip, bool *ok)
- template<>
  QString QuaZip_getFileInfo (QuaZip *zip, bool *ok)

13.149.1 Function Documentation

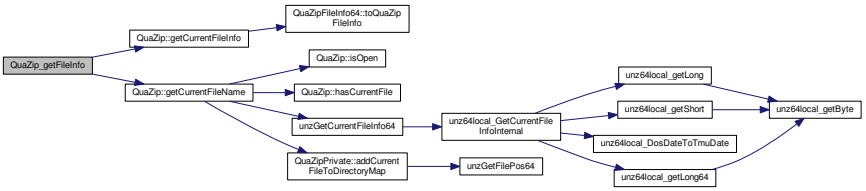
13.149.1.1 template<typename TFileInfo > TFileInfo QuaZip_getFileInfo (QuaZip * *zip*, bool * *ok*)

Definition at line 663 of file quazip.cpp.

References QuaZip::getCurrentFileInfo(), and QuaZip::getCurrentFileName().

```
664 {
665     QuaZipFileInfo info;
666     *ok = zip->getCurrentFileInfo(&info);
667     return info;
668 }
```

Here is the call graph for this function:



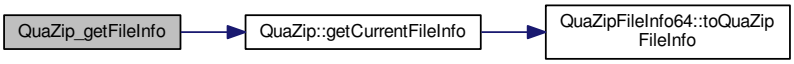
13.149.1.2 template<> QuaZipFileInfo QuaZip_getFileInfo (QuaZip * *zip*, bool * *ok*)

Definition at line 663 of file quazip.cpp.

References QuaZip::getCurrentFileInfo().

```
664 {
665     QuaZipFileInfo info;
666     *ok = zip->getCurrentFileInfo(&info);
667     return info;
668 }
```

Here is the call graph for this function:



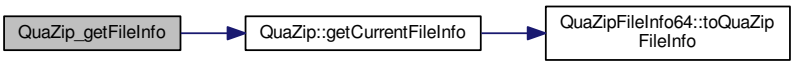
13.149.1.3 template<> QuaZipFileInfo64 QuaZip_getFileInfo (QuaZip * *zip*, bool * *ok*)

Definition at line 671 of file quazip.cpp.

References QuaZip::getCurrentFileInfo().

```
672 {
673     QuaZipFileInfo64 info;
674     *ok = zip->getCurrentFileInfo(&info);
675     return info;
676 }
```

Here is the call graph for this function:



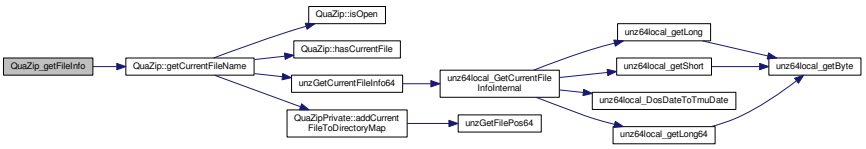
13.149.1.4 template<> QString QuaZip_getFileInfo (QuaZip * *zip*, bool * *ok*)

Definition at line 679 of file quazip.cpp.

References QuaZip::getCurrentFileName().

```
680 {
681     QString name = zip->getCurrentFileName();
682     *ok = !name.isEmpty();
683     return name;
684 }
```

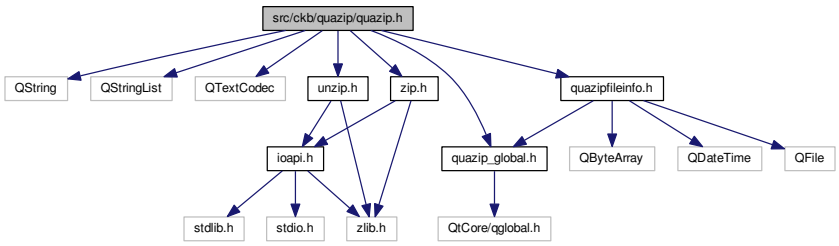
Here is the call graph for this function:



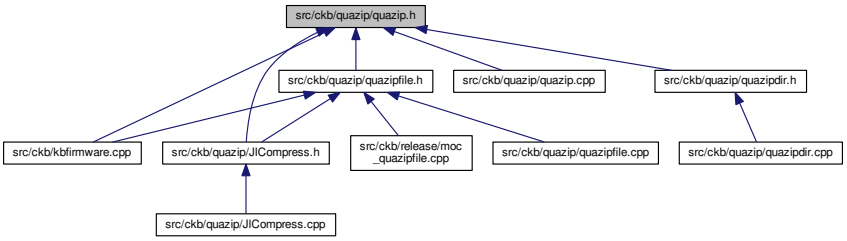
13.150 src/ckb/quazip/quazip.h File Reference

```
#include <QString>
#include <QStringList>
#include <QTextCodec>
#include "zip.h"
#include "unzip.h"
#include "quazip_global.h"
#include "quazipfileinfo.h"
```

Include dependency graph for quazip.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class QuaZip

  ZIP archive.

Macros

- #define UNZ_OPENERROR -1000

13.150.1 Macro Definition Documentation

13.150.1.1 #define UNZ_OPENERROR -1000

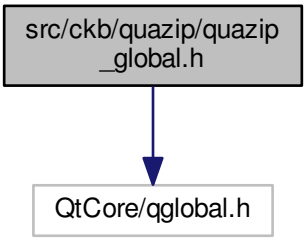
Definition at line 41 of file quazip.h.

Referenced by QuaZip::open().

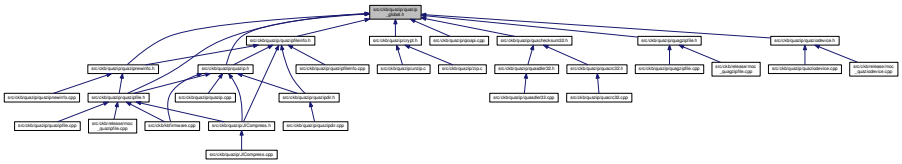
13.151 src/ckb/quazip/quazip_global.h File Reference

```
#include <QtCore/qglobal.h>
```

Include dependency graph for quazip_global.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define QUAZIP_EXPORT Q_DECL_IMPORT

  This is automatically defined when building a static library, but when including QuaZip sources directly into a project, QUAZIP_STATIC should be defined explicitly to avoid possible troubles with unnecessary importing/exporting.
- #define UNUSED
- #define QUAZIP_EXTRA_NTFS_MAGIC 0x000Au
- #define QUAZIP_EXTRA_NTFS_TIME_MAGIC 0x0001u

13.151.1 Macro Definition Documentation

13.151.1.1 #define QUAZIP_EXPORT Q_DECL_IMPORT

When building a DLL with MSVC, QUAZIP_BUILD must be defined. qglobal.h takes care of defining Q_DECL_∗ correctly for msvc/gcc.

Definition at line 46 of file quazip_global.h.

13.151.1.2 #define QUAZIP_EXTRA_NTFS_MAGIC 0x000Au

Definition at line 56 of file quazip_global.h.

Referenced by getNTFSTime(), and setNTFSTime().

13.151.1.3 #define QUAZIP_EXTRA_NTFS_TIME_MAGIC 0x0001u

Definition at line 57 of file quazip_global.h.

Referenced by getNTFSTime(), and setNTFSTime().

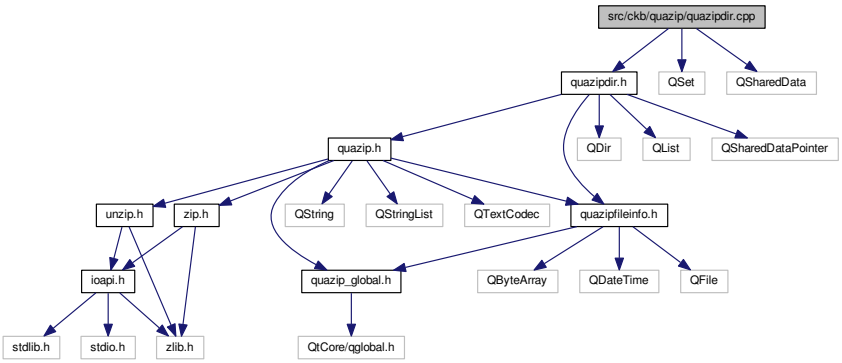
13.151.1.4 #define UNUSED

Definition at line 53 of file quazip_global.h.

13.152 src/ckb/quazip/quazipdir.cpp File Reference

```
#include "quazipdir.h"
#include <QSet>
#include <QSharedData>
```

Include dependency graph for quazipdir.cpp:



Functions

- QuaZipFileInfo64 QuaZipDir_getFileInfo (QuaZip *zip, bool *ok, const QString &relativeName, bool isReal)
- static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > &from, QList< QuaZipFileInfo64 > &to)
- static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > &from, QStringList &to)
- static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > &from, QList< QuaZipFileInfo > &to)

13.152.1 Function Documentation

13.152.1.1 static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > & *from*, QList< QuaZipFileInfo64 > & *to*) [static]

Definition at line 188 of file quazipdir.cpp.

```
190 {
191     to = from;
192 }
```

13.152.1.2 static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > & *from*, QStringList & *to*) [static]

Definition at line 194 of file quazipdir.cpp.

```
196 {
197     to.clear();
198     for (QList<QuaZipFileInfo64>::const_iterator i = from.constBegin();
199             i != from.constEnd();
200             ++i) {
201         to.append(i->name);
202     }
203 }
```

13.152.1.3 static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > & *from,* QList< QuaZipFileInfo > & *to*) [static]

Definition at line 205 of file quazipdir.cpp.

```
207 {
208     to.clear();
209     for (QList<QuaZipFileInfo64>::const_iterator i = from.constBegin();
210             i != from.constEnd();
211             ++i) {
212         QuaZipFileInfo info32;
213         i->toQuaZipFileInfo(info32);
214         to.append(info32);
215     }
216 }
```

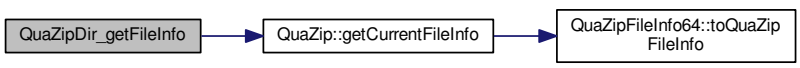
13.152.1.4 QuaZipFileInfo64 QuaZipDir_getFileInfo (QuaZip * *zip,* bool * *ok,* const QString & *relativeName,* bool *isReal*)

Definition at line 165 of file quazipdir.cpp.

References QuaZipFileInfo64::compressedSize, QuaZipFileInfo64::crc, QuaZipFileInfo64::diskNumberStart, QuaZipFileInfo64::externalAttr, QuaZipFileInfo64::flags, QuaZip::getCurrentFileInfo(), QuaZipFileInfo64::internalAttr, QuaZipFileInfo64::method, QuaZipFileInfo64::name, QuaZipFileInfo64::uncompressedSize, QuaZipFileInfo64::versionCreated, and QuaZipFileInfo64::versionNeeded.

```
168 {
169     QuaZipFileInfo64 info;
170     if (isReal) {
171         *ok = zip->getCurrentFileInfo(&info);
172     } else {
173         *ok = true;
174         info.compressedSize = 0;
175         info.crc = 0;
176         info.diskNumberStart = 0;
177         info.externalAttr = 0;
178         info.flags = 0;
179         info.internalAttr = 0;
180         info.method = 0;
181         info.uncompressedSize = 0;
182         info.versionCreated = info.versionNeeded = 0;
183     }
184     info.name = relativeName;
185     return info;
186 }
```

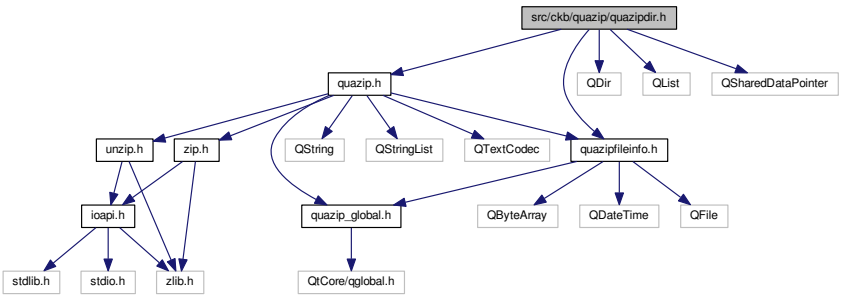
Here is the call graph for this function:



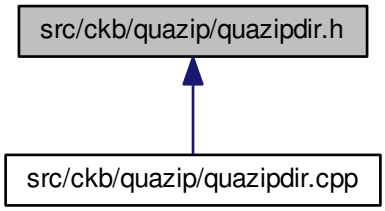
13.153 src/ckb/quazip/quazipdir.h File Reference

```
#include "quazip.h"
#include "quazipfileinfo.h"
#include <QDir>
#include <QList>
#include <QSharedDataPointer>
```

Include dependency graph for quazipdir.h:



This graph shows which files directly or indirectly include this file:



Data Structures

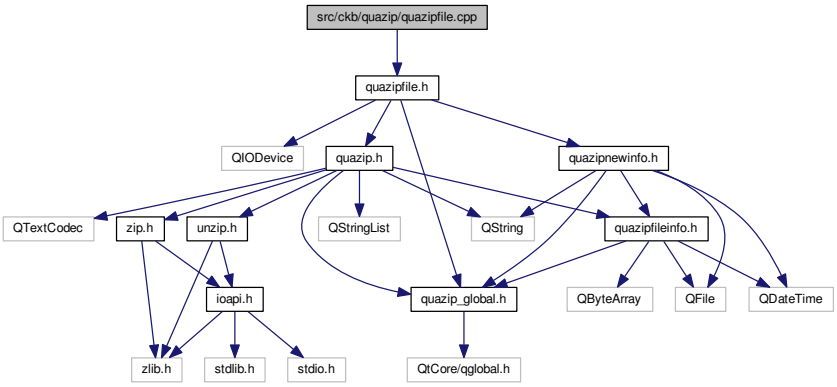
- class QuaZipDir

  Provides ZIP archive navigation.

13.154 src/ckb/quazip/quazipfile.cpp File Reference

```
#include "quazipfile.h"
```

Include dependency graph for quazipfile.cpp:



Data Structures

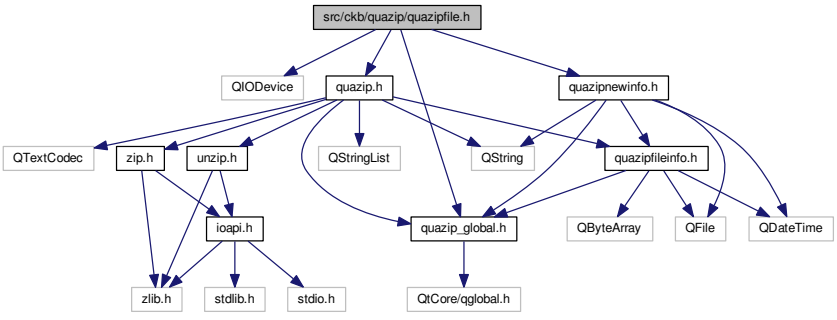
- class QuaZipFilePrivate

  The implementation class for QuaZip.

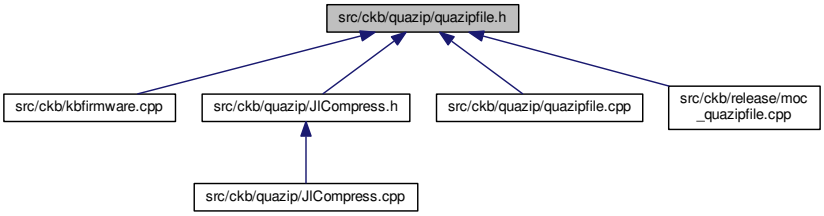
13.155 src/ckb/quazip/quazipfile.h File Reference

```
#include <QIODevice>
#include "quazip_global.h"
#include "quazip.h"
#include "quazipnewinfo.h"
```

Include dependency graph for quazipfile.h:



This graph shows which files directly or indirectly include this file:



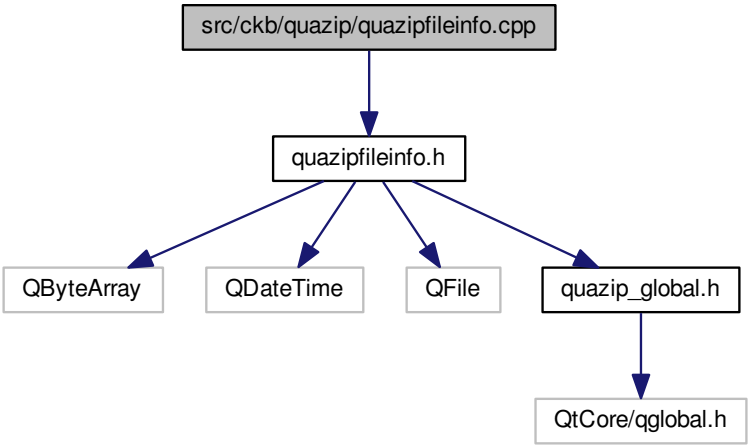
Data Structures

- class QuaZipFile

  A file inside ZIP archive.

13.156 src/ckb/quazip/quazipfileinfo.cpp File Reference

`#include "quazipfileinfo.h"`
Include dependency graph for quazipfileinfo.cpp:



Functions

- static QFile::Permissions permissionsFromExternalAttr (quint32 externalAttr)
- static QDateTime getNTFSTime (const QByteArray &extra, int position, int *fineTicks)

13.156.1 Function Documentation

13.156.1.1 static QDateTime getNTFSTime (const QByteArray & *extra*, int *position*, int * *fineTicks*) [static]

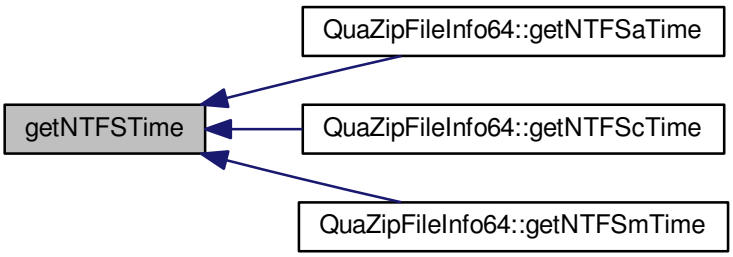
Definition at line 92 of file quazipfileinfo.cpp.

References QUAZIP_EXTRA_NTFS_MAGIC, and QUAZIP_EXTRA_NTFS_TIME_MAGIC.

Referenced by QuaZipFileInfo64::getNTFSaTime(), QuaZipFileInfo64::getNTFScTime(), and QuaZipFileInfo64::getNTFSmTime().

```
94 {
95     QDateTime dateTime;
96     for (int i = 0; i <= extra.size() - 4; ) {
97         unsigned type = static_cast<unsigned>(static_cast<unsigned char>(
98                                                   extra.at(i)))
99                 | (static_cast<unsigned>(static_cast<unsigned char>(
100                                                    extra.at(i + 1))) << 8);
101         i += 2;
102         unsigned length = static_cast<unsigned>(static_cast<unsigned char>(
103                                                    extra.at(i)))
104                 | (static_cast<unsigned>(static_cast<unsigned char>(
105                                                    extra.at(i + 1))) << 8);
106         i += 2;
107         if (type == QUAZIP_EXTRA_NTFS_MAGIC && length >= 32) {
108             i += 4; // reserved
109             while (i <= extra.size() - 4) {
110                 unsigned tag = static_cast<unsigned>(
111                             static_cast<unsigned char>(extra.at(i)))
112                         | (static_cast<unsigned>(
113                                 static_cast<unsigned char>(extra.at(i + 1)))
114                             << 8);
115                 i += 2;
116                 int tagsize = static_cast<unsigned>(
117                             static_cast<unsigned char>(extra.at(i)))
118                         | (static_cast<unsigned>(
119                                 static_cast<unsigned char>(extra.at(i + 1)))
120                             << 8);
121                 i += 2;
122                 if (tag == QUAZIP_EXTRA_NTFS_TIME_MAGIC
123                         && tagsize >= position + 8) {
124                     i += position;
125                     quint64 mtime = static_cast<quint64>(
126                                 static_cast<unsigned char>(extra.at(i)))
127                         | (static_cast<quint64>(static_cast<unsigned char>(
128                                                    extra.at(i + 1))) << 8)
129                         | (static_cast<quint64>(static_cast<unsigned char>(
130                                                    extra.at(i + 2))) << 16)
131                         | (static_cast<quint64>(static_cast<unsigned char>(
132                                                    extra.at(i + 3))) << 24)
133                         | (static_cast<quint64>(static_cast<unsigned char>(
134                                                    extra.at(i + 4))) << 32)
135                         | (static_cast<quint64>(static_cast<unsigned char>(
136                                                    extra.at(i + 5))) << 40)
137                         | (static_cast<quint64>(static_cast<unsigned char>(
138                                                    extra.at(i + 6))) << 48)
139                         | (static_cast<quint64>(static_cast<unsigned char>(
140                                                    extra.at(i + 7))) << 56);
141                     // the NTFS time is measured from 1601 for whatever reason
142                     QDateTime base(QDate(1601, 1, 1), QTime(0, 0), Qt::UTC);
143                     dateTime = base.addMSecs(mtime / 10000);
144                     if (fineTicks != NULL) {
145                         *fineTicks = static_cast<int>(mtime % 10000);
146                     }
147                     i += tagsize - position;
148                 } else {
149                     i += tagsize;
150                 }
151 
152             }
153         } else {
154             i += length;
155         }
156     }
157     if (fineTicks != NULL && dateTime.isNull()) {
158         *fineTicks = 0;
159     }
160     return dateTime;
161 }
```

Here is the caller graph for this function:



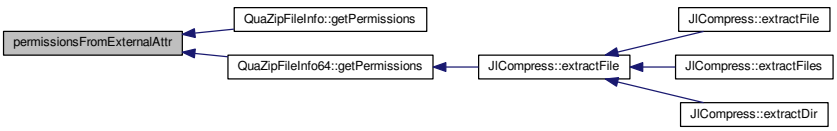
13.156.1.2 static QFile::Permissions permissionsFromExternalAttr (quint32 *externalAttr*) [static]

Definition at line 27 of file quazipfileinfo.cpp.

Referenced by QuaZipFileInfo::getPermissions(), and QuaZipFileInfo64::getPermissions().

```
27                                                                          {
28     quint32 uPerm = (externalAttr & 0xFFFF0000u) >> 16;
29     QFile::Permissions perm = 0;
30     if ((uPerm & 0400) != 0)
31         perm |= QFile::ReadOwner;
32     if ((uPerm & 0200) != 0)
33         perm |= QFile::WriteOwner;
34     if ((uPerm & 0100) != 0)
35         perm |= QFile::ExeOwner;
36     if ((uPerm & 0040) != 0)
37         perm |= QFile::ReadGroup;
38     if ((uPerm & 0020) != 0)
39         perm |= QFile::WriteGroup;
40     if ((uPerm & 0010) != 0)
41         perm |= QFile::ExeGroup;
42     if ((uPerm & 0004) != 0)
43         perm |= QFile::ReadOther;
44     if ((uPerm & 0002) != 0)
45         perm |= QFile::WriteOther;
46     if ((uPerm & 0001) != 0)
47         perm |= QFile::ExeOther;
48     return perm;
49
50 }
```

Here is the caller graph for this function:

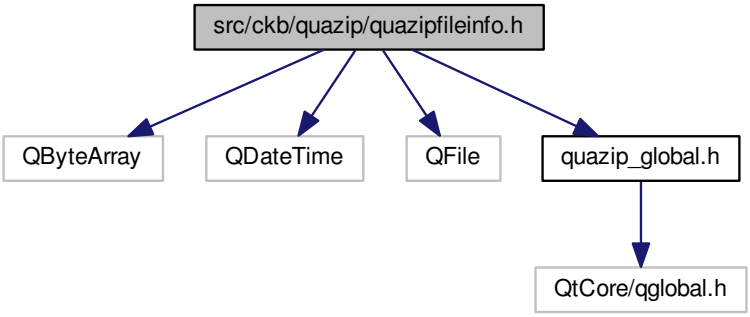


13.157 src/ckb/quazip/quazipfileinfo.h File Reference

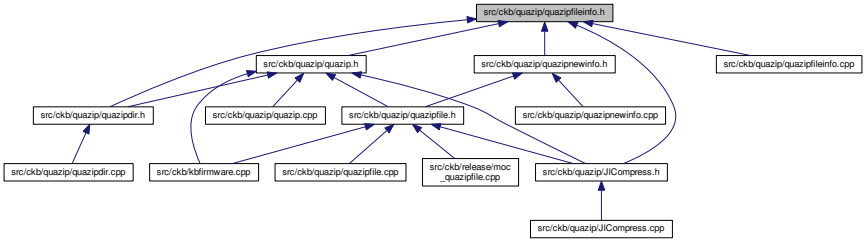
```
#include <QByteArray>
```

```
#include <QDateTime>
#include <QFile>
#include "quazip_global.h"
```

Include dependency graph for quazipfileinfo.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct QuaZipFileInfo

  Information about a file inside archive.

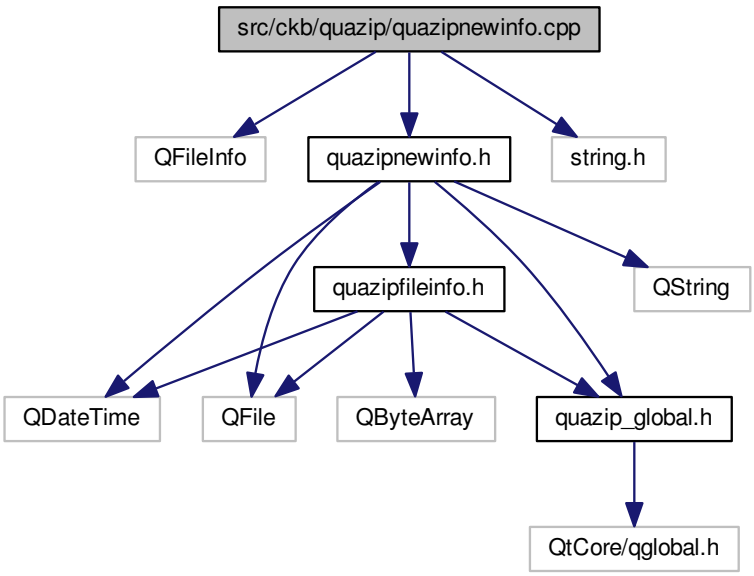
- struct QuaZipFileInfo64

  Information about a file inside archive (with zip64 support).

13.158 src/ckb/quazip/quazipnewinfo.cpp File Reference

```
#include <QFileInfo>
#include "quazipnewinfo.h"
#include <string.h>
```

Include dependency graph for quazipnewinfo.cpp:



Functions

- static void QuaZipNewInfo_setPermissions (QuaZipNewInfo *info, QFile::Permissions perm, bool isDir)
- template<typename FileInfo >
  void QuaZipNewInfo_init (QuaZipNewInfo &self, const FileInfo &existing)
- static void setNTFSTime (QByteArray &extra, const QDateTime &time, int position, int fineTicks)

13.158.1 Function Documentation

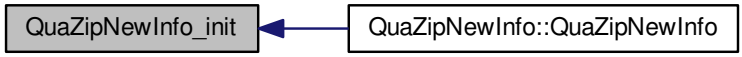
13.158.1.1 template<typename FileInfo > void QuaZipNewInfo_init (QuaZipNewInfo & *self*, const FileInfo & *existing*)

Definition at line 57 of file quazipnewinfo.cpp.

Referenced by QuaZipNewInfo::QuaZipNewInfo().

```
58 {
59     self.name = existing.name;
60     self.dateTime = existing.dateTime;
61     self.internalAttr = existing.internalAttr;
62     self.externalAttr = existing.externalAttr;
63     self.comment = existing.comment;
64     self.extraLocal = existing.extra;
65     self.extraGlobal = existing.extra;
66     self.uncompressedSize = existing.uncompressedSize;
67 }
```

Here is the caller graph for this function:



13.158.1.2 static void QuaZipNewInfo_setPermissions (QuaZipNewInfo * *info*, QFile::Permissions *perm*, bool *isDir*) [static]

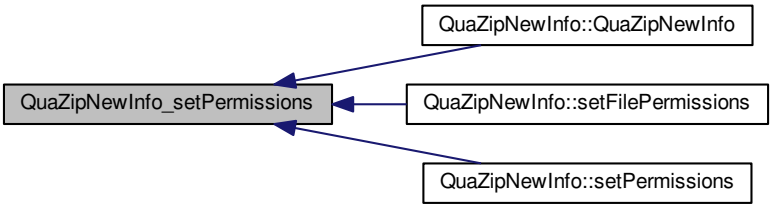
Definition at line 31 of file quazipnewinfo.cpp.

References QuaZipNewInfo::externalAttr.

Referenced by QuaZipNewInfo::QuaZipNewInfo(), QuaZipNewInfo::setFilePermissions(), and QuaZipNewInfo::setPermissions().

```
33 {
34     quint32 uPerm = isDir ? 0040000 : 0100000;
35     if ((perm & QFile::ReadOwner) != 0)
36         uPerm |= 0400;
37     if ((perm & QFile::WriteOwner) != 0)
38         uPerm |= 0200;
39     if ((perm & QFile::ExeOwner) != 0)
40         uPerm |= 0100;
41     if ((perm & QFile::ReadGroup) != 0)
42         uPerm |= 0040;
43     if ((perm & QFile::WriteGroup) != 0)
44         uPerm |= 0020;
45     if ((perm & QFile::ExeGroup) != 0)
46         uPerm |= 0010;
47     if ((perm & QFile::ReadOther) != 0)
48         uPerm |= 0004;
49     if ((perm & QFile::WriteOther) != 0)
50         uPerm |= 0002;
51     if ((perm & QFile::ExeOther) != 0)
52         uPerm |= 0001;
53     info->externalAttr = (info->externalAttr & ~0xFFFF0000u) | (uPerm << 16);
54 }
```

Here is the caller graph for this function:



13.158.1.3 static void setNTFSTime (QByteArray & *extra*, const QDateTime & *time*, int *position*, int *fineTicks*) [static]

Definition at line 131 of file quazipnewinfo.cpp.

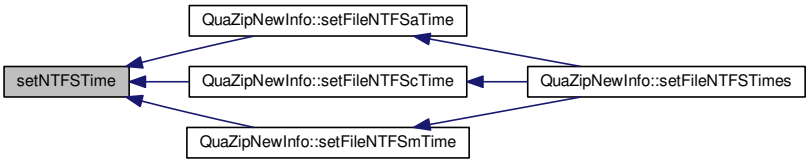
References QUAZIP_EXTRA_NTFS_MAGIC, and QUAZIP_EXTRA_NTFS_TIME_MAGIC.

Referenced by QuaZipNewInfo::setFileNTFSaTime(), QuaZipNewInfo::setFileNTFScTime(), and QuaZipNewInfo::setFileNTFSmTime().

```
132                                     {
133     int ntfsPos = -1, timesPos = -1;
134     unsigned ntfsLength = 0, ntfsTimesLength = 0;
135     for (int i = 0; i <= extra.size() - 4; ) {
136         unsigned type = static_cast<unsigned>(static_cast<unsigned char>(
137                                                 extra.at(i)))
138                 | (static_cast<unsigned>(static_cast<unsigned char>(
139                                                 extra.at(i + 1))) << 8);
140         i += 2;
141         unsigned length = static_cast<unsigned>(static_cast<unsigned char>(
142                                                 extra.at(i)))
143                 | (static_cast<unsigned>(static_cast<unsigned char>(
144                                                 extra.at(i + 1))) << 8);
145         i += 2;
146         if (type == QUAZIP_EXTRA_NTFS_MAGIC) {
147             ntfsPos = i - 4; // the beginning of the NTFS record
148             ntfsLength = length;
149             if (length <= 4) {
150                 break; // no times in the NTFS record
151             }
152             i += 4; // reserved
153             while (i <= extra.size() - 4) {
154                 unsigned tag = static_cast<unsigned>(
155                             static_cast<unsigned char>(extra.at(i)))
156                         | (static_cast<unsigned>(
157                                 static_cast<unsigned char>(extra.at(i + 1)))
158                             << 8);
159                 i += 2;
160                 unsigned tagsize = static_cast<unsigned>(
161                             static_cast<unsigned char>(extra.at(i)))
162                         | (static_cast<unsigned>(
163                                 static_cast<unsigned char>(extra.at(i + 1)))
164                             << 8);
165                 i += 2;
166                 if (tag == QUAZIP_EXTRA_NTFS_TIME_MAGIC) {
167                     timesPos = i - 4; // the beginning of the NTFS times tag
168                     ntfsTimesLength = tagsize;
169                     break;
170                 } else {
171                     i += tagsize;
172                 }
173             }
174             break; // I ain't going to search for yet another NTFS record!
175         } else {
176             i += length;
177         }
178     }
179     if (ntfsPos == -1) {
180         // No NTFS record, need to create one.
181         ntfsPos = extra.size();
182         ntfsLength = 32;
183         extra.resize(extra.size() + 4 + ntfsLength);
184         // the NTFS record header
185         extra[ntfsPos] = static_cast<char>(QUAZIP_EXTRA_NTFS_MAGIC);
186         extra[ntfsPos + 1] = static_cast<char>(QUAZIP_EXTRA_NTFS_MAGIC >> 8);
187         extra[ntfsPos + 2] = 32; // the 2-byte size in LittleEndian
188         extra[ntfsPos + 3] = 0;
189         // zero the record
190         memset(extra.data() + ntfsPos + 4, 0, 32);
191         timesPos = ntfsPos + 8;
192         // now set the tag data
193         extra[timesPos] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC);
194         extra[timesPos + 1] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC
195                                                 >> 8);
196         // the size:
197         extra[timesPos + 2] = 24;
198         extra[timesPos + 3] = 0;
199         ntfsTimesLength = 24;
200     }
201     if (timesPos == -1) {
202         // No time tag in the NTFS record, need to add one.
203         timesPos = ntfsPos + 4 + ntfsLength;
204         extra.resize(extra.size() + 28);
205         // Now we need to move the rest of the field
206         // (possibly zero bytes, but memmove() is OK with that).
207         // 0 ......... ntfsPos .. ntfsPos + 4   ... timesPos
208         // <some data> <header>   <NTFS record>     <need-to-move data>    <end>
209         memmove(extra.data() + timesPos + 28, extra.data() + timesPos,
210                 extra.size() - 28 - timesPos);
211         ntfsLength += 28;
212         // now set the tag data
```

```
213         extra[timesPos] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC);
214         extra[timesPos + 1] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC
215                                                 >> 8);
216         // the size:
217         extra[timesPos + 2] = 24;
218         extra[timesPos + 3] = 0;
219         // zero the record
220         memset(extra.data() + timesPos + 4, 0, 24);
221         ntfsTimesLength = 24;
222     }
223     if (ntfsTimesLength < 24) {
224         // Broken times field. OK, this is really unlikely, but just in case...
225         size_t timesEnd = timesPos + 4 + ntfsTimesLength;
226         extra.resize(extra.size() + (24 - ntfsTimesLength));
227         // Move it!
228         // 0 ......... timesPos .... timesPos + 4 .. timesEnd
229         // <some data> <time header> <broken times> <need-to-move data> <end>
230         memmove(extra.data() + timesEnd + (24 - ntfsTimesLength),
231                 extra.data() + timesEnd,
232                 extra.size() - (24 - ntfsTimesLength) - timesEnd);
233         // Now we have to increase the NTFS record and time tag lengths.
234         ntfsLength += (24 - ntfsTimesLength);
235         ntfsTimesLength = 24;
236         extra[ntfsPos + 2] = static_cast<char>(ntfsLength);
237         extra[ntfsPos + 3] = static_cast<char>(ntfsLength >> 8);
238         extra[timesPos + 2] = static_cast<char>(ntfsTimesLength);
239         extra[timesPos + 3] = static_cast<char>(ntfsTimesLength >> 8);
240     }
241     QDateTime base(QDate(1601, 1, 1), QTime(0, 0), Qt::UTC);
242 #if (QT_VERSION >= 0x040700)
243     quint64 ticks = base.msecsTo(time) * 10000 + fineTicks;
244 #else
245     QDateTime utc = time.toUTC();
246     quint64 ticks = (static_cast<qint64>(base.date().daysTo(utc.date()))
247             * Q_INT64_C(86400000)
248             + static_cast<qint64>(base.time().msecsTo(utc.time())))
249         * Q_INT64_C(10000) + fineTicks;
250 #endif
251     extra[timesPos + 4 + position] = static_cast<char>(ticks);
252     extra[timesPos + 5 + position] = static_cast<char>(ticks >> 8);
253     extra[timesPos + 6 + position] = static_cast<char>(ticks >> 16);
254     extra[timesPos + 7 + position] = static_cast<char>(ticks >> 24);
255     extra[timesPos + 8 + position] = static_cast<char>(ticks >> 32);
256     extra[timesPos + 9 + position] = static_cast<char>(ticks >> 40);
257     extra[timesPos + 10 + position] = static_cast<char>(ticks >> 48);
258     extra[timesPos + 11 + position] = static_cast<char>(ticks >> 56);
259 }
```

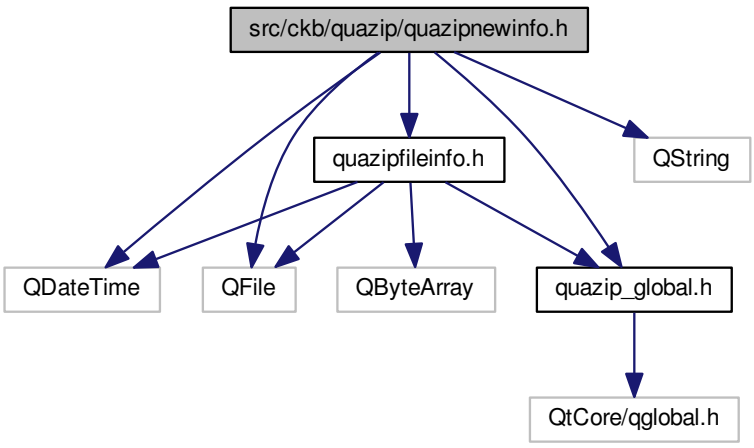
Here is the caller graph for this function:



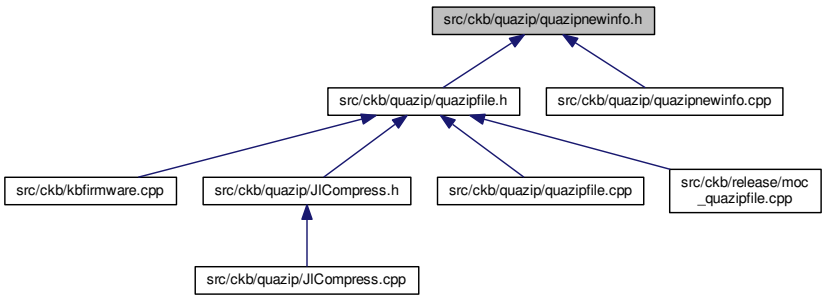
13.159 src/ckb/quazip/quazipnewinfo.h File Reference

```
#include <QDateTime>
#include <QFile>
#include <QString>
#include "quazip_global.h"
#include "quazipfileinfo.h"
```

Include dependency graph for quazipnewinfo.h:



This graph shows which files directly or indirectly include this file:



Data Structures

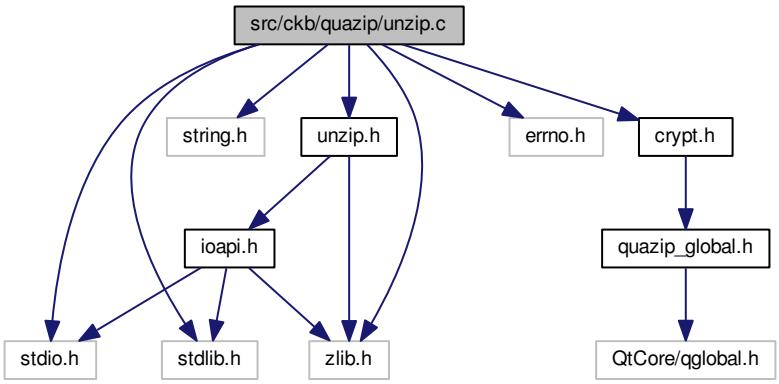
- struct QuaZipNewInfo

  Information about a file to be created.

13.160 src/ckb/quazip/unzip.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "zlib.h"
#include "unzip.h"
#include <errno.h>
#include "crypt.h"
```

Include dependency graph for unzip.c:



Data Structures

- struct unz_file_info64_internal_s
- struct file_in_zip64_read_info_s
- struct unz64_s

Macros

- #define local static
- #define CASESENSITIVITYDEFAULT_NO
- #define UNZ_BUFSIZE (16384)
- #define UNZ_MAXFILENAMEINZIP (256)
- #define ALLOC(size) (malloc(size))
- #define TRYFREE(p) {if (p) free(p);}
- #define SIZECENTRALDIRITEM (0x2e)
- #define SIZEZIPLOCALHEADER (0x1e)
- #define CASESENSITIVITYDEFAULTVALUE 2
- #define STRCMPCASENOSENTIVEFUNCTION strcmpcasenosensitive_internal
- #define BUFREADCOMMENT (0x400)

Typedefs

- typedef uLongf z_crc_t
- typedef struct unz_file_info64_internal_s unz_file_info64_internal

Functions

- static int unz64local_getByte _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi))
- static int unz64local_getByte (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi)
- static int unz64local_getShort _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX))

- static int unz64local_getShort (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)
- static int unz64local_getLong (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)
- static int unz64local_getLong64 _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX))
- static int unz64local_getLong64 (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX)
- static int strcmpcasenosensitive_internal (const char *fileName1, const char *fileName2)
- int ZEXPORT unzStringFileNameCompare (const char *fileName1, const char *fileName2, int iCaseSensitivity)
- static ZPOS64_T unz64local_SearchCentralDir _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream))
- static ZPOS64_T unz64local_SearchCentralDir (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)
- static ZPOS64_T unz64local_SearchCentralDir64 (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)
- unzFile unzOpenInternal (voidpf file, zlib_filefunc64_32_def *pzlib_filefunc64_32_def, int is64bitOpenFunction, unsigned flags)
- unzFile ZEXPORT unzOpen2 (voidpf file, zlib_filefunc_def *pzlib_filefunc32_def)
- unzFile ZEXPORT unzOpen2_64 (voidpf file, zlib_filefunc64_def *pzlib_filefunc_def)
- unzFile ZEXPORT unzOpen (voidpf file)
- unzFile ZEXPORT unzOpen64 (voidpf file)
- int ZEXPORT unzClose (unzFile file)
- int ZEXPORT unzGetGlobalInfo64 (unzFile file, unz_global_info64 *pglobal_info)
- int ZEXPORT unzGetGlobalInfo (unzFile file, unz_global_info *pglobal_info32)
- static void unz64local_DosDateToTmuDate (ZPOS64_T ulDosDate, tm_unz *ptm)
- static int unz64local_GetCurrentFileInfoInternal _Z_OF ((unzFile file, unz_file_info64 *pfile_info, unz_file_info64_internal *pfile_info_internal, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize))
- static int unz64local_GetCurrentFileInfoInternal (unzFile file, unz_file_info64 *pfile_info, unz_file_info64_internal *pfile_info_internal, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize)
- int ZEXPORT unzGetCurrentFileInfo64 (unzFile file, unz_file_info64 *pfile_info, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize)
- int ZEXPORT unzGetCurrentFileInfo (unzFile file, unz_file_info *pfile_info, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize)
- int ZEXPORT unzGoToFirstFile (unzFile file)
- int ZEXPORT unzGoToNextFile (unzFile file)
- int ZEXPORT unzLocateFile (unzFile file, const char *szFileName, int iCaseSensitivity)
- int ZEXPORT unzGetFilePos64 (unzFile file, unz64_file_pos *file_pos)
- int ZEXPORT unzGetFilePos (unzFile file, unz_file_pos *file_pos)
- int ZEXPORT unzGoToFilePos64 (unzFile file, const unz64_file_pos *file_pos)
- int ZEXPORT unzGoToFilePos (unzFile file, unz_file_pos *file_pos)
- static int unz64local_CheckCurrentFileCoherencyHeader (unz64_s *s, uInt *piSizeVar, ZPOS64_T *poffset_local_extrafield, uInt *psize_local_extrafield)
- int ZEXPORT unzOpenCurrentFile3 (unzFile file, int *method, int *level, int raw, const char *password)
- int ZEXPORT unzOpenCurrentFile (unzFile file)
- int ZEXPORT unzOpenCurrentFilePassword (unzFile file, const char *password)
- int ZEXPORT unzOpenCurrentFile2 (unzFile file, int *method, int *level, int raw)
- ZPOS64_T ZEXPORT unzGetCurrentFileZStreamPos64 (unzFile file)

  Addition for GDAL : START.
- int ZEXPORT unzReadCurrentFile (unzFile file, voidp buf, unsigned len)

Addition for GDAL : END.

- z_off_t ZEXPORT unztell (unzFile file)
- ZPOS64_T ZEXPORT unztell64 (unzFile file)
- int ZEXPORT unzeof (unzFile file)
- int ZEXPORT unzGetLocalExtrafield (unzFile file, voidp buf, unsigned len)
- int ZEXPORT unzCloseCurrentFile (unzFile file)
- int ZEXPORT unzGetGlobalComment (unzFile file, char *szComment, uLong uSizeBuf)
- ZPOS64_T ZEXPORT unzGetOffset64 (unzFile file)
- uLong ZEXPORT unzGetOffset (unzFile file)
- int ZEXPORT unzSetOffset64 (unzFile file, ZPOS64_T pos)
- int ZEXPORT unzSetOffset (unzFile file, uLong pos)
- int ZEXPORT unzSetFlags (unzFile file, unsigned flags)
- int ZEXPORT unzClearFlags (unzFile file, unsigned flags)

Variables

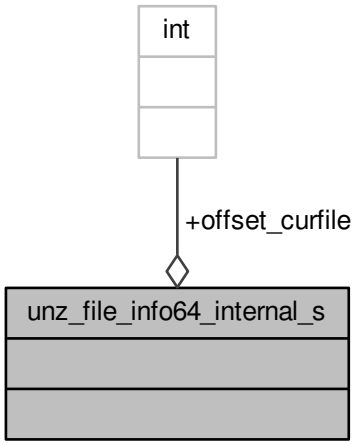
- const char unz_copyright []

13.160.1 Data Structure Documentation

13.160.1.1 struct unz_file_info64_internal_s

Definition at line 128 of file unzip.c.

Collaboration diagram for unz_file_info64_internal_s:



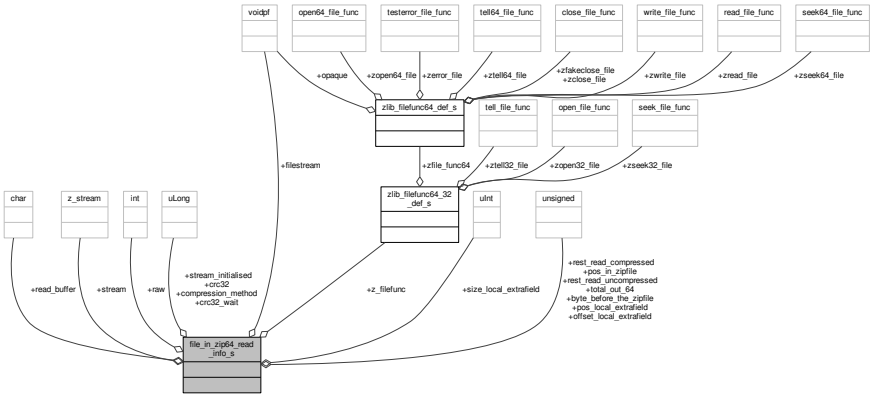
Data Fields

ZPOS64_T	offset_curfile	

13.160.1.2 struct file_in_zip64_read_info_s

Definition at line 136 of file unzip.c.

Collaboration diagram for file_in_zip64_read_info_s:



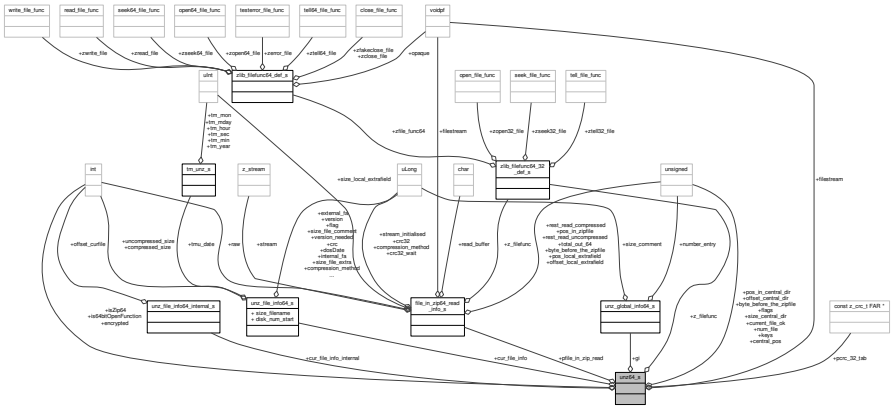
Data Fields

ZPOS64_T	byte_before_the-_zipfile	
uLong	compression_-method	
uLong	crc32	
uLong	crc32_wait	
voidpf	filestream	
ZPOS64_T	offset_local_-extrafield	
ZPOS64_T	pos_in_zipfile	
ZPOS64_T	pos_local_-extrafield	
int	raw	
char ∗	read_buffer	
ZPOS64_T	rest_read_-compressed	
ZPOS64_T	rest_read_-uncompressed	
uInt	size_local_-extrafield	
z_stream	stream	
uLong	stream_-initialised	
ZPOS64_T	total_out_64	
zlib_filefunc64_-32_def	z_filefunc	

13.160.1.3 struct unz64_s

Definition at line 167 of file unzip.c.

Collaboration diagram for unz64_s:



Data Fields

ZPOS64_T	byte_before_the-_zipfile	
ZPOS64_T	central_pos	
unz_file_info64	cur_file_info	
unz_file_info64_-internal	cur_file_info_-internal	
ZPOS64_T	current_file_ok	
int	encrypted	
voidpf	filestream	
unsigned	flags	
unz_global_-info64	gi	
int	is64bitOpen-Function	
int	isZip64	
unsigned long	keys[3]	
ZPOS64_T	num_file	
ZPOS64_T	offset_central_-dir	
const z_crc_t FAR *	pcrc_32_tab	
file_in_zip64_-read_info_s *	pfile_in_zip_read	
ZPOS64_T	pos_in_central_-dir	
ZPOS64_T	size_central_dir	
zlib_filefunc64_-32_def	z_filefunc	

13.160.2 Macro Definition Documentation

13.160.2.1 #define ALLOC(*size*) (malloc(size))

Definition at line 114 of file unzip.c.

Referenced by unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), unzOpenCurrentFile3(), and

unzOpenInternal().

13.160.2.2 #define BUFREADCOMMENT (0x400)

Definition at line 408 of file unzip.c.

Referenced by unz64local_SearchCentralDir(), and unz64local_SearchCentralDir64().

13.160.2.3 #define CASESENSITIVITYDEFAULT_NO

Definition at line 100 of file unzip.c.

13.160.2.4 #define CASESENSITIVITYDEFAULTVALUE 2

Definition at line 375 of file unzip.c.

Referenced by unzStringFileNameCompare().

13.160.2.5 #define local static

Definition at line 93 of file unzip.c.

13.160.2.6 #define SIZECENTRALDIRITEM (0x2e)

Definition at line 120 of file unzip.c.

Referenced by unzGoToNextFile().

13.160.2.7 #define SIZEZIPLOCALHEADER (0x1e)

Definition at line 121 of file unzip.c.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), and unzOpenCurrentFile3().

13.160.2.8 #define STRCMPCASENOSENTIVEFUNCTION strcmpcasenosensitive_internal

Definition at line 381 of file unzip.c.

Referenced by unzStringFileNameCompare().

13.160.2.9 #define TRYFREE(*p*) {if (p) free(p);}

Definition at line 117 of file unzip.c.

Referenced by unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), unzClose(), unzCloseCurrentFile(), and unzOpenCurrentFile3().

13.160.2.10 #define UNZ_BUFSIZE (16384)

Definition at line 106 of file unzip.c.

Referenced by unzOpenCurrentFile3(), and unzReadCurrentFile().

13.160.2.11 #define UNZ_MAXFILENAMEINZIP (256)

Definition at line 110 of file unzip.c.

Referenced by unzLocateFile().

13.160.3 Typedef Documentation

13.160.3.1 typedef struct unz_file_info64_internal_s unz_file_info64_internal

13.160.3.2 typedef uLongf z_crc_t

Definition at line 76 of file unzip.c.

13.160.4 Function Documentation

13.160.4.1 static int unz64local_getByte _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi)) [static]

13.160.4.2 static int unz64local_getShort _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)) [static]

13.160.4.3 static int unz64local_getLong64 _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX)) [static]

13.160.4.4 static ZPOS64_T unz64local_SearchCentralDir _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)) [static]

13.160.4.5 static int unz64local_GetCurrentFileInfoInternal _Z_OF ((unzFile file, unz_file_info64 *pfile_info, unz_file_info64_internal *pfile_info_internal, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize)) [static]

13.160.4.6 static int strcmpcasenosensitive_internal (const char * *fileName1*, const char * *fileName2*) [static]

Definition at line 352 of file unzip.c.

```
353 {
354     for (;;)
355     {
356         char c1=*(fileName1++);
357         char c2=*(fileName2++);
358         if ((c1>='a') && (c1<='z'))
359             c1 -= 0x20;
360         if ((c2>='a') && (c2<='z'))
361             c2 -= 0x20;
362         if (c1=='\0')
363             return ((c2=='\0') ? 0 : -1);
364         if (c2=='\0')
365             return 1;
366         if (c1<c2)
367             return -1;
368         if (c1>c2)
369             return 1;
370     }
371 }
```

13.160.4.7 static int unz64local_CheckCurrentFileCoherencyHeader (unz64_s * *s*, uInt * *piSizeVar*, ZPOS64_T * *poffset_local_extrafield*, uInt * *psize_local_extrafield*) [static]

Definition at line 1393 of file unzip.c.

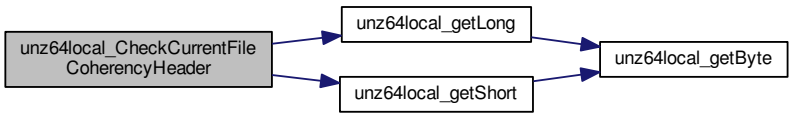
References unz64_s::byte_before_the_zipfile, unz_file_info64_s::compressed_size, unz_file_info64_s::compression_method, unz_file_info64_s::crc, unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::filestream, unz_file_info64_internal_s::offset_curfile, unz_file_info64_s::size_filename, SIZEZIPLOCALHEADER, unz_file_info64_s::uncompressed_size, unz64local_getLong(), unz64local_getShort(), UNZ_BADZIPFILE, UNZ_ERRNO, UNZ_OK, Z_BZIP2ED, unz64_s::z_filefunc, ZLIB_FILEFUNC_SEEK_SET, and ZSEEK64.

Referenced by unzOpenCurrentFile3().

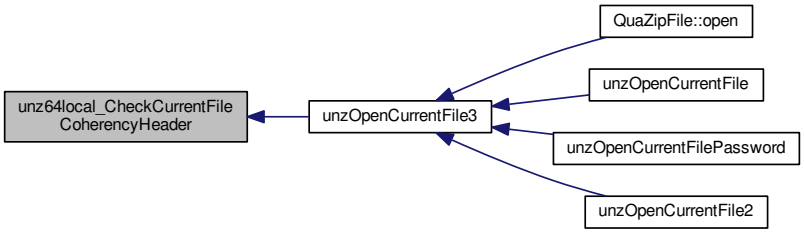
```
1396 {
1397     uLong uMagic,uData,uFlags;
1398     uLong size_filename;
1399     uLong size_extra_field;
1400     int err=UNZ_OK;
1401
1402     *piSizeVar = 0;
1403     *poffset_local_extrafield = 0;
1404     *psize_local_extrafield = 0;
1405
1406     if (ZSEEK64(s->z_filefunc, s->filestream,s->
     cur_file_info_internal.offset_curfile +
1407                                 s->byte_before_the_zipfile,
     ZLIB_FILEFUNC_SEEK_SET)!=0)
1408         return UNZ_ERRNO;
1409
1410
1411     if (err==UNZ_OK)
1412     {
1413         if (unz64local_getLong(&s->z_filefunc, s->
     filestream,&uMagic) != UNZ_OK)
1414             err=UNZ_ERRNO;
1415         else if (uMagic!=0x04034b50)
1416             err=UNZ_BADZIPFILE;
1417     }
1418
1419     if (unz64local_getShort(&s->z_filefunc, s->
     filestream,&uData) != UNZ_OK)
1420         err=UNZ_ERRNO;
1421 /*
1422     else if ((err==UNZ_OK) && (uData!=s->cur_file_info.wVersion))
1423         err=UNZ_BADZIPFILE;
1424 */
1425     if (unz64local_getShort(&s->z_filefunc, s->
     filestream,&uFlags) != UNZ_OK)
1426         err=UNZ_ERRNO;
1427
1428     if (unz64local_getShort(&s->z_filefunc, s->
     filestream,&uData) != UNZ_OK)
1429         err=UNZ_ERRNO;
1430     else if ((err==UNZ_OK) && (uData!=s->cur_file_info.
     compression_method))
1431         err=UNZ_BADZIPFILE;
1432
1433     if ((err==UNZ_OK) && (s->cur_file_info.compression_method!=0) &&
1434 /* #ifdef HAVE_BZIP2 */
1435                         (s->cur_file_info.compression_method!=
     Z_BZIP2ED) &&
1436 /* #endif */
1437                         (s->cur_file_info.compression_method!=Z_DEFLATED))
1438         err=UNZ_BADZIPFILE;
1439
1440     if (unz64local_getLong(&s->z_filefunc, s->
     filestream,&uData) != UNZ_OK) /* date/time */
1441         err=UNZ_ERRNO;
1442
1443     if (unz64local_getLong(&s->z_filefunc, s->
     filestream,&uData) != UNZ_OK) /* crc */
1444         err=UNZ_ERRNO;
1445     else if ((err==UNZ_OK) && (uData!=s->cur_file_info.crc) && ((uFlags & 8)==0))
1446         err=UNZ_BADZIPFILE;
1447
1448     if (unz64local_getLong(&s->z_filefunc, s->
     filestream,&uData) != UNZ_OK) /* size compr */
1449         err=UNZ_ERRNO;
1450     else if (uData != 0xFFFFFFFF && (err==UNZ_OK) && (uData!=s->
     cur_file_info.compressed_size) && ((uFlags & 8)==0))
1451         err=UNZ_BADZIPFILE;
1452
1453     if (unz64local_getLong(&s->z_filefunc, s->
     filestream,&uData) != UNZ_OK) /* size uncompr */
1454         err=UNZ_ERRNO;
1455     else if (uData != 0xFFFFFFFF && (err==UNZ_OK) && (uData!=s->
     cur_file_info.uncompressed_size) && ((uFlags & 8)==0))
1456         err=UNZ_BADZIPFILE;
1457
1458     if (unz64local_getShort(&s->z_filefunc, s->
```

```
    filestream,&size_filename) != UNZ_OK)
1459        err=UNZ_ERRNO;
1460    else if ((err==UNZ_OK) && (size_filename!=s->cur_file_info.
    size_filename))
1461        err=UNZ_BADZIPFILE;
1462
1463    *piSizeVar += (uInt)size_filename;
1464
1465    if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&size_extra_field) != UNZ_OK)
1466        err=UNZ_ERRNO;
1467    *poffset_local_extrafield= s->cur_file_info_internal.
    offset_curfile +
1468                                SIZEZIPLOCALHEADER + size_filename;
1469    *psize_local_extrafield = (uInt)size_extra_field;
1470
1471    *piSizeVar += (uInt)size_extra_field;
1472
1473    return err;
1474 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.8 static void unz64local_DosDateToTmuDate (ZPOS64_T *ulDosDate*, tm_unz * *ptm*) [static]

Definition at line 862 of file unzip.c.

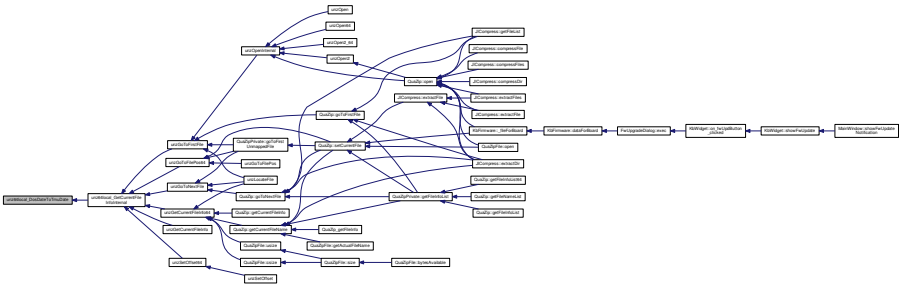
References tm_unz_s::tm_hour, tm_unz_s::tm_mday, tm_unz_s::tm_min, tm_unz_s::tm_mon, tm_unz_s::tm_sec, and tm_unz_s::tm_year.

Referenced by unz64local_GetCurrentFileInfoInternal().

```
863 {
864     ZPOS64_T uDate;
865     uDate = (ZPOS64_T)(ulDosDate>>16);
866     ptm->tm_mday = (uInt)(uDate&0x1f) ;
867     ptm->tm_mon =  (uInt)((((uDate)&0x1E0)/0x20)-1) ;
868     ptm->tm_year = (uInt)(((uDate&0x0FE00)/0x0200)+1980) ;
869
870     ptm->tm_hour = (uInt) ((ulDosDate &0xF800)/0x800);
871     ptm->tm_min =  (uInt) ((ulDosDate&0x7E0)/0x20) ;
```

```
872     ptm->tm_sec =  (uInt) (2*(ulDosDate&0x1f)) ;
873 }
```

Here is the caller graph for this function:



13.160.4.9 static int unz64local_getByte (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* int * *pi*) [static]

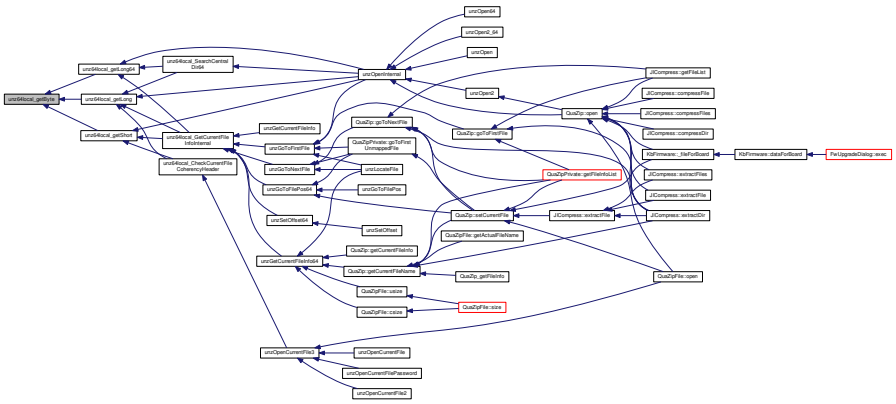
Definition at line 215 of file unzip.c.

References UNZ_EOF, UNZ_ERRNO, UNZ_OK, ZERROR64, and ZREAD64.

Referenced by unz64local_getLong(), unz64local_getLong64(), and unz64local_getShort().

```
216 {
217     unsigned char c;
218     int err = (int)ZREAD64(*pzlib_filefunc_def,filestream,&c,1);
219     if (err==1)
220     {
221         *pi = (int)c;
222         return UNZ_OK;
223     }
224     else
225     {
226         if (ZERROR64(*pzlib_filefunc_def,filestream))
227             return UNZ_ERRNO;
228         else
229             return UNZ_EOF;
230     }
231 }
```

Here is the caller graph for this function:



13.160.4.10 static int unz64local_GetCurrentFileInfoInternal (unzFile *file*, unz_file_info64 * *pfile_info*, unz_file_info64_internal * *pfile_info_internal*, char * *szFileName*, uLong *fileNameBufferSize*, void * *extraField*, uLong *extraFieldBufferSize*, char * *szComment*, uLong *commentBufferSize*) [static]

Definition at line 889 of file unzip.c.

References unz64_s::byte_before_the_zipfile, unz_file_info64_s::compressed_size, unz_file_info64_s::compression_method, unz_file_info64_s::crc, unz_file_info64_s::disk_num_start, unz_file_info64_s::dosDate, unz_file_info64_s::external_fa, unz64_s::filestream, unz_file_info64_s::flag, unz_file_info64_s::internal_fa, unz_file_info64_internal_s::offset_curfile, unz64_s::pos_in_central_dir, unz_file_info64_s::size_file_comment, unz_file_info64_s::size_file_extra, unz_file_info64_s::size_filename, unz_file_info64_s::tmu_date, unz_file_info64_s::uncompressed_size, unz64local_DosDateToTmuDate(), unz64local_getLong(), unz64local_getLong64(), unz64local_getShort(), UNZ_BADZIPFILE, UNZ_ERRNO, UNZ_OK, UNZ_PARAMERROR, unz_file_info64_s::version, unz_file_info64_s::version_needed, unz64_s::z_filefunc, ZLIB_FILEFUNC_SEEK_CUR, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, and ZSEEK64.

Referenced by unzGetCurrentFileInfo(), unzGetCurrentFileInfo64(), unzGoToFilePos64(), unzGoToFirstFile(), unzGoToNextFile(), and unzSetOffset64().

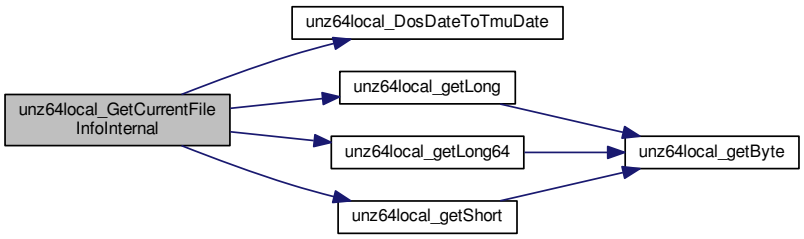
```
899 {
900     unz64_s* s;
901     unz_file_info64 file_info;
902     unz_file_info64_internal file_info_internal;
903     int err=UNZ_OK;
904     uLong uMagic;
905     ZPOS64_T llSeek=0;
906     uLong uL;
907
908     if (file==NULL)
909         return UNZ_PARAMERROR;
910     s=(unz64_s*)file;
911     if (ZSEEK64(s->z_filefunc, s->filestream,
912               s->pos_in_central_dir+s->byte_before_the_zipfile,
913               ZLIB_FILEFUNC_SEEK_SET)!=0)
914         err=UNZ_ERRNO;
915
916
917     /* we check the magic */
918     if (err==UNZ_OK)
919     {
920         if (unz64local_getLong(&s->z_filefunc, s->
    filestream,&uMagic) != UNZ_OK)
921             err=UNZ_ERRNO;
922         else if (uMagic!=0x02014b50)
923             err=UNZ_BADZIPFILE;
924     }
925
926     if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.version) != UNZ_OK)
927         err=UNZ_ERRNO;
928
929     if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.version_needed) != UNZ_OK)
930         err=UNZ_ERRNO;
931
932     if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.flag) != UNZ_OK)
933         err=UNZ_ERRNO;
934
935     if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.compression_method) != UNZ_OK)
936         err=UNZ_ERRNO;
937
938     if (unz64local_getLong(&s->z_filefunc, s->
    filestream,&file_info.dosDate) != UNZ_OK)
939         err=UNZ_ERRNO;
940
941     unz64local_DosDateToTmuDate(file_info.dosDate,&file_info.
    tmu_date);
942
943     if (unz64local_getLong(&s->z_filefunc, s->
    filestream,&file_info.crc) != UNZ_OK)
944         err=UNZ_ERRNO;
945
946     if (unz64local_getLong(&s->z_filefunc, s->
    filestream,&uL) != UNZ_OK)
947         err=UNZ_ERRNO;
948     file_info.compressed_size = uL;
949
950     if (unz64local_getLong(&s->z_filefunc, s->
```

```
    filestream,&uL) != UNZ_OK)
951         err=UNZ_ERRNO;
952       file_info.uncompressed_size = uL;
953
954       if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.size_filename) != UNZ_OK)
955         err=UNZ_ERRNO;
956
957       if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.size_file_extra) != UNZ_OK)
958         err=UNZ_ERRNO;
959
960       if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.size_file_comment) != UNZ_OK)
961         err=UNZ_ERRNO;
962
963       if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.disk_num_start) != UNZ_OK)
964         err=UNZ_ERRNO;
965
966       if (unz64local_getShort(&s->z_filefunc, s->
    filestream,&file_info.internal_fa) != UNZ_OK)
967         err=UNZ_ERRNO;
968
969       if (unz64local_getLong(&s->z_filefunc, s->
    filestream,&file_info.external_fa) != UNZ_OK)
970         err=UNZ_ERRNO;
971
972                 /* relative offset of local header */
973       if (unz64local_getLong(&s->z_filefunc, s->
    filestream,&uL) != UNZ_OK)
974         err=UNZ_ERRNO;
975       file_info_internal.offset_curfile = uL;
976
977       llSeek+=file_info.size_filename;
978       if ((err==UNZ_OK) && (szFileName!=NULL))
979       {
980           uLong uSizeRead ;
981           if (file_info.size_filename<fileNameBufferSize)
982           {
983               *(szFileName+file_info.size_filename)='\0';
984               uSizeRead = file_info.size_filename;
985           }
986           else
987               uSizeRead = fileNameBufferSize;
988
989           if ((file_info.size_filename>0) && (fileNameBufferSize>0))
990               if (ZREAD64(s->z_filefunc, s->filestream,szFileName,uSizeRead)!=
    uSizeRead)
991                   err=UNZ_ERRNO;
992           llSeek -= uSizeRead;
993       }
994
995       /* Read extrafield */
996       if ((err==UNZ_OK) && (extraField!=NULL))
997       {
998           ZPOS64_T uSizeRead ;
999           if (file_info.size_file_extra<extraFieldBufferSize)
1000              uSizeRead = file_info.size_file_extra;
1001          else
1002              uSizeRead = extraFieldBufferSize;
1003
1004          if (llSeek!=0)
1005          {
1006              if (ZSEEK64(s->z_filefunc, s->filestream,llSeek,
    ZLIB_FILEFUNC_SEEK_CUR)==0)
1007                  llSeek=0;
1008              else
1009                  err=UNZ_ERRNO;
1010          }
1011
1012          if ((file_info.size_file_extra>0) && (extraFieldBufferSize>0))
1013              if (ZREAD64(s->z_filefunc, s->filestream,extraField,(uLong)uSizeRead
    )!=uSizeRead)
1014                  err=UNZ_ERRNO;
1015
1016          llSeek += file_info.size_file_extra - (uLong)uSizeRead;
1017      }
1018      else
1019          llSeek += file_info.size_file_extra;
1020
1021
1022      if ((err==UNZ_OK) && (file_info.size_file_extra != 0))
1023      {
1024                                  uLong acc = 0;
1025
1026          /* since lSeek now points to after the extra field we need to move back */
```

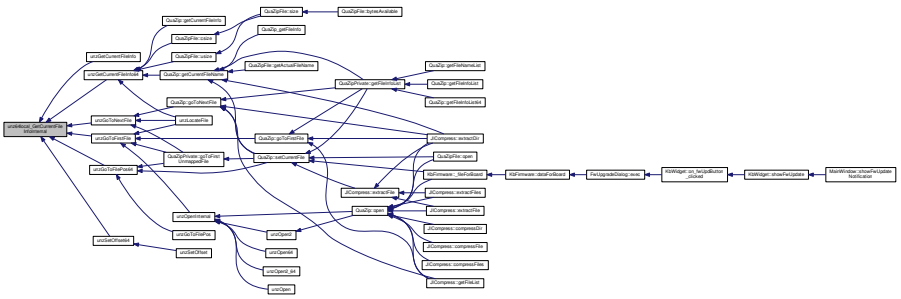
```
1027         llSeek -= file_info.size_file_extra;
1028
1029         if (llSeek!=0)
1030         {
1031             if (ZSEEK64(s->z_filefunc, s->filestream,llSeek,
     ZLIB_FILEFUNC_SEEK_CUR)==0)
1032                 llSeek=0;
1033             else
1034                 err=UNZ_ERRNO;
1035         }
1036
1037         while(acc < file_info.size_file_extra)
1038         {
1039             uLong headerId;
1040                                                 uLong dataSize;
1041
1042             if (unz64local_getShort(&s->z_filefunc, s->
     filestream,&headerId) != UNZ_OK)
1043                 err=UNZ_ERRNO;
1044
1045             if (unz64local_getShort(&s->z_filefunc, s->
     filestream,&dataSize) != UNZ_OK)
1046                 err=UNZ_ERRNO;
1047
1048             /* ZIP64 extra fields */
1049             if (headerId == 0x0001)
1050             {
1051                 uLong uL;
1052
1053                 if(file_info.uncompressed_size == (ZPOS64_T)0xFFFFFFFFu)
1054                 {
1055                     if (unz64local_getLong64(&s->z_filefunc, s->
     filestream,&file_info.uncompressed_size) != UNZ_OK)
1056                         err=UNZ_ERRNO;
1057                 }
1058
1059                 if(file_info.compressed_size == (ZPOS64_T)0xFFFFFFFFu)
1060                 {
1061                     if (unz64local_getLong64(&s->z_filefunc, s->
     filestream,&file_info.compressed_size) != UNZ_OK)
1062                         err=UNZ_ERRNO;
1063                 }
1064
1065                 if(file_info_internal.offset_curfile == (ZPOS64_T)0xFFFFFFFFu)
1066                 {
1067                     /* Relative Header offset */
1068                     if (unz64local_getLong64(&s->z_filefunc, s->
     filestream,&file_info_internal.offset_curfile) != UNZ_OK)
1069                         err=UNZ_ERRNO;
1070                 }
1071
1072                 if(file_info.disk_num_start == 0xFFFFFFFFu)
1073                 {
1074                     /* Disk Start Number */
1075                     if (unz64local_getLong(&s->z_filefunc, s->
     filestream,&uL) != UNZ_OK)
1076                         err=UNZ_ERRNO;
1077                 }
1078
1079             }
1080             else
1081             {
1082                 if (ZSEEK64(s->z_filefunc, s->filestream,dataSize,
     ZLIB_FILEFUNC_SEEK_CUR)!=0)
1083                     err=UNZ_ERRNO;
1084             }
1085
1086             acc += 2 + 2 + dataSize;
1087         }
1088     }
1089
1090     if ((err==UNZ_OK) && (szComment!=NULL))
1091     {
1092         uLong uSizeRead ;
1093         if (file_info.size_file_comment<commentBufferSize)
1094         {
1095             *(szComment+file_info.size_file_comment)='\0';
1096             uSizeRead = file_info.size_file_comment;
1097         }
1098         else
1099             uSizeRead = commentBufferSize;
1100
1101         if (llSeek!=0)
1102         {
1103             if (ZSEEK64(s->z_filefunc, s->filestream,llSeek,
     ZLIB_FILEFUNC_SEEK_CUR)==0)
1104                 llSeek=0;
```

```
1105             else
1106                 err=UNZ_ERRNO;
1107         }
1108
1109         if ((file_info.size_file_comment>0) && (commentBufferSize>0))
1110             if (ZREAD64(s->z_filefunc, s->filestream,szComment,uSizeRead)!=
     uSizeRead)
1111                 err=UNZ_ERRNO;
1112         llSeek+=file_info.size_file_comment - uSizeRead;
1113     }
1114     else
1115         llSeek+=file_info.size_file_comment;
1116
1117
1118     if ((err==UNZ_OK) && (pfile_info!=NULL))
1119         *pfile_info=file_info;
1120
1121     if ((err==UNZ_OK) && (pfile_info_internal!=NULL))
1122         *pfile_info_internal=file_info_internal;
1123
1124     return err;
1125 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.11 static int unz64local_getLong (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* uLong * *pX*) [static]

Definition at line 269 of file unzip.c.

References unz64local_getByte(), UNZ_OK, and x.

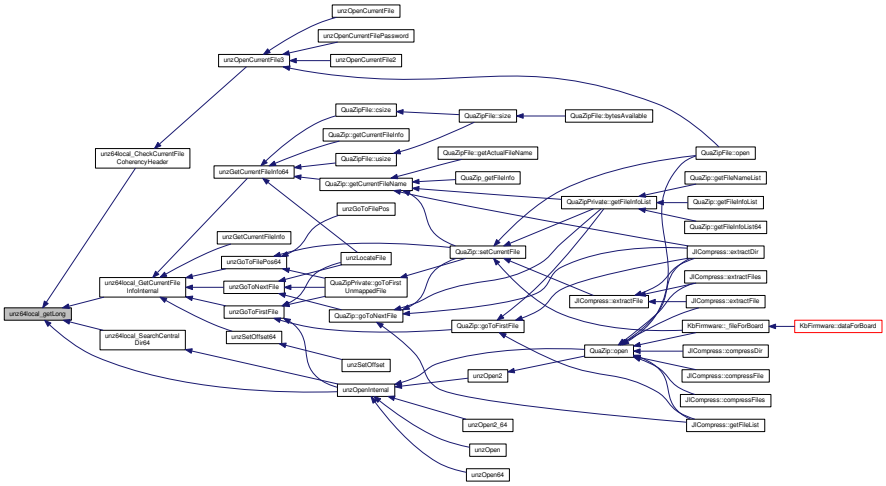
Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir64(), and unzOpenInternal().

```
272 {
273     uLong x ;
```

```
274     int i = 0;
275     int err;
276
277     err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
278     x = (uLong)i;
279
280     if (err==UNZ_OK)
281         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
282     x |= ((uLong)i)<<8;
283
284     if (err==UNZ_OK)
285         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
286     x |= ((uLong)i)<<16;
287
288     if (err==UNZ_OK)
289         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
290     x += ((uLong)i)<<24;
291
292     if (err==UNZ_OK)
293         *pX = x;
294     else
295         *pX = 0;
296     return err;
297 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.160.4.12 static int unz64local_getLong64 (const zlib_filefunc64_32_def * *pzlib_filefunc_def*, voidpf *filestream*, ZPOS64_T * *pX*) [static]

Definition at line 305 of file unzip.c.

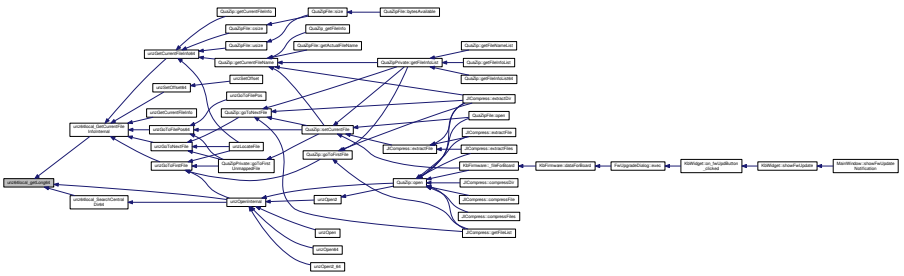
References unz64local_getByte(), UNZ_OK, and x.

Referenced by unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir64(), and unzOpenInternal().

```
308 {
309     ZPOS64_T x ;
310     int i = 0;
311     int err;
312
313     err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
314     x = (ZPOS64_T)i;
315
316     if (err==UNZ_OK)
317         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
318     x |= ((ZPOS64_T)i)<<8;
319
320     if (err==UNZ_OK)
321         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
322     x |= ((ZPOS64_T)i)<<16;
323
324     if (err==UNZ_OK)
325         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
326     x |= ((ZPOS64_T)i)<<24;
327
328     if (err==UNZ_OK)
329         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
330     x |= ((ZPOS64_T)i)<<32;
331
332     if (err==UNZ_OK)
333         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
334     x |= ((ZPOS64_T)i)<<40;
335
336     if (err==UNZ_OK)
337         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
338     x |= ((ZPOS64_T)i)<<48;
339
340     if (err==UNZ_OK)
341         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
342     x |= ((ZPOS64_T)i)<<56;
343
344     if (err==UNZ_OK)
345         *pX = x;
346     else
347         *pX = 0;
348     return err;
349 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.160.4.13 static int unz64local_getShort (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* uLong * *pX*) [static]

Definition at line 242 of file unzip.c.

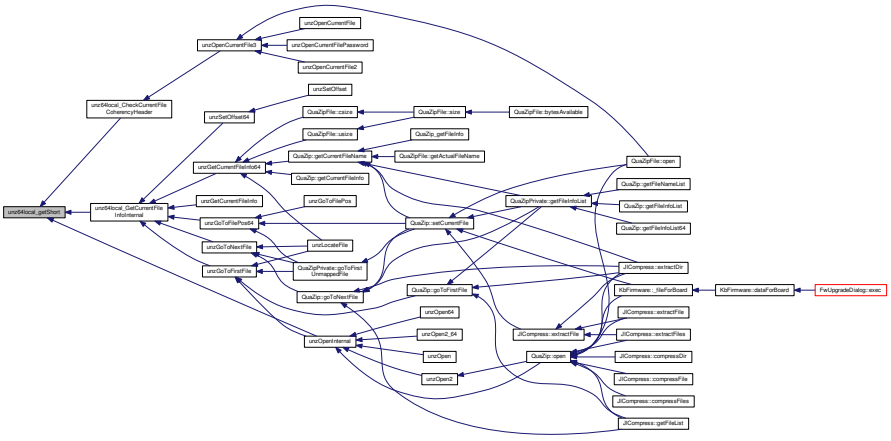
References unz64local_getByte(), UNZ_OK, and x.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), and unzOpenInternal().

```
245 {
246     uLong x ;
247     int i = 0;
248     int err;
249
250     err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
251     x = (uLong)i;
252
253     if (err==UNZ_OK)
254         err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
255     x |= ((uLong)i)<<8;
256
257     if (err==UNZ_OK)
258         *pX = x;
259     else
260         *pX = 0;
261     return err;
262 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.160.4.14 static ZPOS64_T unz64local_SearchCentralDir (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream*) [static]

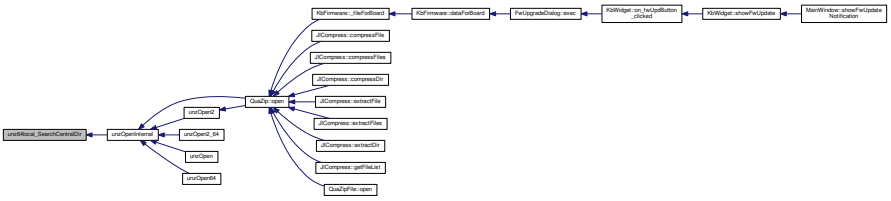
Definition at line 416 of file unzip.c.

References ALLOC, BUFREADCOMMENT, TRYFREE, ZLIB_FILEFUNC_SEEK_END, ZLIB_FILEFUNC_SEEK_-SET, ZREAD64, ZSEEK64, and ZTELL64.

Referenced by unzOpenInternal().

```
417 {
418     unsigned char* buf;
419     ZPOS64_T uSizeFile;
420     ZPOS64_T uBackRead;
421     ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
422     ZPOS64_T uPosFound=0;
423
424     if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
425         return 0;
426
427
428     uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);
429
430     if (uMaxBack>uSizeFile)
431         uMaxBack = uSizeFile;
432
433     buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
434     if (buf==NULL)
435         return 0;
436
437     uBackRead = 4;
438     while (uBackRead<uMaxBack)
439     {
440         uLong uReadSize;
441         ZPOS64_T uReadPos ;
442         int i;
443         if (uBackRead+BUFREADCOMMENT>uMaxBack)
444             uBackRead = uMaxBack;
445         else
446             uBackRead+=BUFREADCOMMENT;
447         uReadPos = uSizeFile-uBackRead ;
448
449         uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
450                      (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
451         if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
452             break;
453
454         if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
455             break;
456
457         for (i=(int)uReadSize-3; (i--)>0;)
458             if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) &&
459                 ((*(buf+i+2))==0x05) && ((*(buf+i+3))==0x06))
460             {
461                 uPosFound = uReadPos+i;
462                 break;
463             }
464
465         if (uPosFound!=0)
466             break;
467     }
468     TRYFREE(buf);
469     return uPosFound;
470 }
```

Here is the caller graph for this function:



13.160.4.15 static ZPOS64_T unz64local_SearchCentralDir64 (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream*) [static]

Definition at line 481 of file unzip.c.

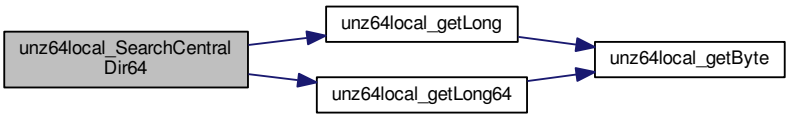
References ALLOC, BUFREADCOMMENT, TRYFREE, unz64local_getLong(), unz64local_getLong64(), UNZ_OK, ZLIB_FILEFUNC_SEEK_END, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, ZSEEK64, and ZTELL64.

Referenced by unzOpenInternal().

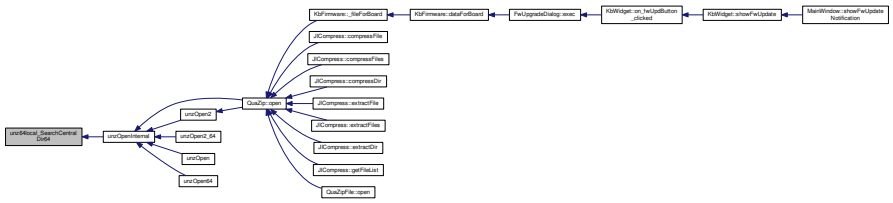
```
483 {
484     unsigned char* buf;
485     ZPOS64_T uSizeFile;
486     ZPOS64_T uBackRead;
487     ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
488     ZPOS64_T uPosFound=0;
489     uLong uL;
490                 ZPOS64_T relativeOffset;
491
492     if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
493         return 0;
494
495
496     uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);
497
498     if (uMaxBack>uSizeFile)
499         uMaxBack = uSizeFile;
500
501     buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
502     if (buf==NULL)
503         return 0;
504
505     uBackRead = 4;
506     while (uBackRead<uMaxBack)
507     {
508         uLong uReadSize;
509         ZPOS64_T uReadPos;
510         int i;
511         if (uBackRead+BUFREADCOMMENT>uMaxBack)
512             uBackRead = uMaxBack;
513         else
514             uBackRead+=BUFREADCOMMENT;
515         uReadPos = uSizeFile-uBackRead ;
516
517         uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
518                      (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
519         if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
520             break;
521
522         if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
523             break;
524
525         for (i=(int)uReadSize-3; (i--)>0;)
526             if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) &&
527                 ((*(buf+i+2))==0x06) && ((*(buf+i+3))==0x07))
528             {
529                 uPosFound = uReadPos+i;
530                 break;
531             }
532
533         if (uPosFound!=0)
534             break;
535     }
536     TRYFREE(buf);
537     if (uPosFound == 0)
538         return 0;
539
540     /* Zip64 end of central directory locator */
541     if (ZSEEK64(*pzlib_filefunc_def,filestream, uPosFound,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
542         return 0;
543
544     /* the signature, already checked */
545     if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    UNZ_OK)
546         return 0;
547
548     /* number of the disk with the start of the zip64 end of  central directory */
549     if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    UNZ_OK)
550         return 0;
551     if (uL != 0)
552         return 0;
```

```
553
554     /* relative offset of the zip64 end of central directory record */
555     if (unz64local_getLong64(pzlib_filefunc_def,filestream,&relativeOffset)!=
    UNZ_OK)
556         return 0;
557
558     /* total number of disks */
559     if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    UNZ_OK)
560         return 0;
561     if (uL != 1)
562         return 0;
563
564     /* Goto end of central directory record */
565     if (ZSEEK64(*pzlib_filefunc_def,filestream, relativeOffset,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
566         return 0;
567
568      /* the signature */
569     if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    UNZ_OK)
570         return 0;
571
572     if (uL != 0x06064b50)
573         return 0;
574
575     return relativeOffset;
576 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.16 int ZEXPORT unzClearFlags (unzFile *file,* unsigned *flags*)

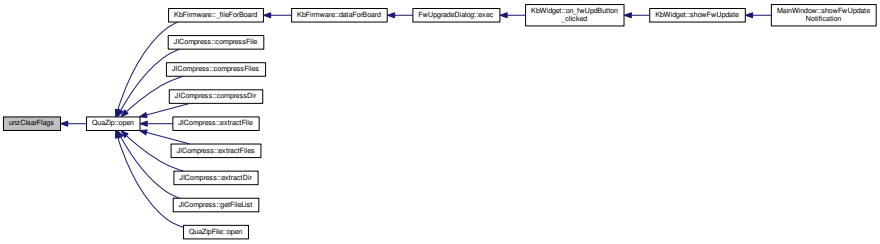
Definition at line 2147 of file unzip.c.

References unz64_s::flags, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZip::open().

```
2148 {
2149     unz64_s* s;
2150     if (file == NULL)
2151         return UNZ_PARAMERROR;
2152     s = (unz64_s*)file;
2153     s->flags &= ~flags;
2154     return UNZ_OK;
2155 }
```

Here is the caller graph for this function:



13.160.4.17 int ZEXPORT unzClose (unzFile *file*)

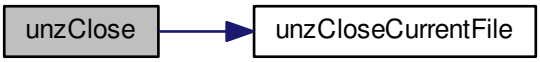
Definition at line 815 of file unzip.c.

References unz64_s::filestream, unz64_s::flags, unz64_s::pfile_in_zip_read, TRYFREE, UNZ_AUTO_CLOSE, UNZ_OK, UNZ_PARAMERROR, unzCloseCurrentFile(), unz64_s::z_filefunc, ZCLOSE64, and ZFAKECLOSE64.

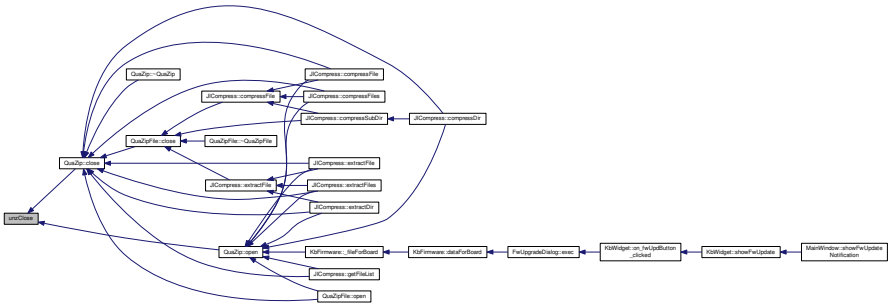
Referenced by QuaZip::close(), and QuaZip::open().

```
816 {
817     unz64_s* s;
818     if (file==NULL)
819         return UNZ_PARAMERROR;
820     s=(unz64_s*)file;
821 
822     if (s->pfile_in_zip_read!=NULL)
823         unzCloseCurrentFile(file);
824 
825     if ((s->flags & UNZ_AUTO_CLOSE) != 0)
826         ZCLOSE64(s->z_filefunc, s->filestream);
827     else
828         ZFAKECLOSE64(s->z_filefunc, s->filestream);
829     TRYFREE(s);
830     return UNZ_OK;
831 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.18 int ZEXPORT unzCloseCurrentFile (unzFile *file*)

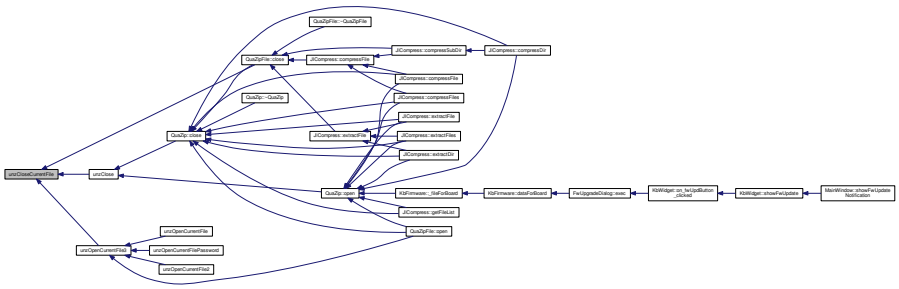
Definition at line 2012 of file unzip.c.

References file_in_zip64_read_info_s::crc32, file_in_zip64_read_info_s::crc32_wait, unz64_s::pfile_in_zip_read, file_in_zip64_read_info_s::raw, file_in_zip64_read_info_s::read_buffer, file_in_zip64_read_info_s::rest_read_uncompressed, file_in_zip64_read_info_s::stream, file_in_zip64_read_info_s::stream_initialised, TRYFREE, UNZ_CRCERROR, UNZ_OK, UNZ_PARAMERROR, and Z_BZIP2ED.

Referenced by QuaZipFile::close(), unzClose(), and unzOpenCurrentFile3().

```
2013 {
2014     int err=UNZ_OK;
2015
2016     unz64_s* s;
2017     file_in_zip64_read_info_s* pfile_in_zip_read_info;
2018     if (file==NULL)
2019         return UNZ_PARAMERROR;
2020     s=(unz64_s*)file;
2021     pfile_in_zip_read_info=s->pfile_in_zip_read;
2022
2023     if (pfile_in_zip_read_info==NULL)
2024         return UNZ_PARAMERROR;
2025
2026
2027     if ((pfile_in_zip_read_info->rest_read_uncompressed == 0) &&
2028         (!pfile_in_zip_read_info->raw))
2029     {
2030         if (pfile_in_zip_read_info->crc32 != pfile_in_zip_read_info->
    crc32_wait)
2031             err=UNZ_CRCERROR;
2032     }
2033
2034
2035     TRYFREE(pfile_in_zip_read_info->read_buffer);
2036     pfile_in_zip_read_info->read_buffer = NULL;
2037     if (pfile_in_zip_read_info->stream_initialised == Z_DEFLATED)
2038         inflateEnd(&pfile_in_zip_read_info->stream);
2039 #ifdef HAVE_BZIP2
2040     else if (pfile_in_zip_read_info->stream_initialised ==
    Z_BZIP2ED)
2041         BZ2_bzDecompressEnd(&pfile_in_zip_read_info->bstream);
2042 #endif
2043
2044
2045     pfile_in_zip_read_info->stream_initialised = 0;
2046     TRYFREE(pfile_in_zip_read_info);
2047
2048     s->pfile_in_zip_read=NULL;
2049
2050     return err;
2051 }
```

Here is the caller graph for this function:



13.160.4.19 int ZEXPORT unzeof (unzFile *file*)

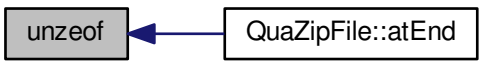
Definition at line 1932 of file unzip.c.

References unz64_s::pfile_in_zip_read, file_in_zip64_read_info_s::rest_read_uncompressed, and UNZ_PARAMERROR.

Referenced by QuaZipFile::atEnd().

```
1933 {
1934     unz64_s* s;
1935     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1936     if (file==NULL)
1937         return UNZ_PARAMERROR;
1938     s=(unz64_s*)file;
1939     pfile_in_zip_read_info=s->pfile_in_zip_read;
1940
1941     if (pfile_in_zip_read_info==NULL)
1942         return UNZ_PARAMERROR;
1943
1944     if (pfile_in_zip_read_info->rest_read_uncompressed == 0)
1945         return 1;
1946     else
1947         return 0;
1948 }
```

Here is the caller graph for this function:



13.160.4.20 int ZEXPORT unzGetCurrentFileInfo (unzFile *file*, unz_file_info * *pfile_info*, char * *szFileName*, uLong *fileNameBufferSize*, void * *extraField*, uLong *extraFieldBufferSize*, char * *szComment*, uLong *commentBufferSize*)

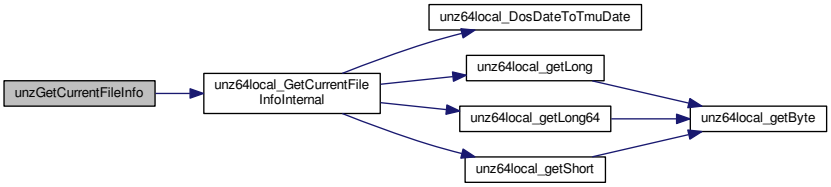
Definition at line 1146 of file unzip.c.

References unz_file_info64_s::compressed_size, unz_file_info_s::compressed_size, unz_file_info64_s::compression_method, unz_file_info_s::compression_method, unz_file_info64_s::crc, unz_file_info_s::crc, unz_file_info64_s::disk_num_start, unz_file_info_s::disk_num_start, unz_file_info64_s::dosDate, unz_file_info_s::dosDate,

unz_file_info64_s::external_fa, unz_file_info_s::external_fa, unz_file_info64_s::flag, unz_file_info_s::flag, unz_file_info64_s::internal_fa, unz_file_info_s::internal_fa, unz_file_info64_s::size_file_comment, unz_file_info_s::size_file_comment, unz_file_info64_s::size_file_extra, unz_file_info_s::size_file_extra, unz_file_info64_s::size_filename, unz_file_info_s::size_filename, unz_file_info64_s::tmu_date, unz_file_info_s::tmu_date, unz_file_info64_s::uncompressed_size, unz_file_info_s::uncompressed_size, unz64local_GetCurrentFileInfoInternal(), UNZ_OK, unz_file_info64_s::version, unz_file_info_s::version, unz_file_info64_s::version_needed, and unz_file_info_s::version_needed.

```
1151 {
1152     int err;
1153     unz_file_info64 file_info64;
1154     err = unz64local_GetCurrentFileInfoInternal(file,&file_info64,NULL
     ,
1155                                                 szFileName,fileNameBufferSize,
1156                                                 extraField,extraFieldBufferSize,
1157                                                 szComment,commentBufferSize);
1158     if (err==UNZ_OK && pfile_info != NULL)
1159     {
1160         pfile_info->version = file_info64.version;
1161         pfile_info->version_needed = file_info64.version_needed;
1162         pfile_info->flag = file_info64.flag;
1163         pfile_info->compression_method = file_info64.
     compression_method;
1164         pfile_info->dosDate = file_info64.dosDate;
1165         pfile_info->crc = file_info64.crc;
1166 
1167         pfile_info->size_filename = file_info64.size_filename;
1168         pfile_info->size_file_extra = file_info64.size_file_extra;
1169         pfile_info->size_file_comment = file_info64.
     size_file_comment;
1170 
1171         pfile_info->disk_num_start = file_info64.disk_num_start;
1172         pfile_info->internal_fa = file_info64.internal_fa;
1173         pfile_info->external_fa = file_info64.external_fa;
1174 
1175         pfile_info->tmu_date = file_info64.tmu_date,
1176 
1177 
1178         pfile_info->compressed_size = (uLong)file_info64.
     compressed_size;
1179         pfile_info->uncompressed_size = (uLong)file_info64.
     uncompressed_size;
1180 
1181     }
1182     return err;
1183 }
```

Here is the call graph for this function:



13.160.4.21 int ZEXPORT unzGetCurrentFileInfo64 (unzFile *file*, unz_file_info64 * *pfile_info*, char * *szFileName*, uLong *fileNameBufferSize*, void * *extraField*, uLong *extraFieldBufferSize*, char * *szComment*, uLong *commentBufferSize*)

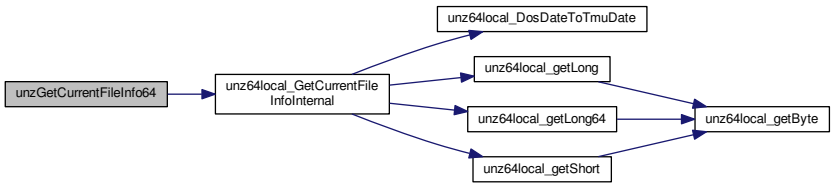
Definition at line 1134 of file unzip.c.

References unz64local_GetCurrentFileInfoInternal().

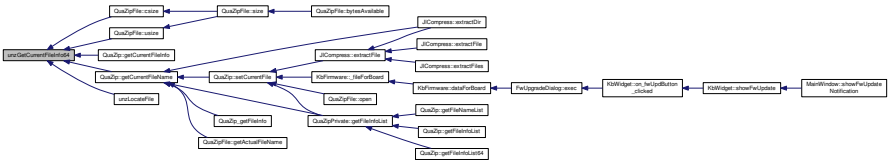
Referenced by QuaZipFile::csize(), QuaZip::getCurrentFileInfo(), QuaZip::getCurrentFileName(), unzLocateFile(), and QuaZipFile::usize().

```
1139 {
1140     return unz64local_GetCurrentFileInfoInternal(file,pfile_info,NULL,
1141                                                  szFileName,fileNameBufferSize,
1142                                                  extraField,extraFieldBufferSize,
1143                                                  szComment,commentBufferSize);
1144 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.22 ZPOS64_T ZEXPORT unzGetCurrentFileZStreamPos64 (unzFile *file*)

Definition at line 1666 of file unzip.c.

References file_in_zip64_read_info_s::byte_before_the_zipfile, unz64_s::pfile_in_zip_read, and file_in_zip64_-read_info_s::pos_in_zipfile.

```
1667 {
1668     unz64_s* s;
1669     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1670     s=(unz64_s*)file;
1671     if (file==NULL)
1672         return 0; /*UNZ_PARAMERROR; */
1673     pfile_in_zip_read_info=s->pfile_in_zip_read;
1674     if (pfile_in_zip_read_info==NULL)
1675         return 0; /*UNZ_PARAMERROR; */
1676     return pfile_in_zip_read_info->pos_in_zipfile +
1677                          pfile_in_zip_read_info->byte_before_the_zipfile;
1678 }
```

13.160.4.23 int ZEXPORT unzGetFilePos (unzFile *file,* unz_file_pos * *file_pos*)

Definition at line 1334 of file unzip.c.

References unz_file_pos_s::num_of_file, unz64_file_pos_s::num_of_file, unz_file_pos_s::pos_in_zip_directory, unz64_file_pos_s::pos_in_zip_directory, UNZ_OK, and unzGetFilePos64().

```
1337 {
1338     unz64_file_pos file_pos64;
1339     int err = unzGetFilePos64(file,&file_pos64);
1340     if (err==UNZ_OK)
1341     {
```

```
1342         file_pos->pos_in_zip_directory = (uLong)file_pos64.
     pos_in_zip_directory;
1343         file_pos->num_of_file = (uLong)file_pos64.num_of_file;
1344     }
1345     return err;
1346 }
```

Here is the call graph for this function:

13.160.4.24 int ZEXPORT unzGetFilePos64 (unzFile *file,* unz64_file_pos * *file_pos*)

Definition at line 1318 of file unzip.c.

References unz64_s::current_file_ok, unz64_s::num_file, unz64_file_pos_s::num_of_file, unz64_s::pos_in_central_dir, unz64_file_pos_s::pos_in_zip_directory, UNZ_END_OF_LIST_OF_FILE, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZipPrivate::addCurrentFileToDirectoryMap(), and unzGetFilePos().

```
1319 {
1320     unz64_s* s;
1321
1322     if (file==NULL || file_pos==NULL)
1323         return UNZ_PARAMERROR;
1324     s=(unz64_s*)file;
1325     if (!s->current_file_ok)
1326         return UNZ_END_OF_LIST_OF_FILE;
1327
1328     file_pos->pos_in_zip_directory  = s->pos_in_central_dir;
1329     file_pos->num_of_file           = s->num_file;
1330
1331     return UNZ_OK;
1332 }
```

Here is the caller graph for this function:

13.160.4.25 int ZEXPORT unzGetGlobalComment (unzFile *file,* char * *szComment,* uLong *uSizeBuf*)

Definition at line 2059 of file unzip.c.

References unz64_s::central_pos, unz64_s::filestream, unz64_s::gi, unz_global_info64_s::size_comment, UNZ_ERRNO, UNZ_PARAMERROR, unz64_s::z_filefunc, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, and ZSEEK64.

Referenced by QuaZip::getComment().

```
2060 {
2061     unz64_s* s;
2062     uLong uReadThis ;
2063     if (file==NULL)
2064         return (int)UNZ_PARAMERROR;
2065     s=(unz64_s*)file;
2066
2067     uReadThis = uSizeBuf;
2068     if (uReadThis>s->gi.size_comment)
2069         uReadThis = s->gi.size_comment;
2070
2071     if (ZSEEK64(s->z_filefunc,s->filestream,s->
     central_pos+22,ZLIB_FILEFUNC_SEEK_SET)!=0)
2072         return UNZ_ERRNO;
2073
2074     if (uReadThis>0)
2075     {
2076       *szComment='\0';
2077       if (ZREAD64(s->z_filefunc,s->filestream,szComment,uReadThis)!=uReadThis)
2078         return UNZ_ERRNO;
2079     }
2080
2081     if ((szComment != NULL) && (uSizeBuf > s->gi.size_comment))
2082         *(szComment+s->gi.size_comment)='\0';
2083     return (int)uReadThis;
2084 }
```

Here is the caller graph for this function:

13.160.4.26 int ZEXPORT unzGetGlobalInfo (unzFile *file*, unz_global_info * *pglobal_info32*)

Definition at line 848 of file unzip.c.

References unz64_s::gi, unz_global_info64_s::number_entry, unz_global_info_s::number_entry, unz_global_info64_s::size_comment, unz_global_info_s::size_comment, UNZ_OK, and UNZ_PARAMERROR.

```
849 {
850     unz64_s* s;
851     if (file==NULL)
852         return UNZ_PARAMERROR;
853     s=(unz64_s*)file;
854     /* to do : check if number_entry is not truncated */
855     pglobal_info32->number_entry = (uLong)s->gi.number_entry;
856     pglobal_info32->size_comment = s->gi.size_comment;
857     return UNZ_OK;
858 }
```

13.160.4.27 int ZEXPORT unzGetGlobalInfo64 (unzFile *file*, unz_global_info64 * *pglobal_info*)

Definition at line 838 of file unzip.c.

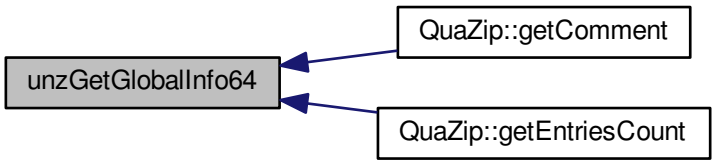
References unz64_s::gi, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZip::getComment(), and QuaZip::getEntriesCount().

```
839 {
840     unz64_s* s;
841     if (file==NULL)
```

```
842         return UNZ_PARAMERROR;
843     s=(unz64_s*)file;
844     *pglobal_info=s->gi;
845     return UNZ_OK;
846 }
```

Here is the caller graph for this function:



13.160.4.28 int ZEXPORT unzGetLocalExtrafield (unzFile *file*, voidp *buf*, unsigned *len*)

Definition at line 1964 of file unzip.c.

References file_in_zip64_read_info_s::filestream, file_in_zip64_read_info_s::offset_local_extrafield, unz64_s::pfile_in_zip_read, file_in_zip64_read_info_s::pos_local_extrafield, file_in_zip64_read_info_s::size_local_extrafield, UNZ_ERRNO, UNZ_PARAMERROR, file_in_zip64_read_info_s::z_filefunc, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, and ZSEEK64.

```
1965 {
1966     unz64_s* s;
1967     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1968     uInt read_now;
1969     ZPOS64_T size_to_read;
1970
1971     if (file==NULL)
1972         return UNZ_PARAMERROR;
1973     s=(unz64_s*)file;
1974     pfile_in_zip_read_info=s->pfile_in_zip_read;
1975
1976     if (pfile_in_zip_read_info==NULL)
1977         return UNZ_PARAMERROR;
1978
1979     size_to_read = (pfile_in_zip_read_info->size_local_extrafield -
1980                 pfile_in_zip_read_info->pos_local_extrafield);
1981
1982     if (buf==NULL)
1983         return (int)size_to_read;
1984
1985     if (len>size_to_read)
1986         read_now = (uInt)size_to_read;
1987     else
1988         read_now = (uInt)len ;
1989
1990     if (read_now==0)
1991         return 0;
1992
1993     if (ZSEEK64(pfile_in_zip_read_info->z_filefunc,
1994               pfile_in_zip_read_info->filestream,
1995               pfile_in_zip_read_info->offset_local_extrafield +
1996               pfile_in_zip_read_info->pos_local_extrafield,
1997               ZLIB_FILEFUNC_SEEK_SET)!=0)
1998         return UNZ_ERRNO;
1999
2000     if (ZREAD64(pfile_in_zip_read_info->z_filefunc,
2001               pfile_in_zip_read_info->filestream,
2002               buf,read_now)!=read_now)
2003         return UNZ_ERRNO;
2004
2005     return (int)read_now;
2006 }
```

13.160.4.29 uLong ZEXPORT unzGetOffset (unzFile *file*)

Definition at line 2102 of file unzip.c.

References unzGetOffset64().

```
2103 {
2104     ZPOS64_T offset64;
2105
2106     if (file==NULL)
2107           return 0; /*UNZ_PARAMERROR; */
2108     offset64 = unzGetOffset64(file);
2109     return (uLong)offset64;
2110 }
```

Here is the call graph for this function:

13.160.4.30 ZPOS64_T ZEXPORT unzGetOffset64 (unzFile *file*)

Definition at line 2087 of file unzip.c.

References unz64_s::current_file_ok, unz64_s::gi, unz64_s::num_file, unz_global_info64_s::number_entry, and unz64_s::pos_in_central_dir.

Referenced by unzGetOffset().

```
2088 {
2089     unz64_s* s;
2090
2091     if (file==NULL)
2092           return 0; /*UNZ_PARAMERROR; */
2093     s=(unz64_s*)file;
2094     if (!s->current_file_ok)
2095       return 0;
2096     if (s->gi.number_entry != 0 && s->gi.number_entry != 0xffff)
2097       if (s->num_file==s->gi.number_entry)
2098          return 0;
2099     return s->pos_in_central_dir;
2100 }
```

Here is the caller graph for this function:

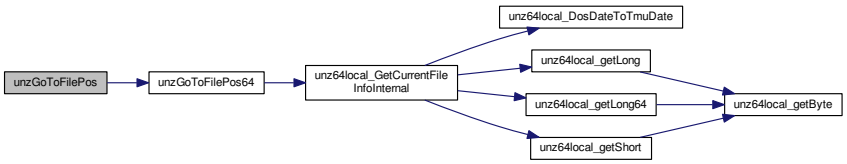
13.160.4.31 int ZEXPORT unzGoToFilePos (unzFile *file*, unz_file_pos * *file_pos*)

Definition at line 1370 of file unzip.c.

References unz_file_pos_s::num_of_file, unz64_file_pos_s::num_of_file, unz_file_pos_s::pos_in_zip_directory, unz64_file_pos_s::pos_in_zip_directory, UNZ_PARAMERROR, and unzGoToFilePos64().

```
1373 {
1374     unz64_file_pos file_pos64;
1375     if (file_pos == NULL)
1376         return UNZ_PARAMERROR;
1377
1378     file_pos64.pos_in_zip_directory = file_pos->
     pos_in_zip_directory;
1379     file_pos64.num_of_file = file_pos->num_of_file;
1380     return unzGoToFilePos64(file,&file_pos64);
1381 }
```

Here is the call graph for this function:



13.160.4.32 int ZEXPORT unzGoToFilePos64 (unzFile *file*, const unz64_file_pos * *file_pos*)

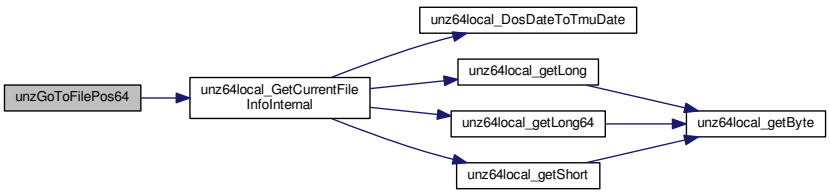
Definition at line 1348 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::num_file, unz64_file_pos_s::num_of_file, unz64_s::pos_in_central_dir, unz64_file_pos_s::pos_in_zip_directory, unz64local-_GetCurrentFileInfoInternal(), UNZ_OK, and UNZ_PARAMERROR.

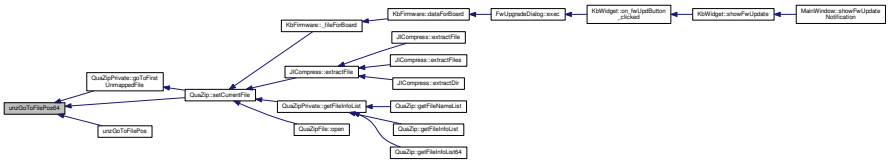
Referenced by QuaZipPrivate::goToFirstUnmappedFile(), QuaZip::setCurrentFile(), and unzGoToFilePos().

```
1349 {
1350     unz64_s* s;
1351     int err;
1352
1353     if (file==NULL || file_pos==NULL)
1354         return UNZ_PARAMERROR;
1355     s=(unz64_s*)file;
1356
1357     /* jump to the right spot */
1358     s->pos_in_central_dir = file_pos->pos_in_zip_directory;
1359     s->num_file           = file_pos->num_of_file;
1360
1361     /* set the current file */
1362     err = unz64local_GetCurrentFileInfoInternal(file,&s->
      cur_file_info,
1363                                                 &s->cur_file_info_internal,
1364                                                 NULL,0,NULL,0,NULL,0);
1365     /* return results */
1366     s->current_file_ok = (err == UNZ_OK);
1367     return err;
1368 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.33 int ZEXPORT unzGoToFirstFile (unzFile *file*)

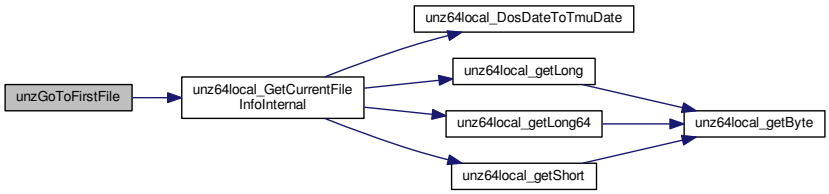
Definition at line 1188 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::num_file, unz64_s::offset_central_dir, unz64_s::pos_in_central_dir, unz64local_GetCurrentFileInfoInternal(), UNZ_OK, and UNZ_PARAMERROR.

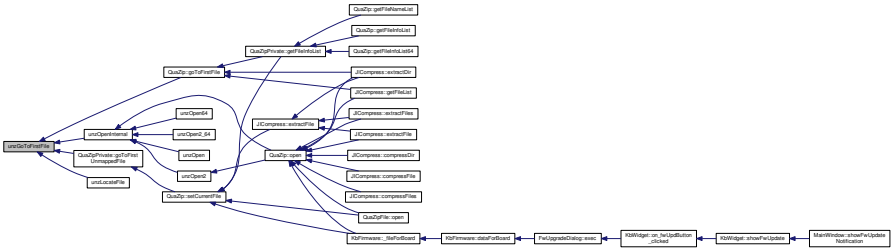
Referenced by QuaZip::goToFirstFile(), QuaZipPrivate::goToFirstUnmappedFile(), unzLocateFile(), and unzOpenInternal().

```
1189 {
1190     int err=UNZ_OK;
1191     unz64_s* s;
1192     if (file==NULL)
1193         return UNZ_PARAMERROR;
1194     s=(unz64_s*)file;
1195     s->pos_in_central_dir=s->offset_central_dir;
1196     s->num_file=0;
1197     err=unz64local_GetCurrentFileInfoInternal(file,&s->
      cur_file_info,
1198                                               &s->cur_file_info_internal,
1199                                               NULL,0,NULL,0,NULL,0);
1200     s->current_file_ok = (err == UNZ_OK);
1201     return err;
1202 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.34 int ZEXPORT unzGoToNextFile (unzFile *file*)

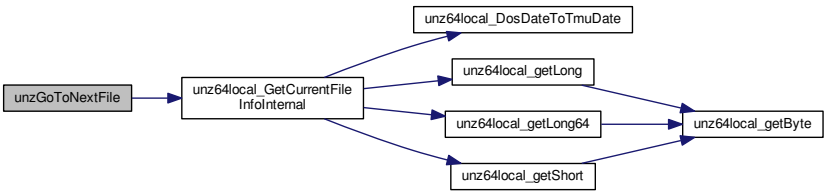
Definition at line 1209 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::gi, unz64_s::num_file, unz_global_info64_s::number_entry, unz64_s::pos_in_central_dir, unz_file_info64_s::size_file_comment, unz_file_info64_s::size_file_extra, unz_file_info64_s::size_filename, SIZECENTRALDIRITEM, unz64local_GetCurrentFileInfoInternal(), UNZ_END_OF_LIST_OF_FILE, UNZ_OK, and UNZ_PARAMERROR.

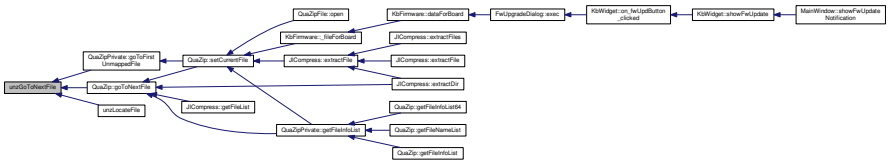
Referenced by QuaZipPrivate::goToFirstUnmappedFile(), QuaZip::goToNextFile(), and unzLocateFile().

```
1210 {
1211     unz64_s* s;
1212     int err;
1213
1214     if (file==NULL)
1215         return UNZ_PARAMERROR;
1216     s=(unz64_s*)file;
1217     if (!s->current_file_ok)
1218         return UNZ_END_OF_LIST_OF_FILE;
1219     if (s->gi.number_entry != 0xffff)    /* 2^16 files overflow hack */
1220       if (s->num_file+1==s->gi.number_entry)
1221         return UNZ_END_OF_LIST_OF_FILE;
1222
1223     s->pos_in_central_dir += SIZECENTRALDIRITEM + s->
     cur_file_info.size_filename +
1224             s->cur_file_info.size_file_extra + s->
     cur_file_info.size_file_comment ;
1225     s->num_file++;
1226     err = unz64local_GetCurrentFileInfoInternal(file,&s->
     cur_file_info,
1227                                                 &s->cur_file_info_internal,
1228                                                 NULL,0,NULL,0,NULL,0);
1229     s->current_file_ok = (err == UNZ_OK);
1230     return err;
1231 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



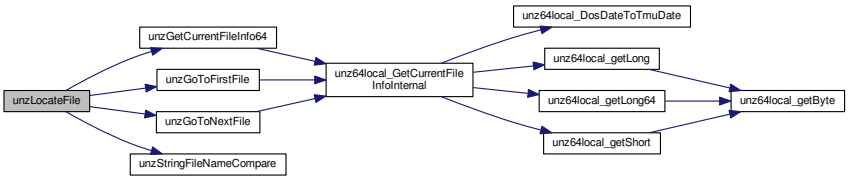
13.160.4.35 int ZEXPORT unzLocateFile (unzFile *file*, const char * *szFileName*, int *iCaseSensitivity*)

Definition at line 1242 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::num_file, unz64_s::pos_in_central_dir, UNZ_END_OF_LIST_OF_FILE, UNZ_MAXFILENAMEINZIP, UNZ_OK, UNZ_PARAMERROR, unzGetCurrentFileInfo64(), unzGoToFirstFile(), unzGoToNextFile(), and unzStringFileNameCompare().

```
1243 {
1244     unz64_s* s;
1245     int err;
1246
1247     /* We remember the 'current' position in the file so that we can jump
1248      * back there if we fail.
1249      */
1250     unz_file_info64 cur_file_infoSaved;
1251     unz_file_info64_internal cur_file_info_internalSaved;
1252     ZPOS64_T num_fileSaved;
1253     ZPOS64_T pos_in_central_dirSaved;
1254
1255
1256     if (file==NULL)
1257         return UNZ_PARAMERROR;
1258
1259     if (strlen(szFileName)>=UNZ_MAXFILENAMEINZIP)
1260         return UNZ_PARAMERROR;
1261
1262     s=(unz64_s*)file;
1263     if (!s->current_file_ok)
1264         return UNZ_END_OF_LIST_OF_FILE;
1265
1266     /* Save the current state */
1267     num_fileSaved = s->num_file;
1268     pos_in_central_dirSaved = s->pos_in_central_dir;
1269     cur_file_infoSaved = s->cur_file_info;
1270     cur_file_info_internalSaved = s->cur_file_info_internal;
1271
1272     err = unzGoToFirstFile(file);
1273
1274     while (err == UNZ_OK)
1275     {
1276         char szCurrentFileName[UNZ_MAXFILENAMEINZIP+1];
1277         err = unzGetCurrentFileInfo64(file,NULL,
1278                                       szCurrentFileName,sizeof(szCurrentFileName)-1,
1279                                       NULL,0,NULL,0);
1280         if (err == UNZ_OK)
1281         {
1282             if (unzStringFileNameCompare(szCurrentFileName,
1283                                          szFileName,iCaseSensitivity)==0)
1284                 return UNZ_OK;
1285             err = unzGoToNextFile(file);
1286         }
1287     }
1288
1289     /* We failed, so restore the state of the 'current file' to where we
1290      * were.
1291      */
1292     s->num_file = num_fileSaved ;
1293     s->pos_in_central_dir = pos_in_central_dirSaved ;
1294     s->cur_file_info = cur_file_infoSaved;
1295     s->cur_file_info_internal = cur_file_info_internalSaved;
1296     return err;
1297 }
```

Here is the call graph for this function:



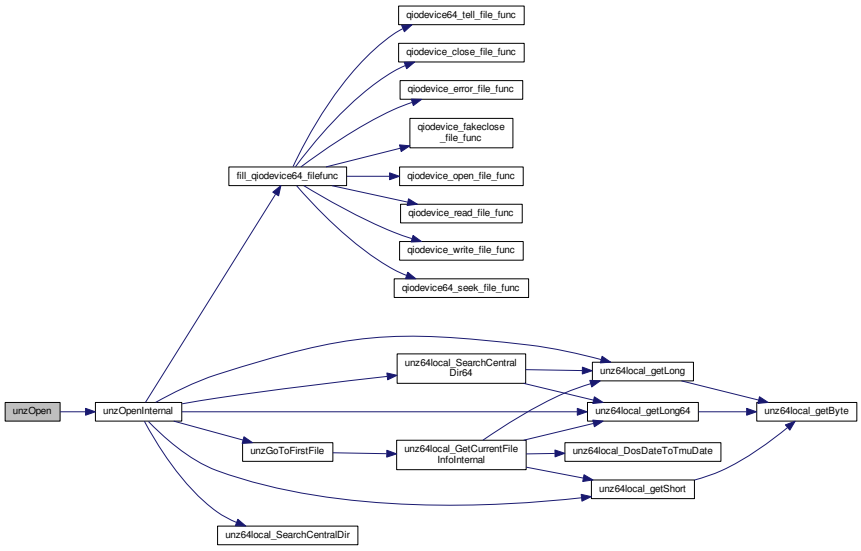
13.160.4.36 unzFile ZEXPORT unzOpen (voidpf *file*)

Definition at line 800 of file unzip.c.

References UNZ_DEFAULT_FLAGS, and unzOpenInternal().

```
801 {
802     return unzOpenInternal(file, NULL, 0, UNZ_DEFAULT_FLAGS);
803 }
```

Here is the call graph for this function:



13.160.4.37 unzFile ZEXPORT unzOpen2 (voidpf *file*, zlib_filefunc_def * *pzlib_filefunc32_def*)

Definition at line 772 of file unzip.c.

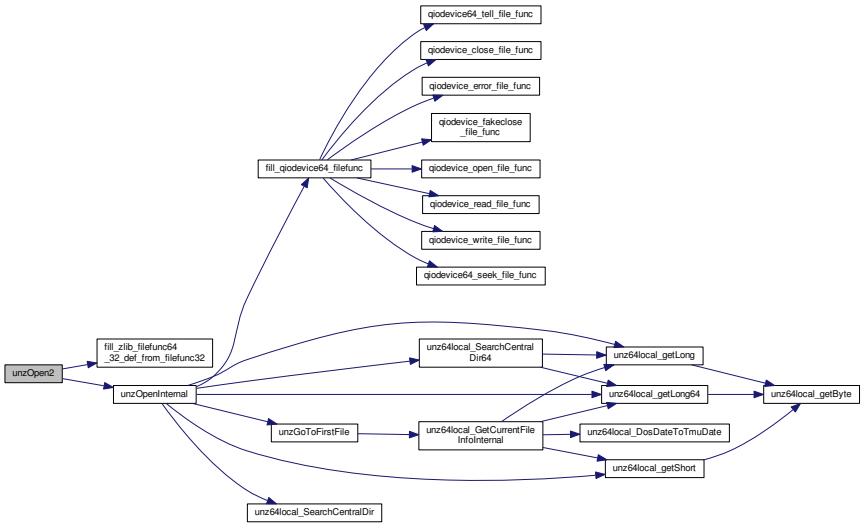
References fill_zlib_filefunc64_32_def_from_filefunc32(), UNZ_DEFAULT_FLAGS, and unzOpenInternal().

Referenced by QuaZip::open().

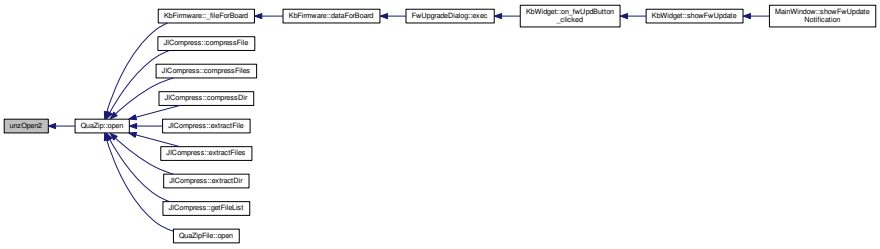
```
774 {
775     if (pzlib_filefunc32_def != NULL)
776     {
777         zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
778         fill_zlib_filefunc64_32_def_from_filefunc32(&
```

```
      zlib_filefunc64_32_def_fill,pzlib_filefunc32_def);
779         return unzOpenInternal(file, &zlib_filefunc64_32_def_fill, 0,
    UNZ_DEFAULT_FLAGS);
780     }
781     else
782         return unzOpenInternal(file, NULL, 0, UNZ_DEFAULT_FLAGS);
783 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.38 unzFile ZEXPORT unzOpen2_64 (voidpf *file*, zlib_filefunc64_def * *pzlib_filefunc_def*)

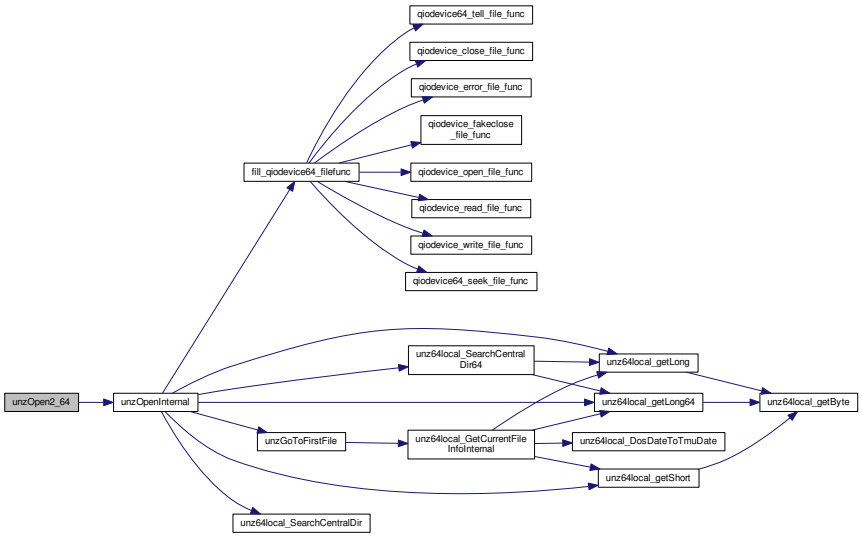
Definition at line 785 of file unzip.c.

References UNZ_DEFAULT_FLAGS, unzOpenInternal(), zlib_filefunc64_32_def_s::zfile_func64, zlib_filefunc64_-32_def_s::zseek32_file, and zlib_filefunc64_32_def_s::ztell32_file.

```
787 {
788     if (pzlib_filefunc_def != NULL)
789     {
790         zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
791         zlib_filefunc64_32_def_fill.zfile_func64 = *pzlib_filefunc_def;
792         zlib_filefunc64_32_def_fill.ztell32_file = NULL;
793         zlib_filefunc64_32_def_fill.zseek32_file = NULL;
794         return unzOpenInternal(file, &zlib_filefunc64_32_def_fill, 1,
    UNZ_DEFAULT_FLAGS);
795     }
796     else
```

```
797         return unzOpenInternal(file, NULL, 1, UNZ_DEFAULT_FLAGS);
798 }
```

Here is the call graph for this function:



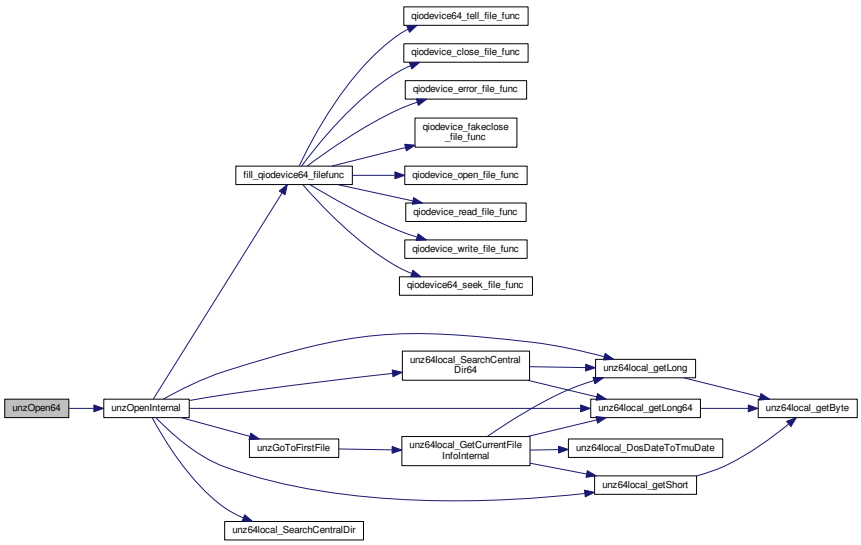
13.160.4.39 unzFile ZEXPORT unzOpen64 (voidpf *file*)

Definition at line 805 of file unzip.c.

References UNZ_DEFAULT_FLAGS, and unzOpenInternal().

```
806 {
807     return unzOpenInternal(file, NULL, 1, UNZ_DEFAULT_FLAGS);
808 }
```

Here is the call graph for this function:



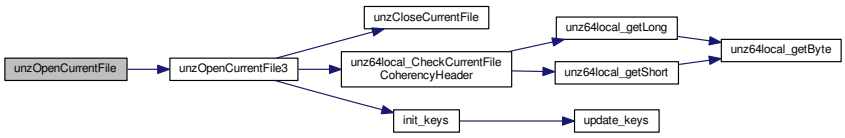
13.160.4.40 int ZEXPORT unzOpenCurrentFile (unzFile *file*)

Definition at line 1649 of file unzip.c.

References unzOpenCurrentFile3().

```
1650 {
1651     return unzOpenCurrentFile3(file, NULL, NULL, 0, NULL);
1652 }
```

Here is the call graph for this function:



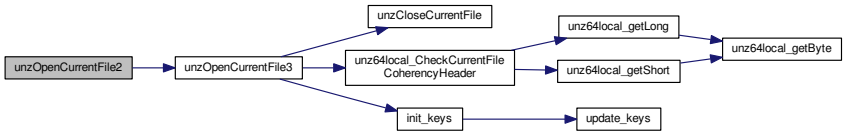
13.160.4.41 int ZEXPORT unzOpenCurrentFile2 (unzFile *file*, int * *method*, int * *level*, int *raw*)

Definition at line 1659 of file unzip.c.

References unzOpenCurrentFile3().

```
1660 {
1661     return unzOpenCurrentFile3(file, method, level, raw, NULL);
1662 }
```

Here is the call graph for this function:



13.160.4.42 int ZEXPORT unzOpenCurrentFile3 (unzFile *file*, int * *method*, int * *level*, int *raw*, const char * *password*)

Definition at line 1480 of file unzip.c.

References ALLOC, file_in_zip64_read_info_s::byte_before_the_zipfile, unz64_s::byte_before_the_zipfile, unz_file_info64_s::compressed_size, unz_file_info64_s::compression_method, file_in_zip64_read_info_s::compression_method, unz_file_info64_s::crc, file_in_zip64_read_info_s::crc32, file_in_zip64_read_info_s::crc32_wait, unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::encrypted, file_in_zip64_read_info_s::filestream, unz64_s::filestream, unz_file_info64_s::flag, init_keys(), unz64_s::keys, unz_file_info64_internal_s::offset_curfile, file_in_zip64_read_info_s::offset_local_extrafield, unz64_s::pcrc_32_tab, unz64_s::pfile_in_zip_read, file_in_zip64_read_info_s::pos_in_zipfile, file_in_zip64_read_info_s::pos_local_extrafield, file_in_zip64_read_info_s::raw, file_in_zip64_read_info_s::read_buffer, file_in_zip64_read_info_s::rest_read_compressed, file_in_zip64_read_info_s::rest_read_uncompressed, SEEK_SET, file_in_zip64_read_info_s::size_local_extrafield,

SIZEZIPLOCALHEADER, file_in_zip64_read_info_s::stream, file_in_zip64_read_info_s::stream_initialised, file_in_zip64_read_info_s::total_out_64, TRYFREE, unz_file_info64_s::uncompressed_size, unz64local_CheckCurrentFileCoherencyHeader(), UNZ_BADZIPFILE, UNZ_BUFSIZE, UNZ_INTERNALERROR, UNZ_OK, UNZ_PARAMERROR, unzCloseCurrentFile(), Z_BZIP2ED, file_in_zip64_read_info_s::z_filefunc, unz64_s::z_filefunc, zdecode, ZREAD64, and ZSEEK64.

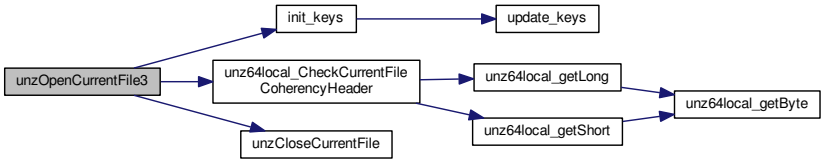
Referenced by QuaZipFile::open(), unzOpenCurrentFile(), unzOpenCurrentFile2(), and unzOpenCurrentFilePassword().

```
1482 {
1483     int err=UNZ_OK;
1484     uInt iSizeVar;
1485     unz64_s* s;
1486     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1487     ZPOS64_T offset_local_extrafield;  /* offset of the local extra field */
1488     uInt  size_local_extrafield;    /* size of the local extra field */
1489 #    ifndef NOUNCRYPT
1490     char source[12];
1491 #    else
1492     if (password != NULL)
1493         return UNZ_PARAMERROR;
1494 #    endif
1495
1496     if (file==NULL)
1497         return UNZ_PARAMERROR;
1498     s=(unz64_s*)file;
1499     if (!s->current_file_ok)
1500         return UNZ_PARAMERROR;
1501
1502     if (s->pfile_in_zip_read != NULL)
1503         unzCloseCurrentFile(file);
1504
1505     if (unz64local_CheckCurrentFileCoherencyHeader(s,&iSizeVar, &
     offset_local_extrafield,&size_local_extrafield)!=UNZ_OK)
1506         return UNZ_BADZIPFILE;
1507
1508     pfile_in_zip_read_info = (file_in_zip64_read_info_s*)
     ALLOC(sizeof(file_in_zip64_read_info_s));
1509     if (pfile_in_zip_read_info==NULL)
1510         return UNZ_INTERNALERROR;
1511
1512     pfile_in_zip_read_info->read_buffer=(char*)ALLOC(UNZ_BUFSIZE);
1513     pfile_in_zip_read_info->offset_local_extrafield = offset_local_extrafield;
1514     pfile_in_zip_read_info->size_local_extrafield = size_local_extrafield;
1515     pfile_in_zip_read_info->pos_local_extrafield=0;
1516     pfile_in_zip_read_info->raw=raw;
1517
1518     if (pfile_in_zip_read_info->read_buffer==NULL)
1519     {
1520         TRYFREE(pfile_in_zip_read_info);
1521         return UNZ_INTERNALERROR;
1522     }
1523
1524     pfile_in_zip_read_info->stream_initialised=0;
1525
1526     if (method!=NULL)
1527         *method = (int)s->cur_file_info.compression_method;
1528
1529     if (level!=NULL)
1530     {
1531         *level = 6;
1532         switch (s->cur_file_info.flag & 0x06)
1533         {
1534           case 6 : *level = 1; break;
1535           case 4 : *level = 2; break;
1536           case 2 : *level = 9; break;
1537         }
1538     }
1539
1540     if ((s->cur_file_info.compression_method!=0) &&
1541 /* #ifdef HAVE_BZIP2 */
1542         (s->cur_file_info.compression_method!=
     Z_BZIP2ED) &&
1543 /* #endif */
1544         (s->cur_file_info.compression_method!=Z_DEFLATED))
1545
1546         err=UNZ_BADZIPFILE;
1547
1548     pfile_in_zip_read_info->crc32_wait=s->cur_file_info.
     crc;
1549     pfile_in_zip_read_info->crc32=0;
1550     pfile_in_zip_read_info->total_out_64=0;
1551     pfile_in_zip_read_info->compression_method = s->
     cur_file_info.compression_method;
```

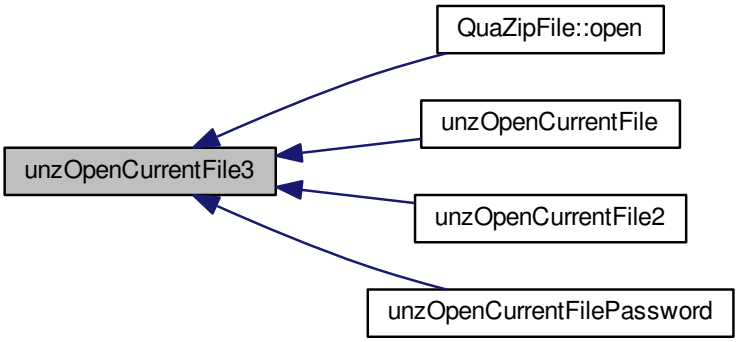
```
1552     pfile_in_zip_read_info->filestream=s->filestream;
1553     pfile_in_zip_read_info->z_filefunc=s->z_filefunc;
1554     pfile_in_zip_read_info->byte_before_the_zipfile=s->
     byte_before_the_zipfile;
1555
1556     pfile_in_zip_read_info->stream.total_out = 0;
1557
1558     if ((s->cur_file_info.compression_method==
     Z_BZIP2ED) && (!raw))
1559     {
1560 #ifdef HAVE_BZIP2
1561       pfile_in_zip_read_info->bstream.bzalloc = (void *(*) (void *, int, int))0;
1562       pfile_in_zip_read_info->bstream.bzfree = (free_func)0;
1563       pfile_in_zip_read_info->bstream.opaque = (voidpf)0;
1564       pfile_in_zip_read_info->bstream.state = (voidpf)0;
1565
1566       pfile_in_zip_read_info->stream.zalloc = (alloc_func)0;
1567       pfile_in_zip_read_info->stream.zfree = (free_func)0;
1568       pfile_in_zip_read_info->stream.opaque = (voidpf)0;
1569       pfile_in_zip_read_info->stream.next_in = (voidpf)0;
1570       pfile_in_zip_read_info->stream.avail_in = 0;
1571
1572       err=BZ2_bzDecompressInit(&pfile_in_zip_read_info->bstream, 0, 0);
1573       if (err == Z_OK)
1574         pfile_in_zip_read_info->stream_initialised=Z_BZIP2ED;
1575       else
1576       {
1577         TRYFREE(pfile_in_zip_read_info);
1578         return err;
1579       }
1580 #else
1581       pfile_in_zip_read_info->raw=1;
1582 #endif
1583     }
1584     else if ((s->cur_file_info.compression_method==Z_DEFLATED) && (!raw))
1585     {
1586       pfile_in_zip_read_info->stream.zalloc = (alloc_func)0;
1587       pfile_in_zip_read_info->stream.zfree = (free_func)0;
1588       pfile_in_zip_read_info->stream.opaque = (voidpf)0;
1589       pfile_in_zip_read_info->stream.next_in = 0;
1590       pfile_in_zip_read_info->stream.avail_in = 0;
1591
1592       err=inflateInit2(&pfile_in_zip_read_info->stream, -MAX_WBITS);
1593       if (err == Z_OK)
1594         pfile_in_zip_read_info->stream_initialised=Z_DEFLATED;
1595       else
1596       {
1597         TRYFREE(pfile_in_zip_read_info);
1598         return err;
1599       }
1600         /* windowBits is passed < 0 to tell that there is no zlib header.
1601          * Note that in this case inflate *requires* an extra "dummy" byte
1602          * after the compressed stream in order to complete decompression and
1603          * return Z_STREAM_END.
1604          * In unzip, i don't wait absolutely Z_STREAM_END because I known the
1605          * size of both compressed and uncompressed data
1606          */
1607     }
1608     pfile_in_zip_read_info->rest_read_compressed =
1609             s->cur_file_info.compressed_size ;
1610     pfile_in_zip_read_info->rest_read_uncompressed =
1611             s->cur_file_info.uncompressed_size ;
1612
1613
1614     pfile_in_zip_read_info->pos_in_zipfile =
1615             s->cur_file_info_internal.offset_curfile +
     SIZEZIPLOCALHEADER +
1616               iSizeVar;
1617
1618     pfile_in_zip_read_info->stream.avail_in = (uInt)0;
1619
1620     s->pfile_in_zip_read = pfile_in_zip_read_info;
1621                 s->encrypted = 0;
1622
1623 #    ifndef NOUNCRYPT
1624     if (password != NULL)
1625     {
1626         int i;
1627         s->pcrc_32_tab = get_crc_table();
1628         init_keys(password,s->keys,s->pcrc_32_tab);
1629         if (ZSEEK64(s->z_filefunc, s->filestream,
1630                   s->pfile_in_zip_read->pos_in_zipfile +
1631                      s->pfile_in_zip_read->
     byte_before_the_zipfile,
1632                   SEEK_SET)!=0)
1633             return UNZ_INTERNALERROR;
1634         if(ZREAD64(s->z_filefunc, s->filestream,source, 12)<12)
```

```
1635             return UNZ_INTERNALERROR;
1636
1637         for (i = 0; i<12; i++)
1638             zdecode(s->keys,s->pcrc_32_tab,source[i]);
1639
1640         s->pfile_in_zip_read->pos_in_zipfile+=12;
1641         s->encrypted=1;
1642     }
1643 #    endif
1644
1645
1646     return UNZ_OK;
1647 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



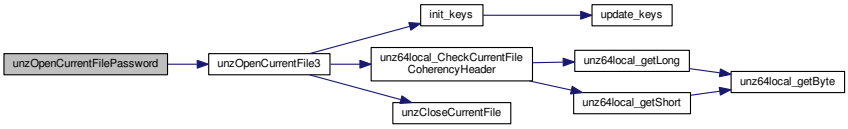
13.160.4.43 int ZEXPORT unzOpenCurrentFilePassword (unzFile *file*, const char * *password*)

Definition at line 1654 of file unzip.c.

References unzOpenCurrentFile3().

```
1655 {
1656     return unzOpenCurrentFile3(file, NULL, NULL, 0, password);
1657 }
```

Here is the call graph for this function:



13.160.4.44 unzFile unzOpenInternal (voidpf *file*, zlib_filefunc64_32_def * *pzlib_filefunc64_32_def*, int *is64bitOpenFunction*, unsigned *flags*)

Definition at line 587 of file unzip.c.

References ALLOC, unz64_s::byte_before_the_zipfile, unz64_s::central_pos, unz64_s::encrypted, unz64_s::filestream, fill_qiodevice64_filefunc(), unz64_s::flags, unz64_s::gi, unz64_s::is64bitOpenFunction, unz64_s::isZip64, unz_global_info64_s::number_entry, unz64_s::offset_central_dir, unz64_s::pfile_in_zip_read, unz64_s::size_central_dir, unz_global_info64_s::size_comment, unz64local_getLong(), unz64local_getLong64(), unz64local_getShort(), unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), UNZ_AUTO_CLOSE, UNZ_BADZIPFILE, unz_copyright, UNZ_ERRNO, UNZ_OK, unzGoToFirstFile(), unz64_s::z_filefunc, ZCLOSE64, ZFAKECLOSE64, zlib_filefunc64_32_def_s::zfile_func64, ZLIB_FILEFUNC_MODE_EXISTING, ZLIB_FILEFUNC_MODE_READ, ZLIB_FILEFUNC_SEEK_SET, ZOPEN64, zlib_filefunc64_32_def_s::zseek32_file, ZSEEK64, and zlib_filefunc64_32_def_s::ztell32_file.

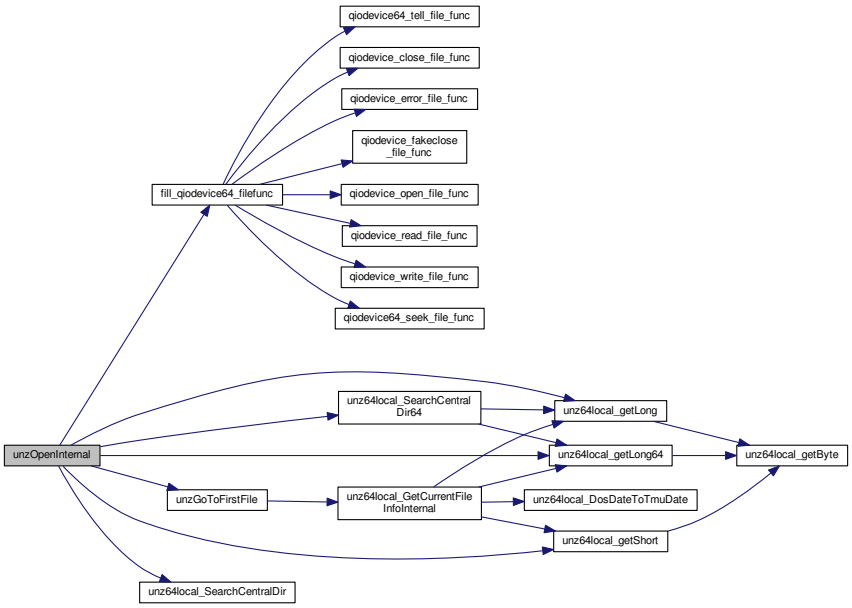
Referenced by QuaZip::open(), unzOpen(), unzOpen2(), unzOpen2_64(), and unzOpen64().

```
590 {
591     unz64_s us;
592     unz64_s *s;
593     ZPOS64_T central_pos;
594     uLong   uL;
595 
596     uLong number_disk;          /* number of the current dist, used for
597                                    spaning ZIP, unsupported, always 0*/
598     uLong number_disk_with_CD;  /* number the the disk with central dir, used
599                                    for spaning ZIP, unsupported, always 0*/
600     ZPOS64_T number_entry_CD;       /* total number of entries in
601                                    the central dir
602                                    (same than number_entry on nospan) */
603 
604     int err=UNZ_OK;
605 
606     if (unz_copyright[0]!=' ')
607         return NULL;
608 
609     us.flags = flags;
610     us.z_filefunc.zseek32_file = NULL;
611     us.z_filefunc.ztell32_file = NULL;
612     if (pzlib_filefunc64_32_def==NULL)
613         fill_qiodevice64_filefunc(&us.z_filefunc.
    zfile_func64);
614     else
615         us.z_filefunc = *pzlib_filefunc64_32_def;
616     us.is64bitOpenFunction = is64bitOpenFunction;
617 
618 
619 
620     us.filestream = ZOPEN64(us.z_filefunc,
621                                                 file,
622                                                 ZLIB_FILEFUNC_MODE_READ |
623                                                 ZLIB_FILEFUNC_MODE_EXISTING);
624     if (us.filestream==NULL)
625         return NULL;
626 
627     central_pos = unz64local_SearchCentralDir64(&us.
    z_filefunc,us.filestream);
628     if (central_pos)
629     {
630         uLong uS;
631         ZPOS64_T uL64;
632 
```

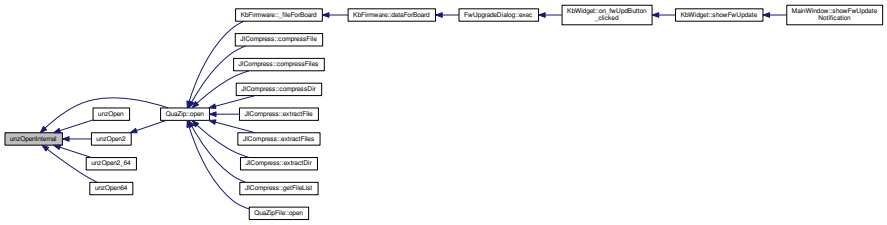
```
633         us.isZip64 = 1;
634
635         if (ZSEEK64(us.z_filefunc, us.filestream,
636                                         central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
637         err=UNZ_ERRNO;
638
639         /* the signature, already checked */
640         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
641             err=UNZ_ERRNO;
642
643         /* size of zip64 end of central directory record */
644         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&uL64)!=UNZ_OK)
645             err=UNZ_ERRNO;
646
647         /* version made by */
648         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uS)!=UNZ_OK)
649             err=UNZ_ERRNO;
650
651         /* version needed to extract */
652         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uS)!=UNZ_OK)
653             err=UNZ_ERRNO;
654
655         /* number of this disk */
656         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&number_disk)!=UNZ_OK)
657             err=UNZ_ERRNO;
658
659         /* number of the disk with the start of the central directory */
660         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&number_disk_with_CD)!=UNZ_OK)
661             err=UNZ_ERRNO;
662
663         /* total number of entries in the central directory on this disk */
664         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&us.gi.number_entry)!=UNZ_OK)
665             err=UNZ_ERRNO;
666
667         /* total number of entries in the central directory */
668         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&number_entry_CD)!=UNZ_OK)
669             err=UNZ_ERRNO;
670
671         if ((number_entry_CD!=us.gi.number_entry) ||
672             (number_disk_with_CD!=0) ||
673             (number_disk!=0))
674             err=UNZ_BADZIPFILE;
675
676         /* size of the central directory */
677         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&us.size_central_dir)!=UNZ_OK)
678             err=UNZ_ERRNO;
679
680         /* offset of start of central directory with respect to the
681           starting disk number */
682         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&us.offset_central_dir)!=UNZ_OK)
683             err=UNZ_ERRNO;
684
685         us.gi.size_comment = 0;
686       }
687       else
688       {
689         central_pos = unz64local_SearchCentralDir(&us.
    z_filefunc,us.filestream);
690         if (central_pos==0)
691             err=UNZ_ERRNO;
692
693         us.isZip64 = 0;
694
695         if (ZSEEK64(us.z_filefunc, us.filestream,
696                                         central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
697             err=UNZ_ERRNO;
698
699         /* the signature, already checked */
700         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
701             err=UNZ_ERRNO;
702
703         /* number of this disk */
704         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&number_disk)!=UNZ_OK)
705             err=UNZ_ERRNO;
706
```

```
707         /* number of the disk with the start of the central directory */
708         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&number_disk_with_CD)!=UNZ_OK)
709             err=UNZ_ERRNO;
710
711         /* total number of entries in the central dir on this disk */
712         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
713             err=UNZ_ERRNO;
714         us.gi.number_entry = uL;
715
716         /* total number of entries in the central dir */
717         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
718             err=UNZ_ERRNO;
719         number_entry_CD = uL;
720
721         if ((number_entry_CD!=us.gi.number_entry) ||
722             (number_disk_with_CD!=0) ||
723             (number_disk!=0))
724             err=UNZ_BADZIPFILE;
725
726         /* size of the central directory */
727         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
728             err=UNZ_ERRNO;
729         us.size_central_dir = uL;
730
731         /* offset of start of central directory with respect to the
732             starting disk number */
733         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
734             err=UNZ_ERRNO;
735         us.offset_central_dir = uL;
736
737         /* zipfile comment length */
738         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&us.gi.size_comment)!=UNZ_OK)
739             err=UNZ_ERRNO;
740     }
741
742     if ((central_pos<us.offset_central_dir+us.size_central_dir) &&
743         (err==UNZ_OK))
744         err=UNZ_BADZIPFILE;
745
746     if (err!=UNZ_OK)
747     {
748         if ((us.flags & UNZ_AUTO_CLOSE) != 0)
749             ZCLOSE64(us.z_filefunc, us.filestream);
750         else
751             ZFAKECLOSE64(us.z_filefunc, us.filestream);
752         return NULL;
753     }
754
755     us.byte_before_the_zipfile = central_pos -
756                             (us.offset_central_dir+us.
    size_central_dir);
757     us.central_pos = central_pos;
758     us.pfile_in_zip_read = NULL;
759     us.encrypted = 0;
760
761
762     s=(unz64_s*)ALLOC(sizeof(unz64_s));
763     if( s != NULL)
764     {
765         *s=us;
766         unzGoToFirstFile((unzFile)s);
767     }
768     return (unzFile)s;
769 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.160.4.45 int ZEXPORT unzReadCurrentFile (unzFile *file,* voidp *buf,* unsigned *len*)

Definition at line 1692 of file unzip.c.

References file_in_zip64_read_info_s::byte_before_the_zipfile, file_in_zip64_read_info_s::compression_method, file_in_zip64_read_info_s::crc32, unz64_s::encrypted, file_in_zip64_read_info_s::filestream, unz64_s::keys, unz64_s::pcrc_32_tab, unz64_s::pfile_in_zip_read, file_in_zip64_read_info_s::pos_in_zipfile, file_in_zip64_read_info_s::raw, file_in_zip64_read_info_s::read_buffer, file_in_zip64_read_info_s::rest_read_compressed, file_in_zip64_read_info_s::rest_read_uncompressed, file_in_zip64_read_info_s::stream, file_in_zip64_read_info_s::total_out_64, UNZ_BUFSIZE, UNZ_END_OF_LIST_OF_FILE, UNZ_EOF, UNZ_ERRNO, UNZ_OK, UNZ_PARAMERROR, Z_BZIP2ED, file_in_zip64_read_info_s::z_filefunc, zdecode, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, and ZSEEK64.

Referenced by QuaZipFile::readData().

```
1693 {
1694     int err=UNZ_OK;
1695     uInt iRead = 0;
1696     unz64_s* s;
1697     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1698     if (file==NULL)
```

```
1699         return UNZ_PARAMERROR;
1700     s=(unz64_s*)file;
1701     pfile_in_zip_read_info=s->pfile_in_zip_read;
1702 
1703     if (pfile_in_zip_read_info==NULL)
1704         return UNZ_PARAMERROR;
1705 
1706 
1707     if (pfile_in_zip_read_info->read_buffer == NULL)
1708         return UNZ_END_OF_LIST_OF_FILE;
1709     if (len==0)
1710         return 0;
1711 
1712     pfile_in_zip_read_info->stream.next_out = (Bytef*)buf;
1713 
1714     pfile_in_zip_read_info->stream.avail_out = (uInt)len;
1715 
1716     if ((len>pfile_in_zip_read_info->rest_read_uncompressed) &&
1717         (!(pfile_in_zip_read_info->raw)))
1718         pfile_in_zip_read_info->stream.avail_out =
1719             (uInt)pfile_in_zip_read_info->rest_read_uncompressed;
1720 
1721     if ((len>pfile_in_zip_read_info->rest_read_compressed+
1722            pfile_in_zip_read_info->stream.avail_in) &&
1723          (pfile_in_zip_read_info->raw))
1724         pfile_in_zip_read_info->stream.avail_out =
1725             (uInt)pfile_in_zip_read_info->rest_read_compressed+
1726             pfile_in_zip_read_info->stream.avail_in;
1727 
1728     while (pfile_in_zip_read_info->stream.avail_out>0)
1729     {
1730         if ((pfile_in_zip_read_info->stream.avail_in==0) &&
1731             (pfile_in_zip_read_info->rest_read_compressed>0))
1732         {
1733             uInt uReadThis = UNZ_BUFSIZE;
1734             if (pfile_in_zip_read_info->rest_read_compressed<uReadThis)
1735                 uReadThis = (uInt)pfile_in_zip_read_info->rest_read_compressed;
1736             if (uReadThis == 0)
1737                 return UNZ_EOF;
1738             if (ZSEEK64(pfile_in_zip_read_info->z_filefunc,
1739                       pfile_in_zip_read_info->filestream,
1740                       pfile_in_zip_read_info->pos_in_zipfile +
1741                          pfile_in_zip_read_info->byte_before_the_zipfile,
1742                          ZLIB_FILEFUNC_SEEK_SET)!=0)
1743                 return UNZ_ERRNO;
1744             if (ZREAD64(pfile_in_zip_read_info->z_filefunc,
1745                       pfile_in_zip_read_info->filestream,
1746                       pfile_in_zip_read_info->read_buffer,
1747                       uReadThis)!=uReadThis)
1748                 return UNZ_ERRNO;
1749 
1750 
1751 #            ifndef NOUNCRYPT
1752             if(s->encrypted)
1753             {
1754                 uInt i;
1755                 for(i=0;i<uReadThis;i++)
1756                   pfile_in_zip_read_info->read_buffer[i] =
1757                       zdecode(s->keys,s->pcrc_32_tab,
1758                               pfile_in_zip_read_info->read_buffer[i]);
1759             }
1760 #            endif
1761 
1762 
1763             pfile_in_zip_read_info->pos_in_zipfile += uReadThis;
1764 
1765             pfile_in_zip_read_info->rest_read_compressed-=uReadThis;
1766 
1767             pfile_in_zip_read_info->stream.next_in =
1768                 (Bytef*)pfile_in_zip_read_info->read_buffer;
1769             pfile_in_zip_read_info->stream.avail_in = (uInt)uReadThis;
1770         }
1771 
1772         if ((pfile_in_zip_read_info->compression_method==0) || (pfile_in_zip_read_info->
    raw))
1773         {
1774             uInt uDoCopy,i ;
1775 
1776             if ((pfile_in_zip_read_info->stream.avail_in == 0) &&
1777                 (pfile_in_zip_read_info->rest_read_compressed == 0))
1778                 return (iRead==0) ? UNZ_EOF : iRead;
1779 
1780             if (pfile_in_zip_read_info->stream.avail_out <
1781                             pfile_in_zip_read_info->stream.avail_in)
1782                 uDoCopy = pfile_in_zip_read_info->stream.avail_out ;
1783             else
1784                 uDoCopy = pfile_in_zip_read_info->stream.avail_in ;
```

```
1785
1786            for (i=0;i<uDoCopy;i++)
1787                *(pfile_in_zip_read_info->stream.next_out+i) =
1788                        *(pfile_in_zip_read_info->stream.next_in+i);
1789
1790            pfile_in_zip_read_info->total_out_64 = pfile_in_zip_read_info->
    total_out_64 + uDoCopy;
1791
1792            pfile_in_zip_read_info->crc32 = crc32(pfile_in_zip_read_info->
    crc32,
1793                                  pfile_in_zip_read_info->stream.next_out,
1794                                  uDoCopy);
1795            pfile_in_zip_read_info->rest_read_uncompressed-=uDoCopy;
1796            pfile_in_zip_read_info->stream.avail_in -= uDoCopy;
1797            pfile_in_zip_read_info->stream.avail_out -= uDoCopy;
1798            pfile_in_zip_read_info->stream.next_out += uDoCopy;
1799            pfile_in_zip_read_info->stream.next_in += uDoCopy;
1800            pfile_in_zip_read_info->stream.total_out += uDoCopy;
1801            iRead += uDoCopy;
1802        }
1803        else if (pfile_in_zip_read_info->compression_method==
    Z_BZIP2ED)
1804        {
1805 #ifdef HAVE_BZIP2
1806            uLong uTotalOutBefore,uTotalOutAfter;
1807            const Bytef *bufBefore;
1808            uLong uOutThis;
1809
1810            pfile_in_zip_read_info->bstream.next_in        = (char*)pfile_in_zip_read_info->
    stream.next_in;
1811            pfile_in_zip_read_info->bstream.avail_in       = pfile_in_zip_read_info->
    stream.avail_in;
1812            pfile_in_zip_read_info->bstream.total_in_lo32  = pfile_in_zip_read_info->
    stream.total_in;
1813            pfile_in_zip_read_info->bstream.total_in_hi32  = 0;
1814            pfile_in_zip_read_info->bstream.next_out       = (char*)pfile_in_zip_read_info->
    stream.next_out;
1815            pfile_in_zip_read_info->bstream.avail_out      = pfile_in_zip_read_info->
    stream.avail_out;
1816            pfile_in_zip_read_info->bstream.total_out_lo32 = pfile_in_zip_read_info->
    stream.total_out;
1817            pfile_in_zip_read_info->bstream.total_out_hi32 = 0;
1818
1819            uTotalOutBefore = pfile_in_zip_read_info->bstream.total_out_lo32;
1820            bufBefore = (const Bytef *)pfile_in_zip_read_info->bstream.next_out;
1821
1822            err=BZ2_bzDecompress(&pfile_in_zip_read_info->bstream);
1823
1824            uTotalOutAfter = pfile_in_zip_read_info->bstream.total_out_lo32;
1825            uOutThis = uTotalOutAfter-uTotalOutBefore;
1826
1827            pfile_in_zip_read_info->total_out_64 = pfile_in_zip_read_info->
    total_out_64 + uOutThis;
1828
1829            pfile_in_zip_read_info->crc32 = crc32(pfile_in_zip_read_info->
    crc32,bufBefore, (uInt)(uOutThis));
1830            pfile_in_zip_read_info->rest_read_uncompressed -= uOutThis;
1831            iRead += (uInt)(uTotalOutAfter - uTotalOutBefore);
1832
1833            pfile_in_zip_read_info->stream.next_in   = (Bytef*)pfile_in_zip_read_info->bstream.
    next_in;
1834            pfile_in_zip_read_info->stream.avail_in  = pfile_in_zip_read_info->bstream.avail_in;
1835            pfile_in_zip_read_info->stream.total_in  = pfile_in_zip_read_info->bstream.total_in_lo32;
1836            pfile_in_zip_read_info->stream.next_out  = (Bytef*)pfile_in_zip_read_info->bstream.
    next_out;
1837            pfile_in_zip_read_info->stream.avail_out = pfile_in_zip_read_info->bstream.avail_out;
1838            pfile_in_zip_read_info->stream.total_out = pfile_in_zip_read_info->bstream.total_out_lo32
    ;
1839
1840            if (err==BZ_STREAM_END)
1841              return (iRead==0) ? UNZ_EOF : iRead;
1842            if (err!=BZ_OK)
1843              break;
1844 #endif
1845        } /* end Z_BZIP2ED */
1846        else
1847        {
1848            ZPOS64_T uTotalOutBefore,uTotalOutAfter;
1849            const Bytef *bufBefore;
1850            ZPOS64_T uOutThis;
1851            int flush=Z_SYNC_FLUSH;
1852
1853            uTotalOutBefore = pfile_in_zip_read_info->stream.total_out;
1854            bufBefore = pfile_in_zip_read_info->stream.next_out;
1855
1856            /*
1857            if ((pfile_in_zip_read_info->rest_read_uncompressed ==
```

```
1858                     pfile_in_zip_read_info->stream.avail_out) &&
1859                 (pfile_in_zip_read_info->rest_read_compressed == 0))
1860                 flush = Z_FINISH;
1861             */
1862             err=inflate(&pfile_in_zip_read_info->stream,flush);
1863 
1864             if ((err>=0) && (pfile_in_zip_read_info->stream.msg!=NULL))
1865               err = Z_DATA_ERROR;
1866 
1867             uTotalOutAfter = pfile_in_zip_read_info->stream.total_out;
1868             uOutThis = uTotalOutAfter-uTotalOutBefore;
1869 
1870             pfile_in_zip_read_info->total_out_64 = pfile_in_zip_read_info->
      total_out_64 + uOutThis;
1871 
1872             pfile_in_zip_read_info->crc32 =
1873                 crc32(pfile_in_zip_read_info->crc32,bufBefore,
1874                         (uInt)(uOutThis));
1875 
1876             pfile_in_zip_read_info->rest_read_uncompressed -=
1877                 uOutThis;
1878 
1879             iRead += (uInt)(uTotalOutAfter - uTotalOutBefore);
1880 
1881             if (err==Z_STREAM_END)
1882                 return (iRead==0) ? UNZ_EOF : iRead;
1883             if (err!=Z_OK)
1884                 break;
1885         }
1886     }
1887 
1888     if (err==Z_OK)
1889         return iRead;
1890     return err;
1891 }
```

Here is the caller graph for this function:

13.160.4.46 int ZEXPORT unzSetFlags (unzFile *file*, unsigned *flags*)

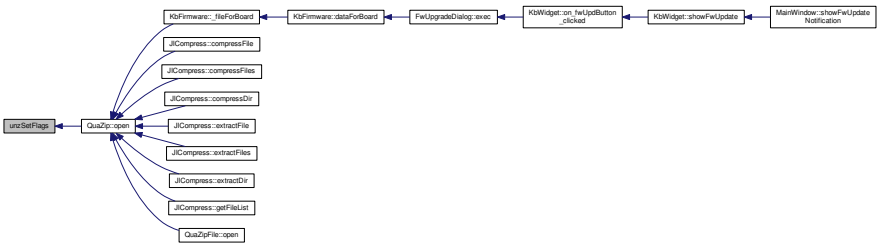
Definition at line 2136 of file unzip.c.

References unz64_s::flags, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZip::open().

```
2137 {
2138     unz64_s* s;
2139     if (file == NULL)
2140         return UNZ_PARAMERROR;
2141     s = (unz64_s*)file;
2142     s->flags |= flags;
2143     return UNZ_OK;
2144 }
```

Here is the caller graph for this function:



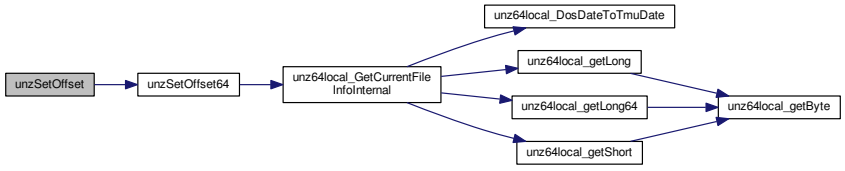
13.160.4.47 int ZEXPORT unzSetOffset (unzFile *file*, uLong *pos*)

Definition at line 2130 of file unzip.c.

References unzSetOffset64().

```
2131 {
2132     return unzSetOffset64(file,pos);
2133 }
```

Here is the call graph for this function:



13.160.4.48 int ZEXPORT unzSetOffset64 (unzFile *file*, ZPOS64_T *pos*)

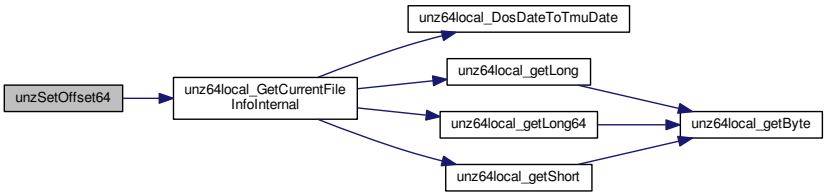
Definition at line 2112 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::gi, unz64_s::num_file, unz_global_info64_s::number_entry, unz64_s::pos_in_central_dir, unz64local_GetCurrentFileInfoInternal(), UNZ_OK, and UNZ_PARAMERROR.

Referenced by unzSetOffset().

```
2113 {
2114     unz64_s* s;
2115     int err;
2116
2117     if (file==NULL)
2118         return UNZ_PARAMERROR;
2119     s=(unz64_s*)file;
2120
2121     s->pos_in_central_dir = pos;
2122     s->num_file = s->gi.number_entry;       /* hack */
2123     err = unz64local_GetCurrentFileInfoInternal(file,&s->
      cur_file_info,
2124                                                 &s->cur_file_info_internal,
2125                                                 NULL,0,NULL,0,NULL,0);
2126     s->current_file_ok = (err == UNZ_OK);
2127     return err;
2128 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.160.4.49 int ZEXPORT unzStringFileNameCompare (const char * *fileName1*, const char * *fileName2*, int *iCaseSensitivity*)

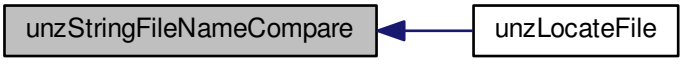
Definition at line 393 of file unzip.c.

References CASESENSITIVITYDEFAULTVALUE, and STRCMPCASENOSENTIVEFUNCTION.

Referenced by unzLocateFile().

```
397 {
398     if (iCaseSensitivity==0)
399         iCaseSensitivity=CASESENSITIVITYDEFAULTVALUE;
400
401     if (iCaseSensitivity==1)
402         return strcmp(fileName1,fileName2);
403
404     return STRCMPCASENOSENTIVEFUNCTION(fileName1,fileName2);
405 }
```

Here is the caller graph for this function:



13.160.4.50 z_off_t ZEXPORT unztell (unzFile *file*)

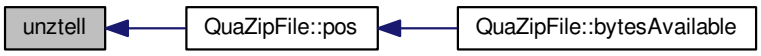
Definition at line 1897 of file unzip.c.

References unz64_s::pfile_in_zip_read, file_in_zip64_read_info_s::stream, and UNZ_PARAMERROR.

Referenced by QuaZipFile::pos().

```
1898 {
1899     unz64_s* s;
1900     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1901     if (file==NULL)
1902         return UNZ_PARAMERROR;
1903     s=(unz64_s*)file;
1904     pfile_in_zip_read_info=s->pfile_in_zip_read;
1905
1906     if (pfile_in_zip_read_info==NULL)
1907         return UNZ_PARAMERROR;
1908
1909     return (z_off_t)pfile_in_zip_read_info->stream.total_out;
1910 }
```

Here is the caller graph for this function:



13.160.4.51 ZPOS64_T ZEXPORT unztell64 (unzFile *file*)

Definition at line 1912 of file unzip.c.

References unz64_s::pfile_in_zip_read, and file_in_zip64_read_info_s::total_out_64.

```
1913 {
1914
1915     unz64_s* s;
1916     file_in_zip64_read_info_s* pfile_in_zip_read_info;
1917     if (file==NULL)
1918         return (ZPOS64_T)-1;
1919     s=(unz64_s*)file;
1920     pfile_in_zip_read_info=s->pfile_in_zip_read;
1921
1922     if (pfile_in_zip_read_info==NULL)
1923         return (ZPOS64_T)-1;
1924
1925     return pfile_in_zip_read_info->total_out_64;
1926 }
```

13.160.5 Variable Documentation

13.160.5.1 const char unz_copyright[ ]

Initial value:

```
=
   " unzip 1.01 Copyright 1998-2004 Gilles Vollant - http://www.winimage.com/zLibDll"
```

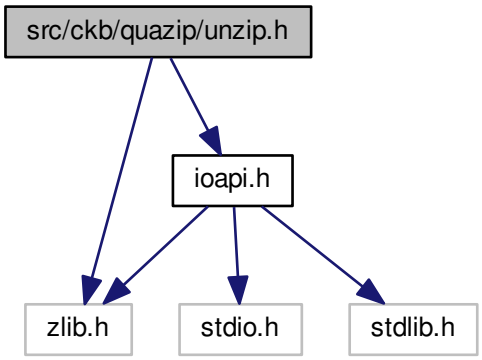
Definition at line 124 of file unzip.c.

Referenced by unzOpenInternal().

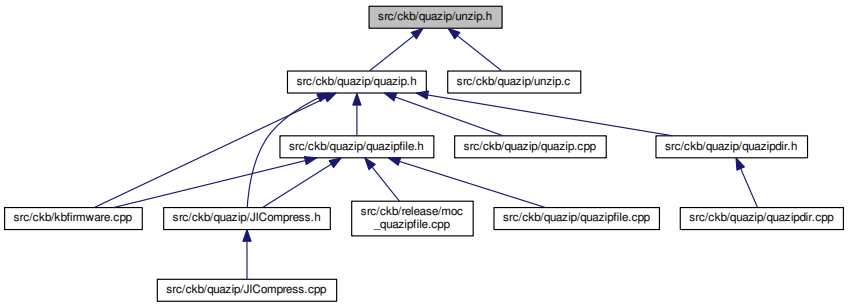
13.161 src/ckb/quazip/unzip.h File Reference

```
#include "zlib.h"
#include "ioapi.h"
```

Include dependency graph for unzip.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct tm_unz_s
- struct unz_global_info64_s
- struct unz_global_info_s
- struct unz_file_info64_s
- struct unz_file_info_s
- struct unz_file_pos_s
- struct unz64_file_pos_s

Macros

- #define Z_BZIP2ED 12

- #define UNZ_OK (0)
- #define UNZ_END_OF_LIST_OF_FILE (-100)
- #define UNZ_ERRNO (Z_ERRNO)
- #define UNZ_EOF (0)
- #define UNZ_PARAMERROR (-102)
- #define UNZ_BADZIPFILE (-103)
- #define UNZ_INTERNALERROR (-104)
- #define UNZ_CRCERROR (-105)
- #define UNZ_AUTO_CLOSE 0x01u
- #define UNZ_DEFAULT_FLAGS UNZ_AUTO_CLOSE

Typedefs

- typedef voidp unzFile
- typedef struct tm_unz_s tm_unz
- typedef struct unz_global_info64_s unz_global_info64
- typedef struct unz_global_info_s unz_global_info
- typedef struct unz_file_info64_s unz_file_info64
- typedef struct unz_file_info_s unz_file_info
- typedef struct unz_file_pos_s unz_file_pos
- typedef struct unz64_file_pos_s unz64_file_pos

Functions

- int ZEXPORT unzStringFileNameCompare _Z_OF ((const char ∗fileName1, const char ∗fileName2, int iCaseSensitivity))
- unzFile ZEXPORT unzOpen _Z_OF ((voidpf file))
- unzFile ZEXPORT unzOpen2 _Z_OF ((voidpf file, zlib_filefunc_def ∗pzlib_filefunc_def))
- unzFile ZEXPORT unzOpen2_64 _Z_OF ((voidpf file, zlib_filefunc64_def ∗pzlib_filefunc_def))
- unzFile unzOpenInternal (voidpf file, zlib_filefunc64_32_def ∗pzlib_filefunc64_32_def, int is64bitOpenFunction, unsigned flags)
- int ZEXPORT unzClose _Z_OF ((unzFile file))

  Addition for GDAL : START.
- int ZEXPORT unzGetGlobalInfo _Z_OF ((unzFile file, unz_global_info ∗pglobal_info))
- int ZEXPORT unzGetGlobalInfo64 _Z_OF ((unzFile file, unz_global_info64 ∗pglobal_info))
- int ZEXPORT unzGetGlobalComment _Z_OF ((unzFile file, char ∗szComment, uLong uSizeBuf))
- int ZEXPORT unzLocateFile _Z_OF ((unzFile file, const char ∗szFileName, int iCaseSensitivity))
- int ZEXPORT unzGetFilePos (unzFile file, unz_file_pos ∗file_pos)
- int ZEXPORT unzGoToFilePos (unzFile file, unz_file_pos ∗file_pos)
- int ZEXPORT unzGetFilePos64 (unzFile file, unz64_file_pos ∗file_pos)
- int ZEXPORT unzGoToFilePos64 (unzFile file, const unz64_file_pos ∗file_pos)
- int ZEXPORT unzGetCurrentFileInfo64 _Z_OF ((unzFile file, unz_file_info64 ∗pfile_info, char ∗szFileName, uLong fileNameBufferSize, void ∗extraField, uLong extraFieldBufferSize, char ∗szComment, uLong commentBufferSize))
- int ZEXPORT unzGetCurrentFileInfo _Z_OF ((unzFile file, unz_file_info ∗pfile_info, char ∗szFileName, uLong fileNameBufferSize, void ∗extraField, uLong extraFieldBufferSize, char ∗szComment, uLong commentBufferSize))
- int ZEXPORT unzOpenCurrentFilePassword _Z_OF ((unzFile file, const char ∗password))
- int ZEXPORT unzOpenCurrentFile2 _Z_OF ((unzFile file, int ∗method, int ∗level, int raw))
- int ZEXPORT unzOpenCurrentFile3 _Z_OF ((unzFile file, int ∗method, int ∗level, int raw, const char ∗password))
- int ZEXPORT unzReadCurrentFile _Z_OF ((unzFile file, voidp buf, unsigned len))
- ZPOS64_T ZEXPORT unzGetOffset64 (unzFile file)

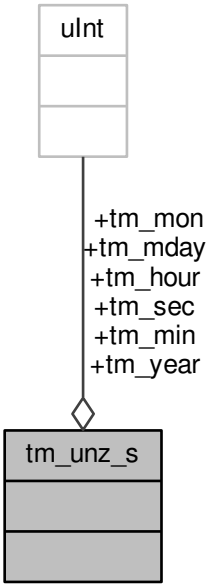
- uLong ZEXPORT unzGetOffset (unzFile file)
- int ZEXPORT unzSetOffset64 (unzFile file, ZPOS64_T pos)
- int ZEXPORT unzSetOffset (unzFile file, uLong pos)
- int ZEXPORT unzSetFlags (unzFile file, unsigned flags)
- int ZEXPORT unzClearFlags (unzFile file, unsigned flags)

13.161.1 Data Structure Documentation

13.161.1.1 struct tm_unz_s

Definition at line 92 of file unzip.h.

Collaboration diagram for tm_unz_s:



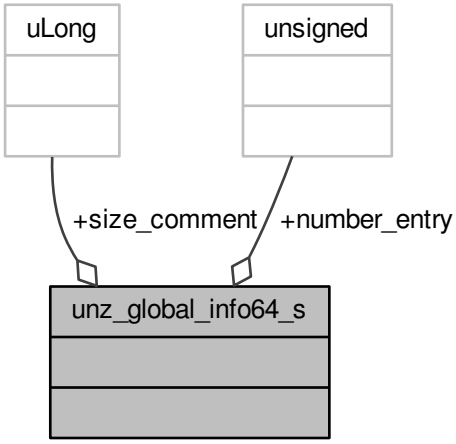
Data Fields

uInt	tm_hour	
uInt	tm_mday	
uInt	tm_min	
uInt	tm_mon	
uInt	tm_sec	
uInt	tm_year	

13.161.1.2 struct unz_global_info64_s

Definition at line 104 of file unzip.h.

Collaboration diagram for unz_global_info64_s:



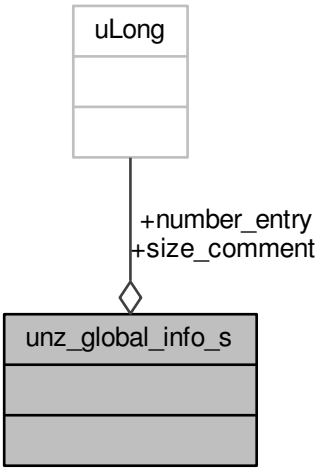
Data Fields

ZPOS64_T	number_entry	
uLong	size_comment	

13.161.1.3 struct unz_global_info_s

Definition at line 111 of file unzip.h.

Collaboration diagram for unz_global_info_s:



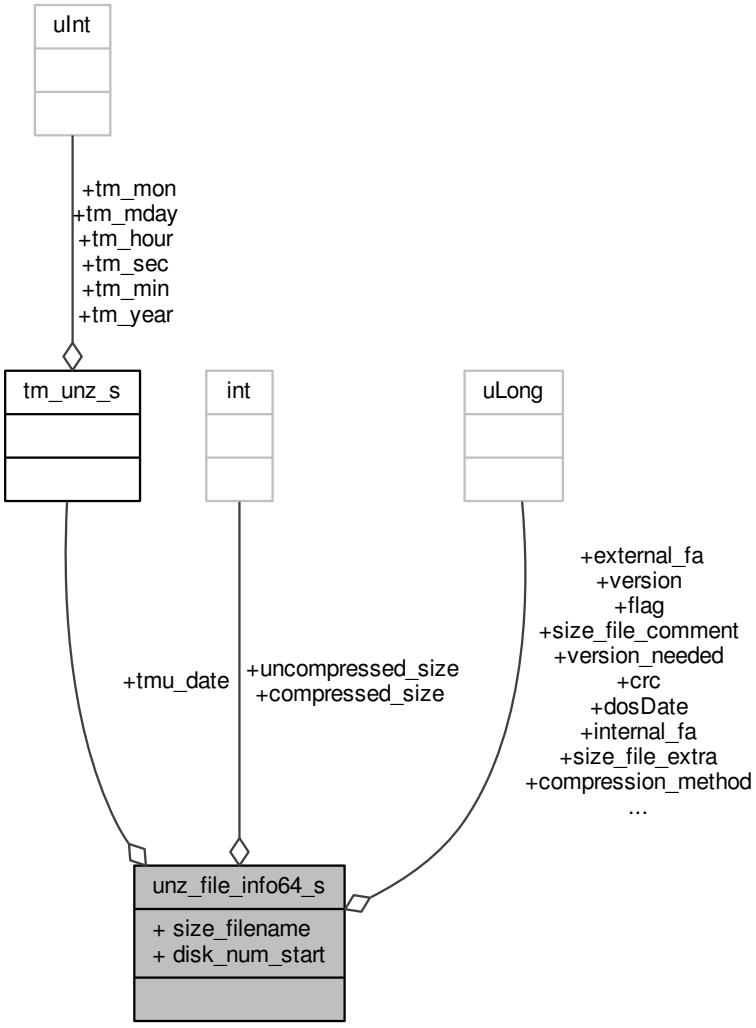
Data Fields

uLong	number_entry	
uLong	size_comment	

13.161.1.4 struct unz_file_info64_s

Definition at line 119 of file unzip.h.

Collaboration diagram for unz_file_info64_s:



Data Fields

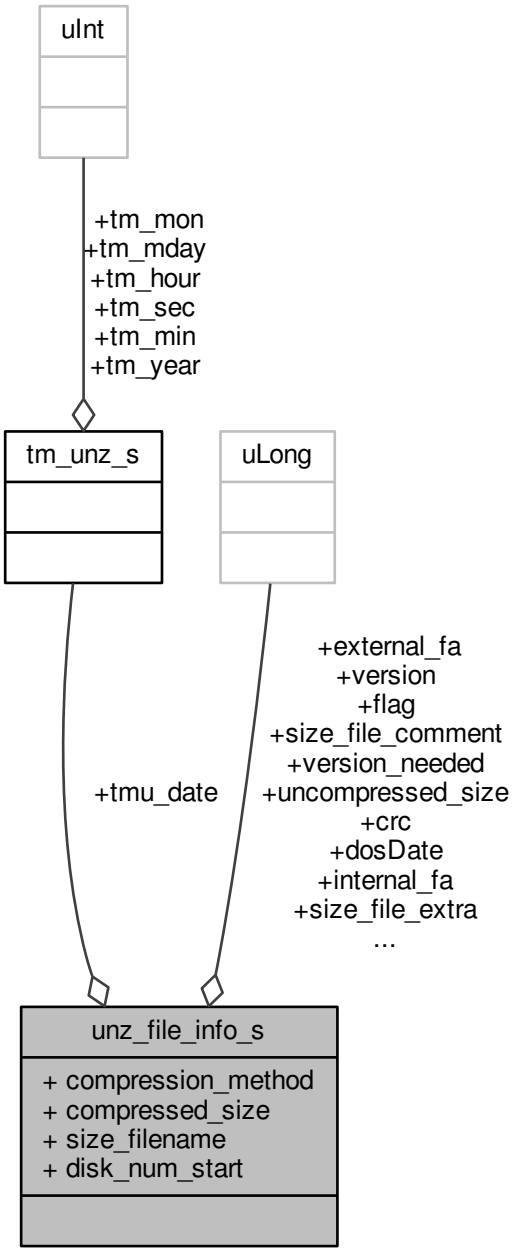
ZPOS64_T	compressed_-size	

uLong	compression_- method	
uLong	crc	
uLong	disk_num_start	
uLong	dosDate	
uLong	external_fa	
uLong	flag	
uLong	internal_fa	
uLong	size_file_- comment	
uLong	size_file_extra	
uLong	size_filename	
tm_unz	tmu_date	
ZPOS64_T	uncompressed_- size	
uLong	version	
uLong	version_needed	

13.161.1.5 struct unz_file_info_s

Definition at line 140 of file unzip.h.

Collaboration diagram for unz_file_info_s:



Data Fields

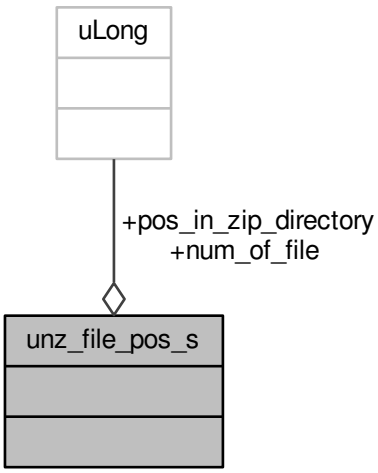
uLong	compressed_-size	

uLong	compression_-method	
uLong	crc	
uLong	disk_num_start	
uLong	dosDate	
uLong	external_fa	
uLong	flag	
uLong	internal_fa	
uLong	size_file_-comment	
uLong	size_file_extra	
uLong	size_filename	
tm_unz	tmu_date	
uLong	uncompressed_-size	
uLong	version	
uLong	version_needed	

13.161.1.6 struct unz_file_pos_s

Definition at line 278 of file unzip.h.

Collaboration diagram for unz_file_pos_s:



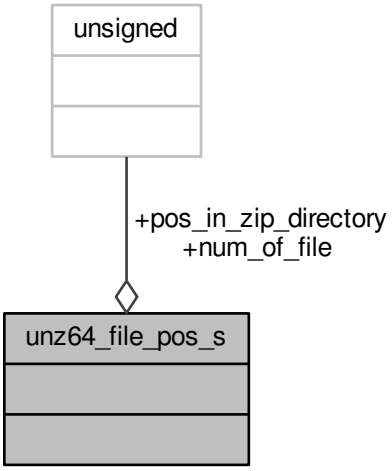
Data Fields

uLong	num_of_file	
uLong	pos_in_zip_-directory	

13.161.1.7 struct unz64_file_pos_s

Definition at line 292 of file unzip.h.

Collaboration diagram for unz64_file_pos_s:



Data Fields

ZPOS64_T	num_of_file	
ZPOS64_T	pos_in_zip_-directory	

13.161.2 Macro Definition Documentation

13.161.2.1 #define UNZ_AUTO_CLOSE 0x01u

Definition at line 88 of file unzip.h.

Referenced by QuaZip::open(), unzClose(), and unzOpenInternal().

13.161.2.2 #define UNZ_BADZIPFILE (-103)

Definition at line 84 of file unzip.h.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), unzOpenCurrentFile3(), and unzOpenInternal().

13.161.2.3 #define UNZ_CRCERROR (-105)

Definition at line 86 of file unzip.h.

Referenced by unzCloseCurrentFile().

13.161.2.4 #define UNZ_DEFAULT_FLAGS UNZ_AUTO_CLOSE

Definition at line 89 of file unzip.h.

Referenced by unzOpen(), unzOpen2(), unzOpen2_64(), and unzOpen64().

13.161.2.5 #define UNZ_END_OF_LIST_OF_FILE (-100)

Definition at line 80 of file unzip.h.

Referenced by QuaZipPrivate::goToFirstUnmappedFile(), QuaZip::goToNextFile(), unzGetFilePos64(), unzGoToNextFile(), unzLocateFile(), and unzReadCurrentFile().

13.161.2.6 #define UNZ_EOF (0)

Definition at line 82 of file unzip.h.

Referenced by unz64local_getByte(), and unzReadCurrentFile().

13.161.2.7 #define UNZ_ERRNO (Z_ERRNO)

Definition at line 81 of file unzip.h.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_getByte(), unz64local_GetCurrentFileInfoInternal(), unzGetGlobalComment(), unzGetLocalExtrafield(), unzOpenInternal(), and unzReadCurrentFile().

13.161.2.8 #define UNZ_INTERNALERROR (-104)

Definition at line 85 of file unzip.h.

Referenced by unzOpenCurrentFile3().

13.161.2.9 #define UNZ_OK (0)

Definition at line 79 of file unzip.h.

Referenced by QuaZip::close(), QuaZipFile::close(), JlCompress::compressFile(), QuaZipFile::csize(), JlCompress::extractFile(), QuaZipFile::getActualFileName(), QuaZip::getComment(), QuaZip::getCurrentFileInfo(), QuaZip::getCurrentFileName(), QuaZip::getEntriesCount(), QuaZipFile::getFileInfo(), QuaZipPrivate::getFileInfoList(), QuaZip::goToFirstFile(), QuaZipPrivate::goToFirstUnmappedFile(), QuaZip::goToNextFile(), QuaZip::open(), QuaZipFile::open(), QuaZipFile::readData(), QuaZipFilePrivate::resetZipError(), QuaZip::setCurrentFile(), QuaZipFilePrivate::setZipError(), unz64local_CheckCurrentFileCoherencyHeader(), unz64local_getByte(), unz64local_GetCurrentFileInfoInternal(), unz64local_getLong(), unz64local_getLong64(), unz64local_getShort(), unz64local_SearchCentralDir64(), unzClearFlags(), unzClose(), unzCloseCurrentFile(), unzGetCurrentFileInfo(), unzGetFilePos(), unzGetFilePos64(), unzGetGlobalInfo(), unzGetGlobalInfo64(), unzGoToFilePos64(), unzGoToFirstFile(), unzGoToNextFile(), unzLocateFile(), unzOpenCurrentFile3(), unzOpenInternal(), unzReadCurrentFile(), unzSetFlags(), unzSetOffset64(), and QuaZipFile::usize().

13.161.2.10 #define UNZ_PARAMERROR (-102)

Definition at line 83 of file unzip.h.

Referenced by QuaZip::setCurrentFile(), unz64local_GetCurrentFileInfoInternal(), unzClearFlags(), unzClose(), unzCloseCurrentFile(), unzeof(), unzGetFilePos64(), unzGetGlobalComment(), unzGetGlobalInfo(), unzGetGlobalInfo64(), unzGetLocalExtrafield(), unzGoToFilePos(), unzGoToFilePos64(), unzGoToFirstFile(), unzGoToNextFile(), unzLocateFile(), unzOpenCurrentFile3(), unzReadCurrentFile(), unzSetFlags(), unzSetOffset64(), and unztell().

13.161.2.11 #define Z_BZIP2ED 12

Definition at line 67 of file unzip.h.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unzCloseCurrentFile(), unzOpenCurrentFile3(), unzReadCurrentFile(), zip64FlushWriteBuffer(), zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), and zipWriteInFileInZip().

13.161.3 Typedef Documentation

13.161.3.1 typedef struct tm_unz_s tm_unz

13.161.3.2 typedef struct unz64_file_pos_s unz64_file_pos

13.161.3.3 typedef struct unz_file_info_s unz_file_info

13.161.3.4 typedef struct unz_file_info64_s unz_file_info64

13.161.3.5 typedef struct unz_file_pos_s unz_file_pos

13.161.3.6 typedef struct unz_global_info_s unz_global_info

13.161.3.7 typedef struct unz_global_info64_s unz_global_info64

13.161.3.8 typedef voidp unzFile

Definition at line 75 of file unzip.h.

13.161.4 Function Documentation

13.161.4.1 int ZEXPORT unzStringFileNameCompare _Z_OF ((const char *fileName1, const char *fileName2, int iCaseSensitivity))

13.161.4.2 unzFile ZEXPORT unzOpen _Z_OF ((voidpf file))

13.161.4.3 unzFile ZEXPORT unzOpen2 _Z_OF ((voidpf file, zlib_filefunc_def *pzlib_filefunc_def))

13.161.4.4 unzFile ZEXPORT unzOpen2_64 _Z_OF ((voidpf file, zlib_filefunc64_def *pzlib_filefunc_def))

13.161.4.5 int ZEXPORT unzClose _Z_OF ((unzFile file))

Addition for GDAL : END.

13.161.4.6 int ZEXPORT unzGetGlobalInfo _Z_OF ((unzFile file, unz_global_info *pglobal_info))

13.161.4.7 int ZEXPORT unzGetGlobalInfo64 _Z_OF ((unzFile file, unz_global_info64 *pglobal_info))

13.161.4.8 int ZEXPORT unzGetGlobalComment _Z_OF ((unzFile file, char *szComment, uLong uSizeBuf))

13.161.4.9 int ZEXPORT unzLocateFile _Z_OF ((unzFile file, const char *szFileName, int iCaseSensitivity))

13.161.4.10 int ZEXPORT unzGetCurrentFileInfo64 _Z_OF ((unzFile file, unz_file_info64 *pfile_info, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize))

13.161.4.11 int ZEXPORT unzGetCurrentFileInfo _Z_OF ((unzFile file, unz_file_info ∗pfile_info, char ∗szFileName, uLong fileNameBufferSize, void ∗extraField, uLong extraFieldBufferSize, char ∗szComment, uLong commentBufferSize))

13.161.4.12 int ZEXPORT unzOpenCurrentFilePassword _Z_OF ((unzFile file, const char ∗password))

13.161.4.13 int ZEXPORT unzOpenCurrentFile2 _Z_OF ((unzFile file, int ∗method, int ∗level, int raw))

13.161.4.14 int ZEXPORT unzOpenCurrentFile3 _Z_OF ((unzFile file, int ∗method, int ∗level, int raw, const char ∗password))

13.161.4.15 int ZEXPORT unzReadCurrentFile _Z_OF ((unzFile file, voidp buf, unsigned len))

13.161.4.16 int ZEXPORT unzClearFlags (unzFile *file,* unsigned *flags*)

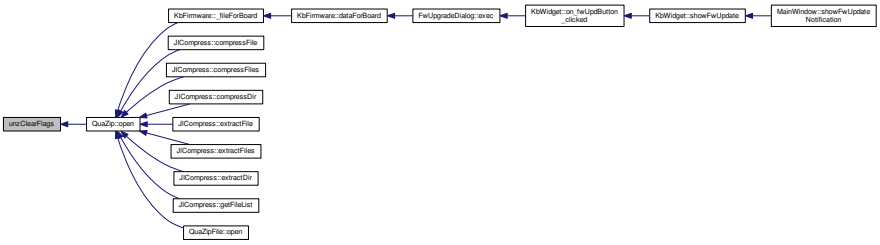
Definition at line 2147 of file unzip.c.

References unz64_s::flags, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZip::open().

```
2148 {
2149     unz64_s* s;
2150     if (file == NULL)
2151         return UNZ_PARAMERROR;
2152     s = (unz64_s*)file;
2153     s->flags &= ~flags;
2154     return UNZ_OK;
2155 }
```

Here is the caller graph for this function:



13.161.4.17 int ZEXPORT unzGetFilePos (unzFile *file,* unz_file_pos ∗ *file_pos*)

Definition at line 1334 of file unzip.c.

References unz_file_pos_s::num_of_file, unz64_file_pos_s::num_of_file, unz_file_pos_s::pos_in_zip_directory, unz64_file_pos_s::pos_in_zip_directory, UNZ_OK, and unzGetFilePos64().

```
1337 {
1338     unz64_file_pos file_pos64;
1339     int err = unzGetFilePos64(file,&file_pos64);
1340     if (err==UNZ_OK)
1341     {
1342         file_pos->pos_in_zip_directory = (uLong)file_pos64.
      pos_in_zip_directory;
1343         file_pos->num_of_file = (uLong)file_pos64.num_of_file;
1344     }
1345     return err;
1346 }
```

Here is the call graph for this function:

13.161.4.18 int ZEXPORT unzGetFilePos64 (unzFile *file*, unz64_file_pos * *file_pos*)

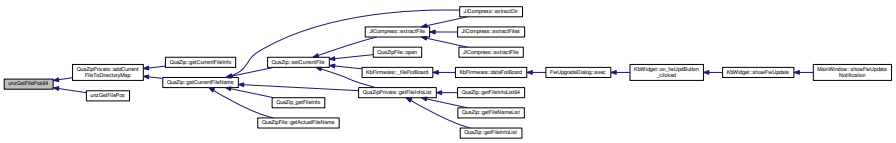
Definition at line 1318 of file unzip.c.

References unz64_s::current_file_ok, unz64_s::num_file, unz64_file_pos_s::num_of_file, unz64_s::pos_in_central_dir, unz64_file_pos_s::pos_in_zip_directory, UNZ_END_OF_LIST_OF_FILE, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZipPrivate::addCurrentFileToDirectoryMap(), and unzGetFilePos().

```
1319 {
1320     unz64_s* s;
1321
1322     if (file==NULL || file_pos==NULL)
1323         return UNZ_PARAMERROR;
1324     s=(unz64_s*)file;
1325     if (!s->current_file_ok)
1326         return UNZ_END_OF_LIST_OF_FILE;
1327
1328     file_pos->pos_in_zip_directory  = s->pos_in_central_dir;
1329     file_pos->num_of_file           = s->num_file;
1330
1331     return UNZ_OK;
1332 }
```

Here is the caller graph for this function:



13.161.4.19 uLong ZEXPORT unzGetOffset (unzFile *file*)

Definition at line 2102 of file unzip.c.

References unzGetOffset64().

```
2103 {
2104     ZPOS64_T offset64;
2105
2106     if (file==NULL)
2107           return 0; /*UNZ_PARAMERROR; */
2108     offset64 = unzGetOffset64(file);
2109     return (uLong)offset64;
2110 }
```

Here is the call graph for this function:

13.161.4.20 ZPOS64_T ZEXPORT unzGetOffset64 (unzFile *file*)

Definition at line 2087 of file unzip.c.

References unz64_s::current_file_ok, unz64_s::gi, unz64_s::num_file, unz_global_info64_s::number_entry, and unz64_s::pos_in_central_dir.

Referenced by unzGetOffset().

```
2088 {
2089     unz64_s* s;
2090
2091     if (file==NULL)
2092           return 0; /*UNZ_PARAMERROR; */
2093     s=(unz64_s*)file;
2094     if (!s->current_file_ok)
2095       return 0;
2096     if (s->gi.number_entry != 0 && s->gi.number_entry != 0xffff)
2097       if (s->num_file==s->gi.number_entry)
2098          return 0;
2099     return s->pos_in_central_dir;
2100 }
```

Here is the caller graph for this function:

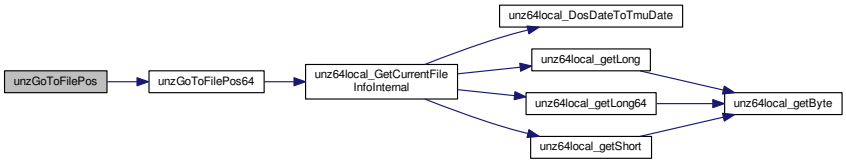
13.161.4.21 int ZEXPORT unzGoToFilePos (unzFile *file,* unz_file_pos * *file_pos*)

Definition at line 1370 of file unzip.c.

References unz_file_pos_s::num_of_file, unz64_file_pos_s::num_of_file, unz_file_pos_s::pos_in_zip_directory, unz64_file_pos_s::pos_in_zip_directory, UNZ_PARAMERROR, and unzGoToFilePos64().

```
1373 {
1374     unz64_file_pos file_pos64;
1375     if (file_pos == NULL)
1376         return UNZ_PARAMERROR;
1377
1378     file_pos64.pos_in_zip_directory = file_pos->
     pos_in_zip_directory;
1379     file_pos64.num_of_file = file_pos->num_of_file;
1380     return unzGoToFilePos64(file,&file_pos64);
1381 }
```

Here is the call graph for this function:



13.161.4.22 int ZEXPORT unzGoToFilePos64 (unzFile *file*, const unz64_file_pos * *file_pos*)

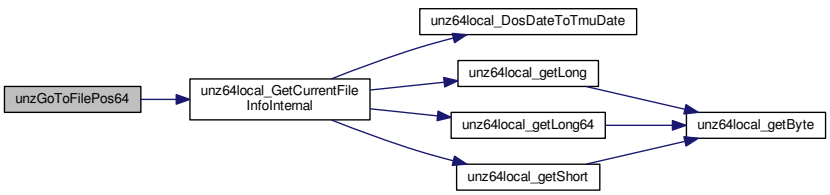
Definition at line 1348 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::num_file, unz64_file_pos_s::num_of_file, unz64_s::pos_in_central_dir, unz64_file_pos_s::pos_in_zip_directory, unz64local_GetCurrentFileInfoInternal(), UNZ_OK, and UNZ_PARAMERROR.

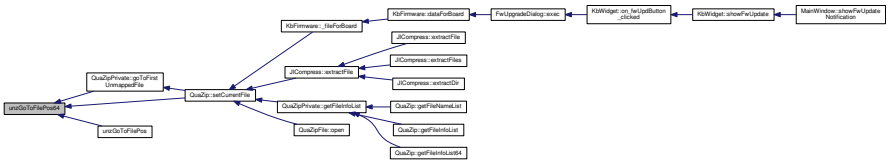
Referenced by QuaZipPrivate::goToFirstUnmappedFile(), QuaZip::setCurrentFile(), and unzGoToFilePos().

```
1349 {
1350     unz64_s* s;
1351     int err;
1352
1353     if (file==NULL || file_pos==NULL)
1354         return UNZ_PARAMERROR;
1355     s=(unz64_s*)file;
1356
1357     /* jump to the right spot */
1358     s->pos_in_central_dir = file_pos->pos_in_zip_directory;
1359     s->num_file           = file_pos->num_of_file;
1360
1361     /* set the current file */
1362     err = unz64local_GetCurrentFileInfoInternal(file,&s->
      cur_file_info,
1363                                                 &s->cur_file_info_internal,
1364                                                 NULL,0,NULL,0,NULL,0);
1365     /* return results */
1366     s->current_file_ok = (err == UNZ_OK);
1367     return err;
1368 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.161.4.23 unzFile unzOpenInternal (voidpf *file*, zlib_filefunc64_32_def * *pzlib_filefunc64_32_def*, int *is64bitOpenFunction*, unsigned *flags*)

Definition at line 587 of file unzip.c.

References ALLOC, unz64_s::byte_before_the_zipfile, unz64_s::central_pos, unz64_s::encrypted, unz64_s::filestream, fill_qiodevice64_filefunc(), unz64_s::flags, unz64_s::gi, unz64_s::is64bitOpenFunction, unz64_s::isZip64, unz_global_info64_s::number_entry, unz64_s::offset_central_dir, unz64_s::pfile_in_zip_read, unz64_s::size_central_dir, unz_global_info64_s::size_comment, unz64local_getLong(), unz64local_getLong64(), unz64local_getShort(), unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), UNZ_AUTO_CLOSE, UNZ_BADZIPFILE, unz_copyright, UNZ_ERRNO, UNZ_OK, unzGoToFirstFile(), unz64_s::z_filefunc, ZCLOSE64, ZFAKECLOSE64, zlib_filefunc64_32_def_s::zfile_func64, ZLIB_FILEFUNC_MODE_EXISTING, ZLIB_FILEFUNC_MODE_READ, ZLIB_FILEFUNC_SEEK_SET, ZOPEN64, zlib_filefunc64_32_def_s::zseek32_file, ZSEEK64, and zlib_filefunc64_32_def_s::ztell32_file.

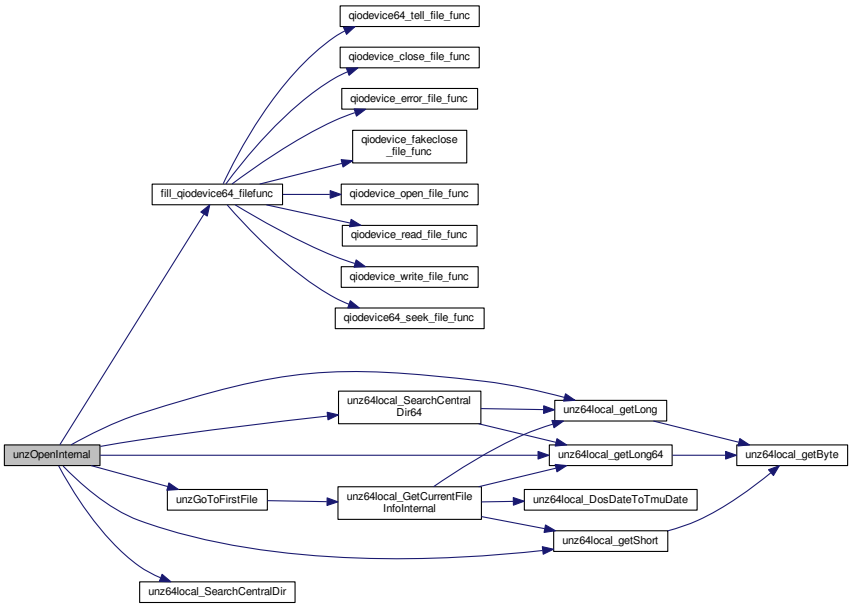
Referenced by QuaZip::open(), unzOpen(), unzOpen2(), unzOpen2_64(), and unzOpen64().

```
590 {
591     unz64_s us;
592     unz64_s *s;
593     ZPOS64_T central_pos;
594     uLong   uL;
595 
596     uLong number_disk;          /* number of the current dist, used for
597                                    spaning ZIP, unsupported, always 0*/
598     uLong number_disk_with_CD;  /* number the the disk with central dir, used
599                                    for spaning ZIP, unsupported, always 0*/
600     ZPOS64_T number_entry_CD;       /* total number of entries in
601                                    the central dir
602                                    (same than number_entry on nospan) */
603 
604     int err=UNZ_OK;
605 
606     if (unz_copyright[0]!=' ')
607         return NULL;
608 
609     us.flags = flags;
610     us.z_filefunc.zseek32_file = NULL;
611     us.z_filefunc.ztell32_file = NULL;
612     if (pzlib_filefunc64_32_def==NULL)
613         fill_qiodevice64_filefunc(&us.z_filefunc.
    zfile_func64);
614     else
615         us.z_filefunc = *pzlib_filefunc64_32_def;
616     us.is64bitOpenFunction = is64bitOpenFunction;
617 
618 
619 
620     us.filestream = ZOPEN64(us.z_filefunc,
621                                                 file,
622                                                 ZLIB_FILEFUNC_MODE_READ |
623                                                 ZLIB_FILEFUNC_MODE_EXISTING);
624     if (us.filestream==NULL)
625         return NULL;
626 
627     central_pos = unz64local_SearchCentralDir64(&us.
    z_filefunc,us.filestream);
628     if (central_pos)
629     {
630         uLong uS;
631         ZPOS64_T uL64;
632 
```

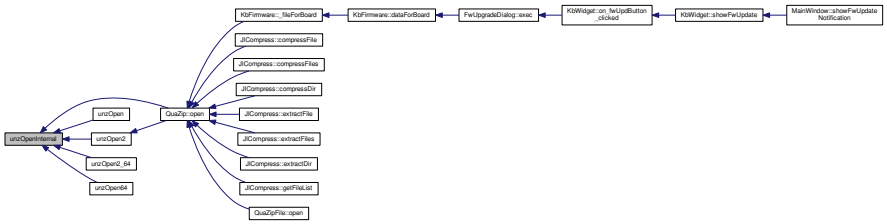
```
633         us.isZip64 = 1;
634
635         if (ZSEEK64(us.z_filefunc, us.filestream,
636                                         central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
637         err=UNZ_ERRNO;
638
639         /* the signature, already checked */
640         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
641             err=UNZ_ERRNO;
642
643         /* size of zip64 end of central directory record */
644         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&uL64)!=UNZ_OK)
645             err=UNZ_ERRNO;
646
647         /* version made by */
648         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uS)!=UNZ_OK)
649             err=UNZ_ERRNO;
650
651         /* version needed to extract */
652         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uS)!=UNZ_OK)
653             err=UNZ_ERRNO;
654
655         /* number of this disk */
656         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&number_disk)!=UNZ_OK)
657             err=UNZ_ERRNO;
658
659         /* number of the disk with the start of the central directory */
660         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&number_disk_with_CD)!=UNZ_OK)
661             err=UNZ_ERRNO;
662
663         /* total number of entries in the central directory on this disk */
664         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&us.gi.number_entry)!=UNZ_OK)
665             err=UNZ_ERRNO;
666
667         /* total number of entries in the central directory */
668         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&number_entry_CD)!=UNZ_OK)
669             err=UNZ_ERRNO;
670
671         if ((number_entry_CD!=us.gi.number_entry) ||
672             (number_disk_with_CD!=0) ||
673             (number_disk!=0))
674             err=UNZ_BADZIPFILE;
675
676         /* size of the central directory */
677         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&us.size_central_dir)!=UNZ_OK)
678             err=UNZ_ERRNO;
679
680         /* offset of start of central directory with respect to the
681           starting disk number */
682         if (unz64local_getLong64(&us.z_filefunc, us.
    filestream,&us.offset_central_dir)!=UNZ_OK)
683             err=UNZ_ERRNO;
684
685         us.gi.size_comment = 0;
686       }
687       else
688       {
689         central_pos = unz64local_SearchCentralDir(&us.
    z_filefunc,us.filestream);
690         if (central_pos==0)
691             err=UNZ_ERRNO;
692
693         us.isZip64 = 0;
694
695         if (ZSEEK64(us.z_filefunc, us.filestream,
696                                         central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
697             err=UNZ_ERRNO;
698
699         /* the signature, already checked */
700         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
701             err=UNZ_ERRNO;
702
703         /* number of this disk */
704         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&number_disk)!=UNZ_OK)
705             err=UNZ_ERRNO;
706
```

```
707         /* number of the disk with the start of the central directory */
708         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&number_disk_with_CD)!=UNZ_OK)
709             err=UNZ_ERRNO;
710
711         /* total number of entries in the central dir on this disk */
712         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
713             err=UNZ_ERRNO;
714         us.gi.number_entry = uL;
715
716         /* total number of entries in the central dir */
717         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
718             err=UNZ_ERRNO;
719         number_entry_CD = uL;
720
721         if ((number_entry_CD!=us.gi.number_entry) ||
722             (number_disk_with_CD!=0) ||
723             (number_disk!=0))
724             err=UNZ_BADZIPFILE;
725
726         /* size of the central directory */
727         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
728             err=UNZ_ERRNO;
729         us.size_central_dir = uL;
730
731         /* offset of start of central directory with respect to the
732             starting disk number */
733         if (unz64local_getLong(&us.z_filefunc, us.
    filestream,&uL)!=UNZ_OK)
734             err=UNZ_ERRNO;
735         us.offset_central_dir = uL;
736
737         /* zipfile comment length */
738         if (unz64local_getShort(&us.z_filefunc, us.
    filestream,&us.gi.size_comment)!=UNZ_OK)
739             err=UNZ_ERRNO;
740     }
741
742     if ((central_pos<us.offset_central_dir+us.size_central_dir) &&
743         (err==UNZ_OK))
744         err=UNZ_BADZIPFILE;
745
746     if (err!=UNZ_OK)
747     {
748         if ((us.flags & UNZ_AUTO_CLOSE) != 0)
749             ZCLOSE64(us.z_filefunc, us.filestream);
750         else
751             ZFAKECLOSE64(us.z_filefunc, us.filestream);
752         return NULL;
753     }
754
755     us.byte_before_the_zipfile = central_pos -
756                             (us.offset_central_dir+us.
    size_central_dir);
757     us.central_pos = central_pos;
758     us.pfile_in_zip_read = NULL;
759     us.encrypted = 0;
760
761
762     s=(unz64_s*)ALLOC(sizeof(unz64_s));
763     if( s != NULL)
764     {
765         *s=us;
766         unzGoToFirstFile((unzFile)s);
767     }
768     return (unzFile)s;
769 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.161.4.24 int ZEXPORT unzSetFlags (unzFile *file*, unsigned *flags*)

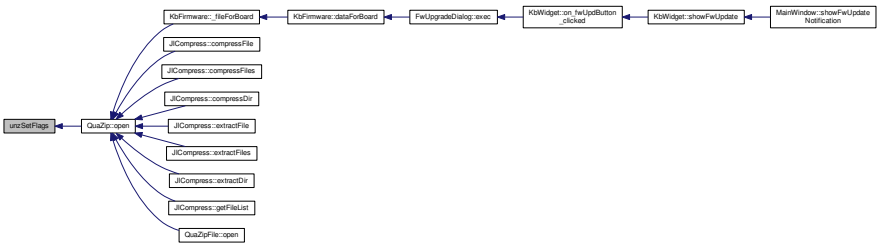
Definition at line 2136 of file unzip.c.

References unz64_s::flags, UNZ_OK, and UNZ_PARAMERROR.

Referenced by QuaZip::open().

```
2137 {
2138     unz64_s* s;
2139     if (file == NULL)
2140         return UNZ_PARAMERROR;
2141     s = (unz64_s*)file;
2142     s->flags |= flags;
2143     return UNZ_OK;
2144 }
```

Here is the caller graph for this function:



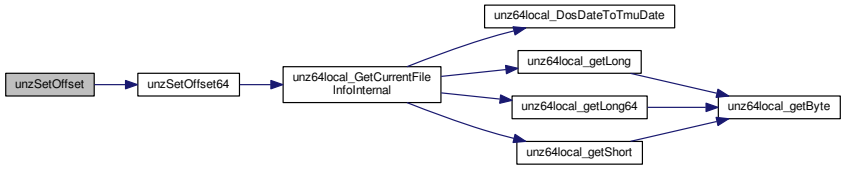
13.161.4.25 int ZEXPORT unzSetOffset (unzFile *file*, uLong *pos*)

Definition at line 2130 of file unzip.c.

References unzSetOffset64().

```
2131 {
2132     return unzSetOffset64(file,pos);
2133 }
```

Here is the call graph for this function:



13.161.4.26 int ZEXPORT unzSetOffset64 (unzFile *file*, ZPOS64_T *pos*)

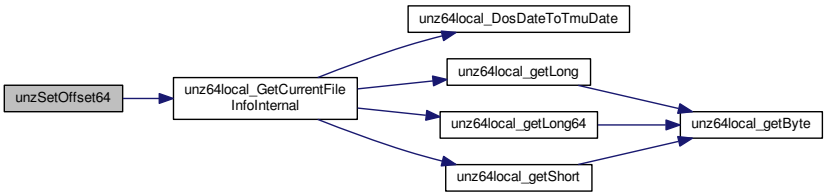
Definition at line 2112 of file unzip.c.

References unz64_s::cur_file_info, unz64_s::cur_file_info_internal, unz64_s::current_file_ok, unz64_s::gi, unz64_s::num_file, unz_global_info64_s::number_entry, unz64_s::pos_in_central_dir, unz64local_GetCurrentFileInfoInternal(), UNZ_OK, and UNZ_PARAMERROR.

Referenced by unzSetOffset().

```
2113 {
2114     unz64_s* s;
2115     int err;
2116
2117     if (file==NULL)
2118         return UNZ_PARAMERROR;
2119     s=(unz64_s*)file;
2120
2121     s->pos_in_central_dir = pos;
2122     s->num_file = s->gi.number_entry;       /* hack */
2123     err = unz64local_GetCurrentFileInfoInternal(file,&s->
      cur_file_info,
2124                                                 &s->cur_file_info_internal,
2125                                                 NULL,0,NULL,0,NULL,0);
2126     s->current_file_ok = (err == UNZ_OK);
2127     return err;
2128 }
```

Here is the call graph for this function:

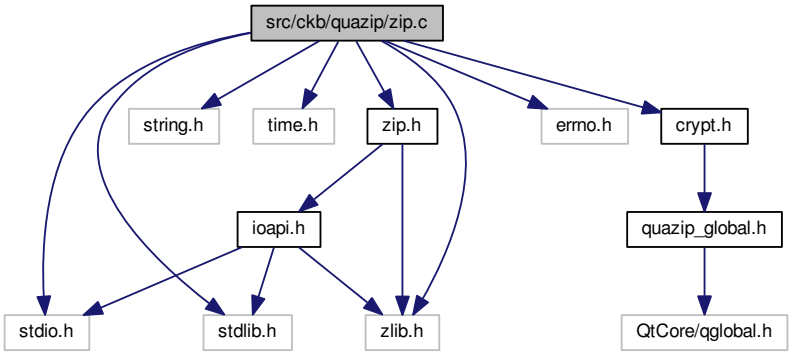


Here is the caller graph for this function:

13.162 src/ckb/quazip/zip.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "zlib.h"
#include "zip.h"
#include <errno.h>
#include "crypt.h"
```

Include dependency graph for zip.c:



Data Structures

- struct linkedlist_datablock_internal_s
- struct linkedlist_data_s
- struct curfile64_info
- struct zip64_internal

Macros

- #define local static
- #define VERSIONMADEBY (0x031e) /∗ best for standard pkware crypt ∗/
- #define Z_BUFSIZE (64∗1024) /∗ (16384) ∗/
- #define Z_MAXFILENAMEINZIP (256)
- #define ALLOC(size) (malloc(size))
- #define TRYFREE(p) {if (p) free(p);}
- #define MAKEULONG64(a, b) ((ZPOS64_T)(((unsigned long)(a)) | ((ZPOS64_T)((unsigned long)(b))) << 32))
- #define SEEK_CUR 1
- #define SEEK_END 2
- #define SEEK_SET 0
- #define SIZEDATA_INDATABLOCK (4096-(4∗4))
- #define LOCALHEADERMAGIC (0x04034b50)
- #define DESCRIPTORHEADERMAGIC (0x08074b50)
- #define CENTRALHEADERMAGIC (0x02014b50)
- #define ENDHEADERMAGIC (0x06054b50)
- #define ZIP64ENDHEADERMAGIC (0x6064b50)
- #define ZIP64ENDLOCHEADERMAGIC (0x7064b50)
- #define FLAG_LOCALHEADER_OFFSET (0x06)
- #define CRC_LOCALHEADER_OFFSET (0x0e)
- #define SIZECENTRALHEADER (0x2e) /∗ 46 ∗/
- #define INCLUDECRYPTINGCODE_IFCRYPTALLOWED
- #define BUFREADCOMMENT (0x400)

Typedefs

- typedef uLongf z_crc_t
- typedef struct linkedlist_datablock_internal_s linkedlist_datablock_internal
- typedef struct linkedlist_data_s linkedlist_data

Functions

- static linkedlist_datablock_internal ∗ allocate_new_datablock ()
- static void free_datablock (linkedlist_datablock_internal ∗ldi)
- static void init_linkedlist (linkedlist_data ∗ll)
- static void free_linkedlist (linkedlist_data ∗ll)
- static int add_data_in_datablock (linkedlist_data ∗ll, const void ∗buf, uLong len)
- static int zip64local_putValue _Z_OF ((const zlib_filefunc64_32_def ∗pzlib_filefunc_def, voidpf filestream, ZPOS64_T x, int nbByte))
- static int zip64local_putValue (const zlib_filefunc64_32_def ∗pzlib_filefunc_def, voidpf filestream, ZPOS64_T x, int nbByte)
- static void zip64local_putValue_inmemory _Z_OF ((void ∗dest, ZPOS64_T x, int nbByte))

- static void zip64local_putValue_inmemory (void *dest, ZPOS64_T x, int nbByte)
- static uLong zip64local_TmzDateToDosDate (const tm_zip *ptm)
- static int zip64local_getByte _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi))
- static int zip64local_getByte (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi)
- static int zip64local_getShort _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX))
- static int zip64local_getShort (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)
- static int zip64local_getLong (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)
- static int zip64local_getLong64 _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX))
- static int zip64local_getLong64 (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX)
- static ZPOS64_T zip64local_SearchCentralDir _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream))
- static ZPOS64_T zip64local_SearchCentralDir (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)
- static ZPOS64_T zip64local_SearchCentralDir64 (const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)
- int LoadCentralDirectoryRecord (zip64_internal *pziinit)
- zipFile ZEXPORT zipOpen3 (voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc64_32_def *pzlib_filefunc64_32_def, unsigned flags)
- zipFile ZEXPORT zipOpen2 (voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc_def *pzlib_filefunc32_def)
- zipFile ZEXPORT zipOpen2_64 (voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc64_def *pzlib_filefunc_def)
- zipFile ZEXPORT zipOpen (voidpf file, int append)
- zipFile ZEXPORT zipOpen64 (voidpf file, int append)
- int Write_LocalFileHeader (zip64_internal *zi, const char *filename, uInt size_extrafield_local, const void *extrafield_local, uLong version_to_extract)
- int ZEXPORT zipOpenNewFileInZip4_64 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char *password, uLong crcForCrypting, uLong versionMadeBy, uLong flagBase, int zip64)
- int ZEXPORT zipOpenNewFileInZip4 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char *password, uLong crcForCrypting, uLong versionMadeBy, uLong flagBase)
- int ZEXPORT zipOpenNewFileInZip3 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char *password, uLong crcForCrypting)
- int ZEXPORT zipOpenNewFileInZip3_64 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char *password, uLong crcForCrypting, int zip64)
- int ZEXPORT zipOpenNewFileInZip2 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int raw)
- int ZEXPORT zipOpenNewFileInZip2_64 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int raw, int zip64)
- int ZEXPORT zipOpenNewFileInZip64 (zipFile file, const char *filename, const zip_fileinfo *zipfi, const void *extrafield_local, uInt size_extrafield_local, const void *extrafield_global, uInt size_extrafield_global, const char *comment, int method, int level, int zip64)

- int ZEXPORT zipOpenNewFileInZip (zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level)
- static int zip64FlushWriteBuffer (zip64_internal ∗zi)
- int ZEXPORT zipWriteInFileInZip (zipFile file, const void ∗buf, unsigned int len)
- int ZEXPORT zipCloseFileInZipRaw (zipFile file, uLong uncompressed_size, uLong crc32)
- int ZEXPORT zipCloseFileInZipRaw64 (zipFile file, ZPOS64_T uncompressed_size, uLong crc32)
- int ZEXPORT zipCloseFileInZip (zipFile file)
- int Write_Zip64EndOfCentralDirectoryLocator (zip64_internal ∗zi, ZPOS64_T zip64eocd_pos_inzip)
- int Write_Zip64EndOfCentralDirectoryRecord (zip64_internal ∗zi, uLong size_centraldir, ZPOS64_T centraldir_pos_inzip)
- int Write_EndOfCentralDirectoryRecord (zip64_internal ∗zi, uLong size_centraldir, ZPOS64_T centraldir_pos_inzip)
- int Write_GlobalComment (zip64_internal ∗zi, const char ∗global_comment)
- int ZEXPORT zipClose (zipFile file, const char ∗global_comment)
- int ZEXPORT zipRemoveExtraInfoBlock (char ∗pData, int ∗dataLen, short sHeader)
- int ZEXPORT zipSetFlags (zipFile file, unsigned flags)
- int ZEXPORT zipClearFlags (zipFile file, unsigned flags)

Variables

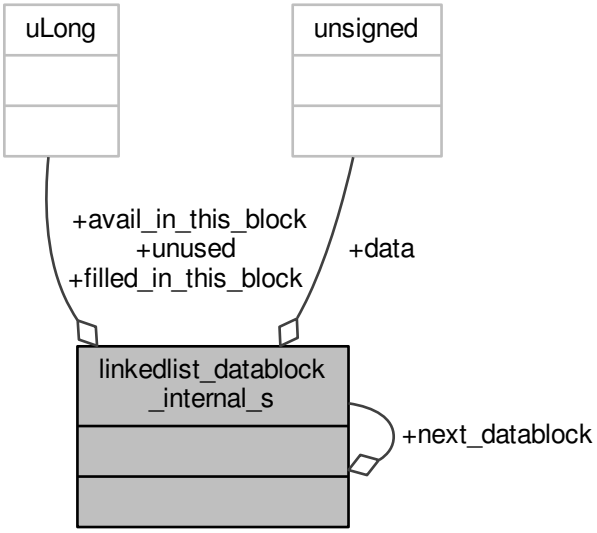
- const char zip_copyright [] =" zip 1.01 Copyright 1998-2004 Gilles Vollant - http://www.winimage.com/zLibDll"

13.162.1 Data Structure Documentation

13.162.1.1 struct linkedlist_datablock_internal_s

Definition at line 121 of file zip.c.

Collaboration diagram for linkedlist_datablock_internal_s:



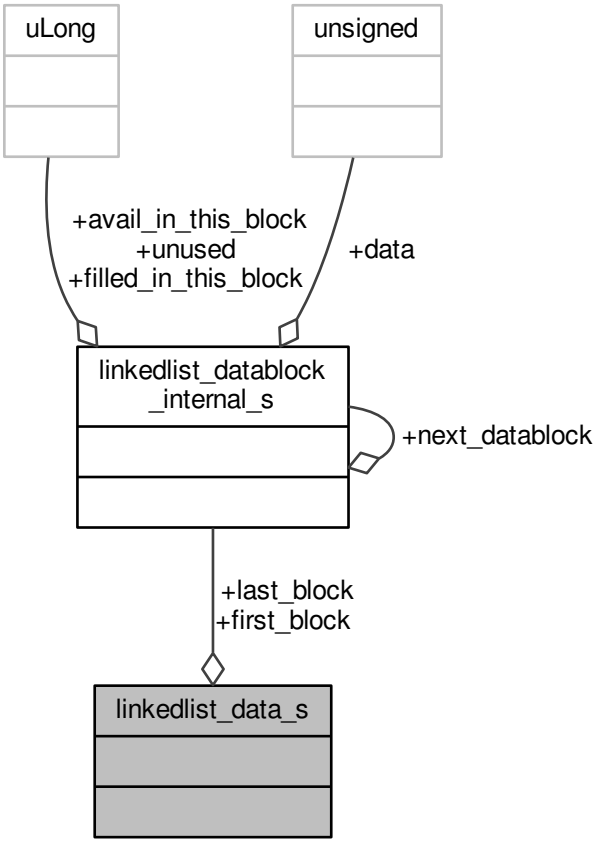
Data Fields

uLong	avail_in_this_-block	
unsigned char	data[(4096-(4 ∗4))]	
uLong	filled_in_this_-block	
struct linkedlist_-datablock_-internal_s ∗	next_datablock	
uLong	unused	

13.162.1.2 struct linkedlist_data_s

Definition at line 130 of file zip.c.

Collaboration diagram for linkedlist_data_s:



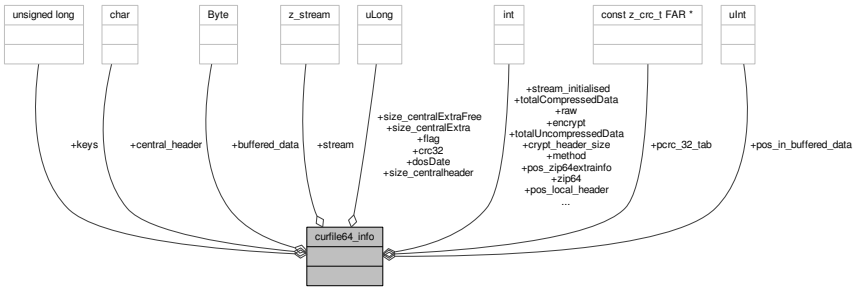
Data Fields

linkedlist_-datablock_-internal *	first_block	
linkedlist_-datablock_-internal *	last_block	

13.162.1.3 struct curfile64_info

Definition at line 137 of file zip.c.

Collaboration diagram for curfile64_info:



Data Fields

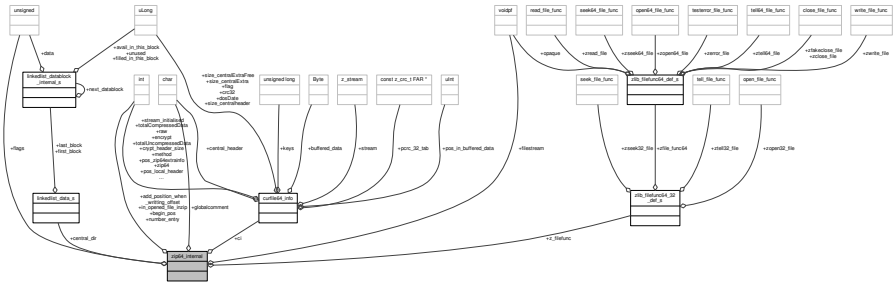
Byte	buffered_-data[(64 ∗1024)]	
char ∗	central_header	
uLong	crc32	
int	crypt_header_-size	
uLong	dosDate	
int	encrypt	
uLong	flag	
unsigned long	keys[3]	
int	method	
const z_crc_t FAR ∗	pcrc_32_tab	
uInt	pos_in_buffered-_data	
ZPOS64_T	pos_local_-header	
ZPOS64_T	pos_-zip64extrainfo	

int	raw	
uLong	size_central-Extra	
uLong	size_central-ExtraFree	
uLong	size_-centralheader	
z_stream	stream	
int	stream_-initialised	
ZPOS64_T	total-Compressed-Data	
ZPOS64_T	total-Uncompressed-Data	
int	zip64	

13.162.1.4 struct zip64_internal

Definition at line 172 of file zip.c.

Collaboration diagram for zip64_internal:



Data Fields

ZPOS64_T	add_position_-when_writting_-offset	
ZPOS64_T	begin_pos	
linkedlist_data	central_dir	
curfile64_info	ci	
voidpf	filestream	
unsigned	flags	
char *	globalcomment	
int	in_opened_file_-inzip	
ZPOS64_T	number_entry	
zlib_filefunc64_-32_def	z_filefunc	

13.162.2 Macro Definition Documentation

13.162.2.1 #define ALLOC(*size*) (malloc(size))

Definition at line 68 of file zip.c.

Referenced by allocate_new_datablock(), LoadCentralDirectoryRecord(), zip64local_SearchCentralDir(), zip64local_SearchCentralDir64(), zipOpen3(), zipOpenNewFileInZip4_64(), and zipRemoveExtraInfoBlock().

13.162.2.2 #define BUFREADCOMMENT (0x400)

Definition at line 481 of file zip.c.

Referenced by zip64local_SearchCentralDir(), and zip64local_SearchCentralDir64().

13.162.2.3 #define CENTRALHEADERMAGIC (0x02014b50)

Definition at line 111 of file zip.c.

Referenced by zipOpenNewFileInZip4_64().

13.162.2.4 #define CRC_LOCALHEADER_OFFSET (0x0e)

Definition at line 117 of file zip.c.

13.162.2.5 #define DESCRIPTORHEADERMAGIC (0x08074b50)

Definition at line 110 of file zip.c.

Referenced by zipCloseFileInZipRaw64().

13.162.2.6 #define ENDHEADERMAGIC (0x06054b50)

Definition at line 112 of file zip.c.

Referenced by Write_EndOfCentralDirectoryRecord().

13.162.2.7 #define FLAG_LOCALHEADER_OFFSET (0x06)

Definition at line 116 of file zip.c.

13.162.2.8 #define INCLUDECRYPTINGCODE_IFCRYPTALLOWED

Definition at line 194 of file zip.c.

13.162.2.9 #define local static

Definition at line 51 of file zip.c.

13.162.2.10 #define LOCALHEADERMAGIC (0x04034b50)

Definition at line 109 of file zip.c.

Referenced by Write_LocalFileHeader().

13.162.2.11 #define MAKEULONG64(a, b) ((ZPOS64_T)(((unsigned long)(a)) | ((ZPOS64_T)((unsigned long)(b))) << 32))

Definition at line 83 of file zip.c.

13.162.2.12 #define SEEK_CUR 1

Definition at line 86 of file zip.c.

13.162.2.13 #define SEEK_END 2

Definition at line 90 of file zip.c.

Referenced by zipOpen3().

13.162.2.14 #define SEEK_SET 0

Definition at line 94 of file zip.c.

13.162.2.15 #define SIZECENTRALHEADER (0x2e) /* 46 */

Definition at line 119 of file zip.c.

Referenced by zipOpenNewFileInZip4_64().

13.162.2.16 #define SIZEDATA_INDATABLOCK (4096-(4*4))

Definition at line 107 of file zip.c.

Referenced by allocate_new_datablock(), and LoadCentralDirectoryRecord().

13.162.2.17 #define TRYFREE(p) {if (p) free(p);}

Definition at line 71 of file zip.c.

Referenced by free_datablock(), LoadCentralDirectoryRecord(), zip64local_SearchCentralDir(), zip64local_SearchCentralDir64(), zipClose(), zipOpen3(), and zipRemoveExtraInfoBlock().

13.162.2.18 #define VERSIONMADEBY (0x031e) /* best for standard pkware crypt */

Definition at line 56 of file zip.c.

Referenced by zipOpenNewFileInZip(), zipOpenNewFileInZip2(), zipOpenNewFileInZip2_64(), zipOpenNewFileInZip3(), zipOpenNewFileInZip3_64(), and zipOpenNewFileInZip64().

13.162.2.19 #define Z_BUFSIZE (64*1024) /* (16384) */

Definition at line 60 of file zip.c.

Referenced by zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), and zipWriteInFileInZip().

13.162.2.20 #define Z_MAXFILENAMEINZIP (256)

Definition at line 64 of file zip.c.

13.162.2.21 #define ZIP64ENDHEADERMAGIC (0x6064b50)

Definition at line 113 of file zip.c.

Referenced by Write_Zip64EndOfCentralDirectoryRecord().

13.162.2.22 #define ZIP64ENDLOCHEADERMAGIC (0x7064b50)

Definition at line 114 of file zip.c.

Referenced by Write_Zip64EndOfCentralDirectoryLocator().

13.162.3 Typedef Documentation

13.162.3.1 typedef struct linkedlist_data_s linkedlist_data

13.162.3.2 typedef struct linkedlist_datablock_internal_s linkedlist_datablock_internal

13.162.3.3 typedef uLongf z_crc_t

Definition at line 34 of file zip.c.

13.162.4 Function Documentation

13.162.4.1 static int zip64local_putValue _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T x, int nbByte)) [static]

13.162.4.2 static void zip64local_putValue_inmemory _Z_OF ((void *dest, ZPOS64_T x, int nbByte)) [static]

13.162.4.3 static int zip64local_getByte _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi)) [static]

13.162.4.4 static int zip64local_getShort _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)) [static]

13.162.4.5 static int zip64local_getLong64 _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX)) [static]

13.162.4.6 static ZPOS64_T zip64local_SearchCentralDir _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)) [static]

13.162.4.7 static int add_data_in_datablock (linkedlist_data * *ll*, const void * *buf*, uLong *len*) [static]

Definition at line 234 of file zip.c.

References allocate_new_datablock(), linkedlist_datablock_internal_s::avail_in_this_block, linkedlist_datablock_internal_s::data, linkedlist_datablock_internal_s::filled_in_this_block, linkedlist_data_s::first_block, linkedlist_data_s::last_block, linkedlist_datablock_internal_s::next_datablock, ZIP_INTERNALERROR, and ZIP_OK.

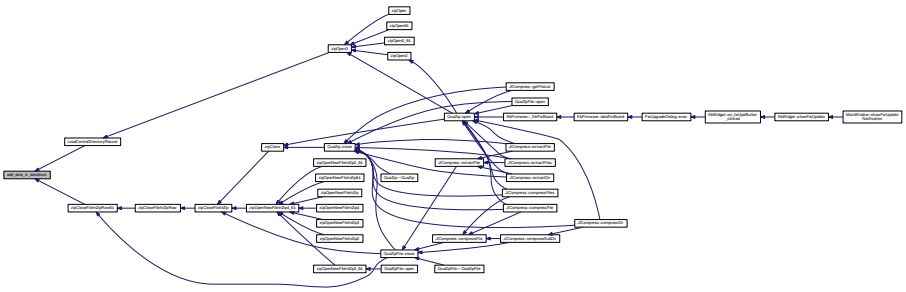
Referenced by LoadCentralDirectoryRecord(), and zipCloseFileInZipRaw64().

```
235 {
236     linkedlist_datablock_internal* ldi;
237     const unsigned char* from_copy;
238
239     if (ll==NULL)
240         return ZIP_INTERNALERROR;
241
242     if (ll->last_block == NULL)
243     {
```

```
244         ll->first_block = ll->last_block =
    allocate_new_datablock();
245         if (ll->first_block == NULL)
246             return ZIP_INTERNALERROR;
247     }
248
249     ldi = ll->last_block;
250     from_copy = (unsigned char*)buf;
251
252     while (len>0)
253     {
254         uInt copy_this;
255         uInt i;
256         unsigned char* to_copy;
257
258         if (ldi->avail_in_this_block==0)
259         {
260             ldi->next_datablock = allocate_new_datablock();
261             if (ldi->next_datablock == NULL)
262                 return ZIP_INTERNALERROR;
263             ldi = ldi->next_datablock ;
264             ll->last_block = ldi;
265         }
266
267         if (ldi->avail_in_this_block < len)
268             copy_this = (uInt)ldi->avail_in_this_block;
269         else
270             copy_this = (uInt)len;
271
272         to_copy = &(ldi->data[ldi->filled_in_this_block]);
273
274         for (i=0;i<copy_this;i++)
275             *(to_copy+i)=*(from_copy+i);
276
277         ldi->filled_in_this_block += copy_this;
278         ldi->avail_in_this_block -= copy_this;
279         from_copy += copy_this ;
280         len -= copy_this;
281     }
282     return ZIP_OK;
283 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.162.4.8 static linkedlist_datablock_internal* allocate_new_datablock ( ) [static]

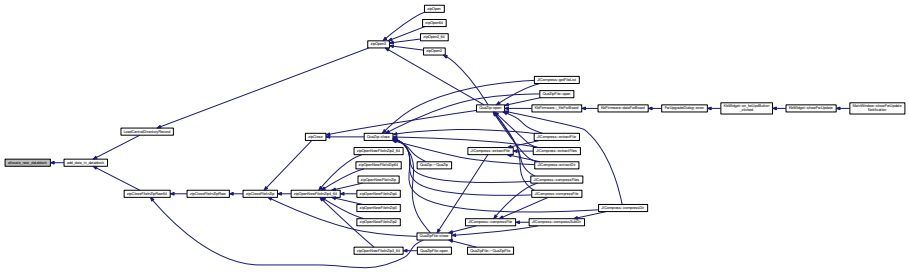
Definition at line 198 of file zip.c.

References ALLOC, linkedlist_datablock_internal_s::avail_in_this_block, linkedlist_datablock_internal_s::filled_in_this_block, linkedlist_datablock_internal_s::next_datablock, and SIZEDATA_INDATABLOCK.

Referenced by add_data_in_datablock().

```
199 {
200     linkedlist_datablock_internal* ldi;
201     ldi = (linkedlist_datablock_internal*)
202                 ALLOC(sizeof(linkedlist_datablock_internal));
203     if (ldi!=NULL)
204     {
205         ldi->next_datablock = NULL ;
206         ldi->filled_in_this_block = 0 ;
207         ldi->avail_in_this_block = SIZEDATA_INDATABLOCK ;
208     }
209     return ldi;
210 }
```

Here is the caller graph for this function:



13.162.4.9 static void free_datablock (linkedlist_datablock_internal * ldi) [static]

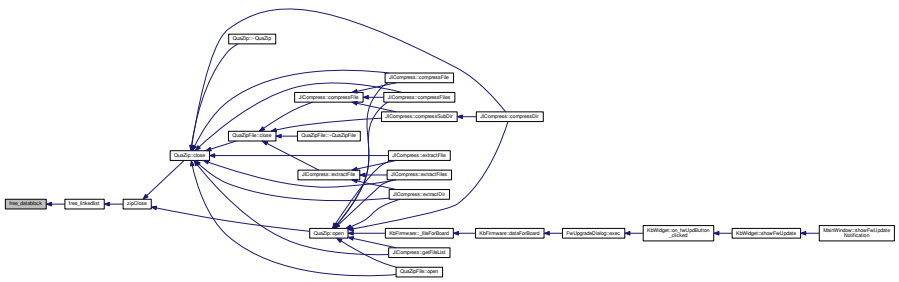
Definition at line 212 of file zip.c.

References linkedlist_datablock_internal_s::next_datablock, and TRYFREE.

Referenced by free_linkedlist().

```
213 {
214     while (ldi!=NULL)
215     {
216         linkedlist_datablock_internal* ldinext = ldi->
    next_datablock;
217         TRYFREE(ldi);
218         ldi = ldinext;
219     }
220 }
```

Here is the caller graph for this function:



13.162.4.10 static void free_linkedlist (linkedlist_data * ll) [static]

Definition at line 227 of file zip.c.

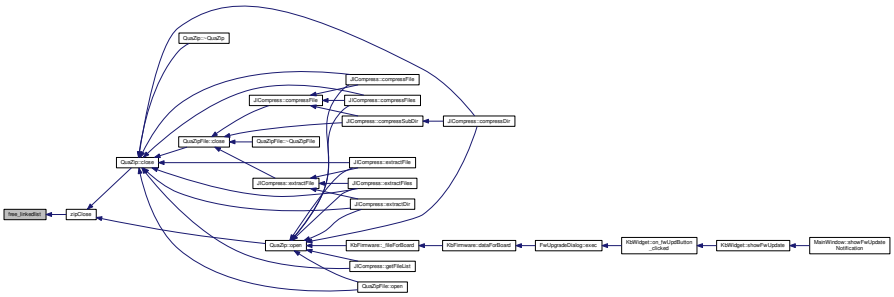
References linkedlist_data_s::first_block, free_datablock(), and linkedlist_data_s::last_block.

Referenced by zipClose().

```
228 {
229     free_datablock(ll->first_block);
230     ll->first_block = ll->last_block = NULL;
231 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.162.4.11 static void init_linkedlist (linkedlist_data * ll) [static]

Definition at line 222 of file zip.c.

References linkedlist_data_s::first_block, and linkedlist_data_s::last_block.

Referenced by zipOpen3().

```
223 {
224     ll->first_block = ll->last_block = NULL;
225 }
```

Here is the caller graph for this function:

13.162.4.12 int LoadCentralDirectoryRecord (zip64_internal * *pziinit*)

Definition at line 649 of file zip.c.

References add_data_in_datablock(), zip64_internal::add_position_when_writting_offset, ALLOC, zip64_internal::begin_pos, zip64_internal::central_dir, zip64_internal::filestream, zip64_internal::flags, zip64_internal::globalcomment, zip64_internal::number_entry, SIZEDATA_INDATABLOCK, TRYFREE, zip64_internal::z_filefunc, ZCLOSE64, ZFAKECLOSE64, zip64local_getLong(), zip64local_getLong64(), zip64local_getShort(), zip64local_SearchCentralDir(), zip64local_SearchCentralDir64(), ZIP_AUTO_CLOSE, ZIP_BADZIPFILE, ZIP_ERRNO, ZIP_OK, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, and ZSEEK64.

Referenced by zipOpen3().

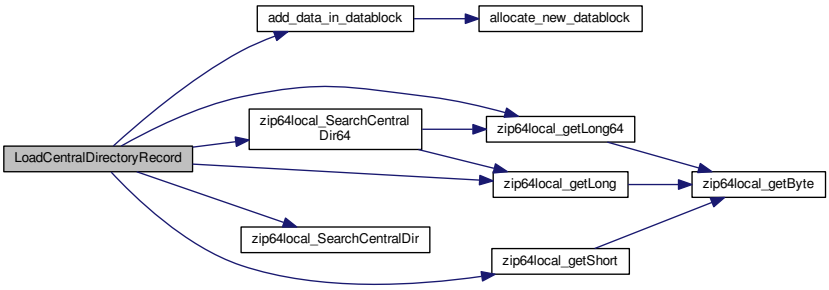
```
650 {
651   int err=ZIP_OK;
652   ZPOS64_T byte_before_the_zipfile;/* byte before the zipfile, (>0 for sfx)*/
653
654   ZPOS64_T size_central_dir;     /* size of the central directory  */
655   ZPOS64_T offset_central_dir;   /* offset of start of central directory */
656   ZPOS64_T central_pos;
657   uLong uL;
658
659   uLong number_disk;          /* number of the current dist, used for
660                                  spaning ZIP, unsupported, always 0*/
661   uLong number_disk_with_CD;  /* number the the disk with central dir, used
662                                  for spaning ZIP, unsupported, always 0*/
663   ZPOS64_T number_entry;
664   ZPOS64_T number_entry_CD;      /* total number of entries in
665                                    the central dir
666                                    (same than number_entry on nospan) */
667   uLong VersionMadeBy;
668   uLong VersionNeeded;
669   uLong size_comment;
670
671   int hasZIP64Record = 0;
672
673   /* check first if we find a ZIP64 record */
674   central_pos = zip64local_SearchCentralDir64(&pziinit->
      z_filefunc,pziinit->filestream);
675   if(central_pos > 0)
676   {
677     hasZIP64Record = 1;
678   }
679   else if(central_pos == 0)
680   {
681     central_pos = zip64local_SearchCentralDir(&pziinit->
      z_filefunc,pziinit->filestream);
682   }
683
684 /* disable to allow appending to empty ZIP archive
685         if (central_pos==0)
686             err=ZIP_ERRNO;
687 */
688
689   if(hasZIP64Record)
690   {
691     ZPOS64_T sizeEndOfCentralDirectory;
692     if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, central_pos,
      ZLIB_FILEFUNC_SEEK_SET) != 0)
693       err=ZIP_ERRNO;
694
695     /* the signature, already checked */
```

```
696     if (zip64local_getLong(&pziinit->z_filefunc, pziinit->
      filestream,&uL)!=ZIP_OK)
697       err=ZIP_ERRNO;
698
699     /* size of zip64 end of central directory record */
700     if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->
      filestream, &sizeEndOfCentralDirectory)!=ZIP_OK)
701       err=ZIP_ERRNO;
702
703     /* version made by */
704     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
      filestream, &VersionMadeBy)!=ZIP_OK)
705       err=ZIP_ERRNO;
706
707     /* version needed to extract */
708     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
      filestream, &VersionNeeded)!=ZIP_OK)
709       err=ZIP_ERRNO;
710
711     /* number of this disk */
712     if (zip64local_getLong(&pziinit->z_filefunc, pziinit->
      filestream,&number_disk)!=ZIP_OK)
713       err=ZIP_ERRNO;
714
715     /* number of the disk with the start of the central directory */
716     if (zip64local_getLong(&pziinit->z_filefunc, pziinit->
      filestream,&number_disk_with_CD)!=ZIP_OK)
717       err=ZIP_ERRNO;
718
719     /* total number of entries in the central directory on this disk */
720     if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->
      filestream, &number_entry)!=ZIP_OK)
721       err=ZIP_ERRNO;
722
723     /* total number of entries in the central directory */
724     if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->
      filestream,&number_entry_CD)!=ZIP_OK)
725       err=ZIP_ERRNO;
726
727     if ((number_entry_CD!=number_entry) || (number_disk_with_CD!=0) || (number_disk!=0))
728       err=ZIP_BADZIPFILE;
729
730     /* size of the central directory */
731     if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->
      filestream,&size_central_dir)!=ZIP_OK)
732       err=ZIP_ERRNO;
733
734     /* offset of start of central directory with respect to the
735     starting disk number */
736     if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->
      filestream,&offset_central_dir)!=ZIP_OK)
737       err=ZIP_ERRNO;
738
739     /* TODO.. */
740     /* read the comment from the standard central header. */
741     size_comment = 0;
742   }
743   else
744   {
745     /* Read End of central Directory info */
746     if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, central_pos,
      ZLIB_FILEFUNC_SEEK_SET)!=0)
747       err=ZIP_ERRNO;
748
749     /* the signature, already checked */
750     if (zip64local_getLong(&pziinit->z_filefunc, pziinit->
      filestream,&uL)!=ZIP_OK)
751       err=ZIP_ERRNO;
752
753     /* number of this disk */
754     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
      filestream,&number_disk)!=ZIP_OK)
755       err=ZIP_ERRNO;
756
757     /* number of the disk with the start of the central directory */
758     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
      filestream,&number_disk_with_CD)!=ZIP_OK)
759       err=ZIP_ERRNO;
760
761     /* total number of entries in the central dir on this disk */
762     number_entry = 0;
763     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
      filestream, &uL)!=ZIP_OK)
764       err=ZIP_ERRNO;
765     else
766       number_entry = uL;
767
```

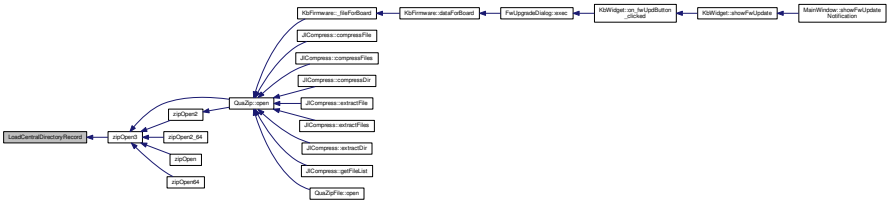
```
768     /* total number of entries in the central dir */
769     number_entry_CD = 0;
770     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
    filestream, &uL)!=ZIP_OK)
771       err=ZIP_ERRNO;
772     else
773       number_entry_CD = uL;
774
775     if ((number_entry_CD!=number_entry) || (number_disk_with_CD!=0) || (number_disk!=0))
776       err=ZIP_BADZIPFILE;
777
778     /* size of the central directory */
779     size_central_dir = 0;
780     if (zip64local_getLong(&pziinit->z_filefunc, pziinit->
    filestream, &uL)!=ZIP_OK)
781       err=ZIP_ERRNO;
782     else
783       size_central_dir = uL;
784
785     /* offset of start of central directory with respect to the starting disk number */
786     offset_central_dir = 0;
787     if (zip64local_getLong(&pziinit->z_filefunc, pziinit->
    filestream, &uL)!=ZIP_OK)
788       err=ZIP_ERRNO;
789     else
790       offset_central_dir = uL;
791
792
793     /* zipfile global comment length */
794     if (zip64local_getShort(&pziinit->z_filefunc, pziinit->
    filestream, &size_comment)!=ZIP_OK)
795       err=ZIP_ERRNO;
796   }
797
798   if ((central_pos<offset_central_dir+size_central_dir) &&
799     (err==ZIP_OK))
800     err=ZIP_BADZIPFILE;
801
802   if (err!=ZIP_OK)
803   {
804       if ((pziinit->flags & ZIP_AUTO_CLOSE) != 0) {
805         ZCLOSE64(pziinit->z_filefunc, pziinit->filestream);
806       } else {
807         ZFAKECLOSE64(pziinit->z_filefunc, pziinit->
    filestream);
808       }
809     return ZIP_ERRNO;
810   }
811
812   if (size_comment>0)
813   {
814     pziinit->globalcomment = (char*)ALLOC(size_comment+1);
815     if (pziinit->globalcomment)
816     {
817       size_comment = ZREAD64(pziinit->z_filefunc, pziinit->
    filestream, pziinit->globalcomment,size_comment);
818       pziinit->globalcomment[size_comment]=0;
819     }
820   }
821
822   byte_before_the_zipfile = central_pos - (offset_central_dir+size_central_dir);
823   pziinit->add_position_when_writting_offset = byte_before_the_zipfile;
824
825   {
826     ZPOS64_T size_central_dir_to_read = size_central_dir;
827     size_t buf_size = SIZEDATA_INDATABLOCK;
828     void* buf_read = (void*)ALLOC(buf_size);
829     if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, offset_central_dir +
    byte_before_the_zipfile, ZLIB_FILEFUNC_SEEK_SET) != 0)
830       err=ZIP_ERRNO;
831
832     while ((size_central_dir_to_read>0) && (err==ZIP_OK))
833     {
834       ZPOS64_T read_this = SIZEDATA_INDATABLOCK;
835       if (read_this > size_central_dir_to_read)
836         read_this = size_central_dir_to_read;
837
838       if (ZREAD64(pziinit->z_filefunc, pziinit->filestream,buf_read,(uLong)
    read_this) != read_this)
839         err=ZIP_ERRNO;
840
841       if (err==ZIP_OK)
842         err = add_data_in_datablock(&pziinit->central_dir,buf_read, (uLong)
    read_this);
843
844       size_central_dir_to_read-=read_this;
845     }
```

```
846     TRYFREE(buf_read);
847   }
848   pziinit->begin_pos = byte_before_the_zipfile;
849   pziinit->number_entry = number_entry_CD;
850
851   if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, offset_central_dir+
      byte_before_the_zipfile,ZLIB_FILEFUNC_SEEK_SET) != 0)
852     err=ZIP_ERRNO;
853
854   return err;
855 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.13 int Write_EndOfCentralDirectoryRecord (zip64_internal * *zi*, uLong *size_centraldir*, ZPOS64_T *centraldir_pos_inzip*)

Definition at line 1876 of file zip.c.

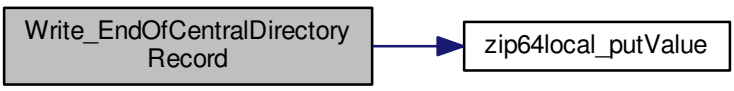
References zip64_internal::add_position_when_writting_offset, ENDHEADERMAGIC, zip64_internal::filestream, zip64_internal::number_entry, zip64_internal::z_filefunc, zip64local_putValue(), and ZIP_OK.

Referenced by zipClose().

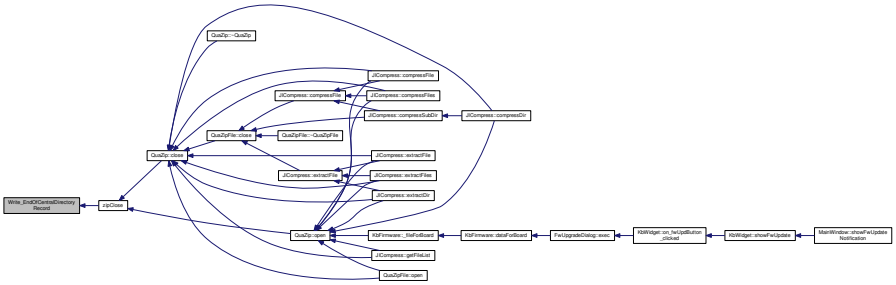
```
1877 {
1878   int err = ZIP_OK;
1879
1880   /*signature*/
1881   err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)ENDHEADERMAGIC,4);
1882
1883   if (err==ZIP_OK) /* number of this disk */
1884     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0,2);
1885
1886   if (err==ZIP_OK) /* number of the disk with the start of the central directory */
1887     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0,2);
1888
1889   if (err==ZIP_OK) /* total number of entries in the central dir on this disk */
```

```
1890   {
1891     {
1892       if(zi->number_entry >= 0xFFFF)
1893         err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,(uLong)0xffff,2); /* use value in ZIP64 record */
1894       else
1895         err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,(uLong)zi->number_entry,2);
1896     }
1897   }
1898
1899   if (err==ZIP_OK) /* total number of entries in the central dir */
1900   {
1901     if(zi->number_entry >= 0xFFFF)
1902       err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,(uLong)0xffff,2); /* use value in ZIP64 record */
1903     else
1904       err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,(uLong)zi->number_entry,2);
1905   }
1906
1907   if (err==ZIP_OK) /* size of the central directory */
1908     err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,(uLong)size_centraldir,4);
1909
1910   if (err==ZIP_OK) /* offset of start of central directory with respect to the starting disk number
     */
1911   {
1912     ZPOS64_T pos = centraldir_pos_inzip - zi->
     add_position_when_writting_offset;
1913     if(pos >= 0xffffffff)
1914     {
1915       err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream, (uLong)0xffffffff,4);
1916     }
1917     else
1918                   err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream, (uLong)(centraldir_pos_inzip - zi->
     add_position_when_writting_offset),4);
1919   }
1920
1921    return err;
1922 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.14 int Write_GlobalComment (zip64_internal * *zi,* const char * *global_comment*)

Definition at line 1924 of file zip.c.

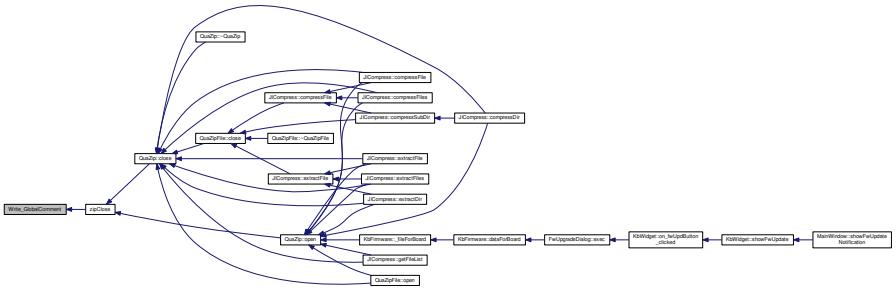
References zip64_internal::filestream, zip64_internal::z_filefunc, zip64local_putValue(), ZIP_ERRNO, ZIP_OK, and ZWRITE64.

Referenced by zipClose().

```
1925 {
1926   int err = ZIP_OK;
1927   uInt size_global_comment = 0;
1928
1929   if(global_comment != NULL)
1930     size_global_comment = (uInt)strlen(global_comment);
1931
1932   err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)size_global_comment,2);
1933
1934   if (err == ZIP_OK && size_global_comment > 0)
1935   {
1936     if (ZWRITE64(zi->z_filefunc,zi->filestream, global_comment,
      size_global_comment) != size_global_comment)
1937       err = ZIP_ERRNO;
1938   }
1939   return err;
1940 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.162.4.15 int Write_LocalFileHeader (zip64_internal * *zi,* const char * *filename,* uInt *size_extrafield_local,* const void * *extrafield_local,* uLong *version_to_extract*)

Definition at line 977 of file zip.c.

References zip64_internal::ci, curfile64_info::dosDate, zip64_internal::filestream, curfile64_info::flag, LOCALHEADERMAGIC, curfile64_info::method, curfile64_info::pos_zip64extrainfo, zip64_internal::z_filefunc, curfile64_info::zip64, zip64local_putValue(), ZIP_ERRNO, ZIP_OK, ZTELL64, and ZWRITE64.

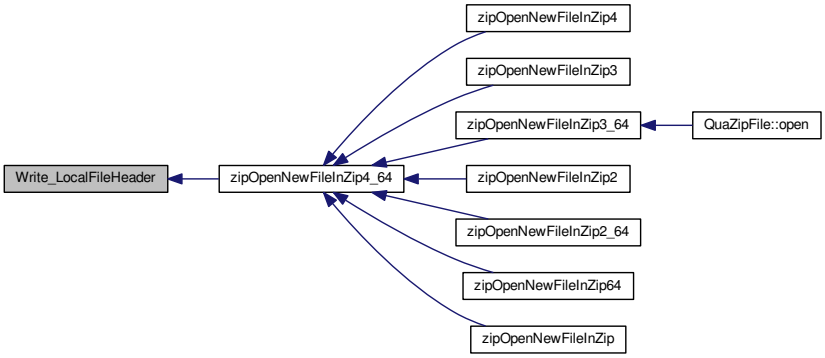
Referenced by zipOpenNewFileInZip4_64().

```
981 {
982   /* write the local header */
983   int err;
984   uInt size_filename = (uInt)strlen(filename);
985   uInt size_extrafield = size_extrafield_local;
986 
987   err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)LOCALHEADERMAGIC, 4);
988 
989   if (err==ZIP_OK)
990   {
991     if(zi->ci.zip64)
992       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)45,2);/* version needed to extract */
993     else
994       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)version_to_extract,2);
995   }
996 
997   if (err==ZIP_OK)
998     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)zi->ci.flag,2);
999 
1000   if (err==ZIP_OK)
1001     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)zi->ci.method,2);
1002 
1003   if (err==ZIP_OK)
1004     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)zi->ci.dosDate,4);
1005 
1006   /* CRC / Compressed size / Uncompressed size will be filled in later and rewritten later */
1007   if (err==ZIP_OK)
1008     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0,4); /* crc 32, unknown */
1009   if (err==ZIP_OK)
1010   {
1011     if(zi->ci.zip64)
1012       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0xFFFFFFFF,4); /* compressed size, unknown */
1013     else
1014       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0,4); /* compressed size, unknown */
1015   }
1016   if (err==ZIP_OK)
1017   {
1018     if(zi->ci.zip64)
1019       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0xFFFFFFFF,4); /* uncompressed size, unknown */
1020     else
1021       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0,4); /* uncompressed size, unknown */
1022   }
1023 
1024   if (err==ZIP_OK)
1025     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)size_filename,2);
1026 
1027   if(zi->ci.zip64)
1028   {
1029     size_extrafield += 20;
1030   }
1031 
1032   if (err==ZIP_OK)
1033     err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)size_extrafield,2);
1034 
1035   if ((err==ZIP_OK) && (size_filename > 0))
1036   {
1037     if (ZWRITE64(zi->z_filefunc,zi->filestream,filename,size_filename)!=
      size_filename)
1038       err = ZIP_ERRNO;
1039   }
1040 
1041   if ((err==ZIP_OK) && (size_extrafield_local > 0))
1042   {
1043     if (ZWRITE64(zi->z_filefunc, zi->filestream, extrafield_local,
      size_extrafield_local) != size_extrafield_local)
1044       err = ZIP_ERRNO;
1045   }
1046 
1047 
1048   if ((err==ZIP_OK) && (zi->ci.zip64))
1049   {
1050       /* write the Zip64 extended info */
1051       short HeaderID = 1;
1052       short DataSize = 16;
```

```
1053       ZPOS64_T CompressedSize = 0;
1054       ZPOS64_T UncompressedSize = 0;
1055
1056       /* Remember position of Zip64 extended info for the local file header. (needed when we update size
     after done with file) */
1057       zi->ci.pos_zip64extrainfo = ZTELL64(zi->
    z_filefunc,zi->filestream);
1058
1059       err = zip64local_putValue(&zi->z_filefunc, zi->
    filestream, (short)HeaderID,2);
1060       err = zip64local_putValue(&zi->z_filefunc, zi->
    filestream, (short)DataSize,2);
1061
1062       err = zip64local_putValue(&zi->z_filefunc, zi->
    filestream, (ZPOS64_T)UncompressedSize,8);
1063       err = zip64local_putValue(&zi->z_filefunc, zi->
    filestream, (ZPOS64_T)CompressedSize,8);
1064     }
1065
1066     return err;
1067 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.162.4.16 int Write_Zip64EndOfCentralDirectoryLocator (zip64_internal * *zi*, ZPOS64_T *zip64eocd_pos_inzip*)

Definition at line 1815 of file zip.c.

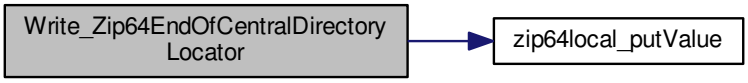
References zip64_internal::add_position_when_writting_offset, zip64_internal::filestream, zip64_internal::z_filefunc, ZIP64ENDLOCHEADERMAGIC, zip64local_putValue(), and ZIP_OK.

Referenced by zipClose().

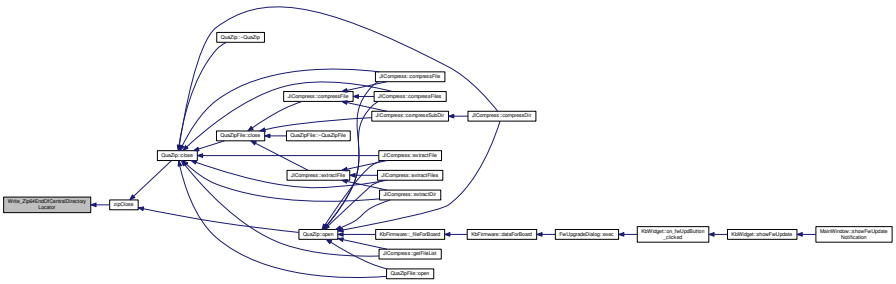
```
1816 {
1817   int err = ZIP_OK;
```

```
1818   ZPOS64_T pos = zip64eocd_pos_inzip - zi->
      add_position_when_writting_offset;
1819
1820   err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)ZIP64ENDLOCHEADERMAGIC,4);
1821
1822   /*num disks*/
1823     if (err==ZIP_OK) /* number of the disk with the start of the central directory */
1824       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)0,4);
1825
1826   /*relative offset*/
1827     if (err==ZIP_OK) /* Relative offset to the Zip64EndOfCentralDirectory */
1828       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream, pos,8);
1829
1830   /*total disks*/ /* Do not support spawning of disk so always say 1 here*/
1831     if (err==ZIP_OK) /* number of the disk with the start of the central directory */
1832       err = zip64local_putValue(&zi->z_filefunc,zi->
      filestream,(uLong)1,4);
1833
1834     return err;
1835 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.17 int Write_Zip64EndOfCentralDirectoryRecord (zip64_internal * *zi*, uLong *size_centraldir*, ZPOS64_T *centraldir_pos_inzip*)

Definition at line 1837 of file zip.c.

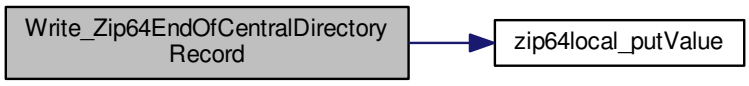
References zip64_internal::add_position_when_writting_offset, zip64_internal::filestream, zip64_internal::number-_entry, zip64_internal::z_filefunc, ZIP64ENDHEADERMAGIC, zip64local_putValue(), and ZIP_OK.

Referenced by zipClose().

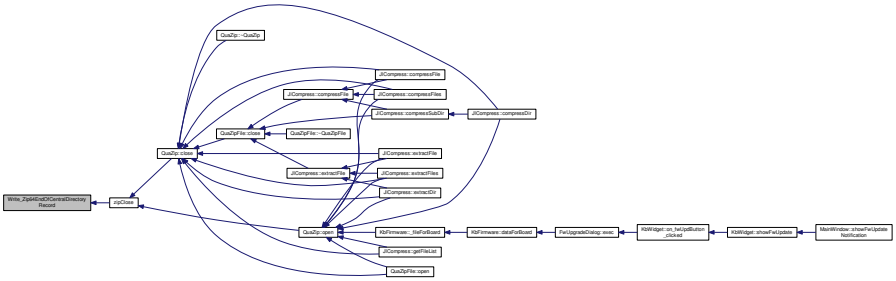
```
1838 {
1839   int err = ZIP_OK;
1840
1841   uLong Zip64DataSize = 44;
1842
1843   err = zip64local_putValue(&zi->z_filefunc,zi->
```

```
    filestream,(uLong)ZIP64ENDHEADERMAGIC,4);
1844
1845   if (err==ZIP_OK) /* size of this 'zip64 end of central directory' */
1846     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream,(ZPOS64_T)Zip64DataSize,8); /* why ZPOS64_T of this ? */
1847
1848   if (err==ZIP_OK) /* version made by */
1849     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream,(uLong)45,2);
1850
1851   if (err==ZIP_OK) /* version needed */
1852     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream,(uLong)45,2);
1853
1854   if (err==ZIP_OK) /* number of this disk */
1855     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream,(uLong)0,4);
1856
1857   if (err==ZIP_OK) /* number of the disk with the start of the central directory */
1858     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream,(uLong)0,4);
1859
1860   if (err==ZIP_OK) /* total number of entries in the central dir on this disk */
1861     err = zip64local_putValue(&zi->z_filefunc, zi->
    filestream, zi->number_entry, 8);
1862
1863   if (err==ZIP_OK) /* total number of entries in the central dir */
1864     err = zip64local_putValue(&zi->z_filefunc, zi->
    filestream, zi->number_entry, 8);
1865
1866   if (err==ZIP_OK) /* size of the central directory */
1867     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream,(ZPOS64_T)size_centraldir,8);
1868
1869   if (err==ZIP_OK) /* offset of start of central directory with respect to the starting disk number
    */
1870   {
1871     ZPOS64_T pos = centraldir_pos_inzip - zi->
    add_position_when_writting_offset;
1872     err = zip64local_putValue(&zi->z_filefunc,zi->
    filestream, (ZPOS64_T)pos,8);
1873   }
1874   return err;
1875 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.18 static int zip64FlushWriteBuffer (zip64_internal * *zi*) [static]

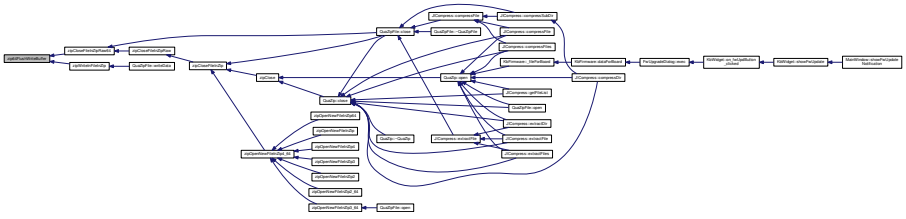
Definition at line 1403 of file zip.c.

References curfile64_info::buffered_data, zip64_internal::ci, curfile64_info::encrypt, zip64_internal::filestream, curfile64_info::keys, curfile64_info::method, curfile64_info::pcrc_32_tab, curfile64_info::pos_in_buffered_data, curfile64_info::stream, curfile64_info::totalCompressedData, curfile64_info::totalUncompressedData, Z_BZIP2ED, zip64_internal::z_filefunc, zencode, ZIP_ERRNO, ZIP_OK, and ZWRITE64.

Referenced by zipCloseFileInZipRaw64(), and zipWriteInFileInZip().

```
1404 {
1405     int err=ZIP_OK;
1406
1407     if (zi->ci.encrypt != 0)
1408     {
1409 #ifndef NOCRYPT
1410         uInt i;
1411         int t;
1412         for (i=0;i<zi->ci.pos_in_buffered_data;i++)
1413             zi->ci.buffered_data[i] = zencode(zi->ci.
      keys, zi->ci.pcrc_32_tab, zi->ci.buffered_data[i],t);
1414 #endif
1415     }
1416
1417     if (ZWRITE64(zi->z_filefunc,zi->filestream,zi->
      ci.buffered_data,zi->ci.pos_in_buffered_data) != zi->
      ci.pos_in_buffered_data)
1418       err = ZIP_ERRNO;
1419
1420     zi->ci.totalCompressedData += zi->ci.
      pos_in_buffered_data;
1421
1422 #ifdef HAVE_BZIP2
1423     if(zi->ci.method == Z_BZIP2ED)
1424     {
1425       zi->ci.totalUncompressedData += zi->ci.bstream.total_in_lo32;
1426       zi->ci.bstream.total_in_lo32 = 0;
1427       zi->ci.bstream.total_in_hi32 = 0;
1428     }
1429     else
1430 #endif
1431     {
1432       zi->ci.totalUncompressedData += zi->ci.stream.total_in;
1433       zi->ci.stream.total_in = 0;
1434     }
1435
1436
1437     zi->ci.pos_in_buffered_data = 0;
1438
1439     return err;
1440 }
```

Here is the caller graph for this function:



13.162.4.19 static int zip64local_getByte (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* int * *pi*) [static]

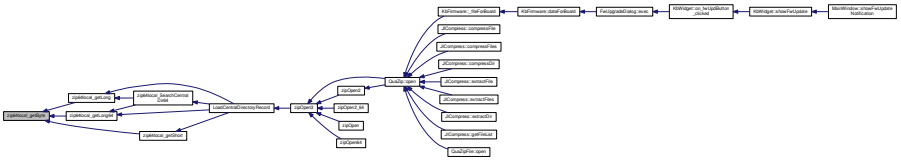
Definition at line 358 of file zip.c.

References ZERROR64, ZIP_EOF, ZIP_ERRNO, ZIP_OK, and ZREAD64.

Referenced by zip64local_getLong(), zip64local_getLong64(), and zip64local_getShort().

```
359 {
360     unsigned char c;
361     int err = (int)ZREAD64(*pzlib_filefunc_def,filestream,&c,1);
362     if (err==1)
363     {
364         *pi = (int)c;
365         return ZIP_OK;
366     }
367     else
368     {
369         if (ZERROR64(*pzlib_filefunc_def,filestream))
370             return ZIP_ERRNO;
371         else
372             return ZIP_EOF;
373     }
374 }
```

Here is the caller graph for this function:



13.162.4.20 static int zip64local_getLong (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* uLong * *pX*) [static]

Definition at line 404 of file zip.c.

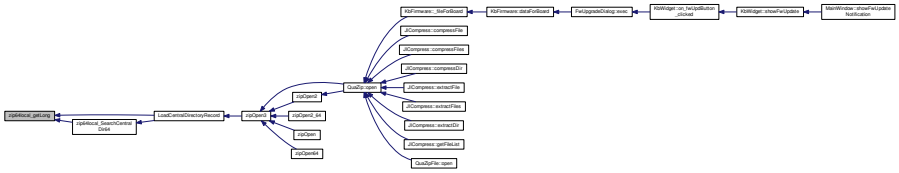
References x, zip64local_getByte(), and ZIP_OK.

Referenced by LoadCentralDirectoryRecord(), and zip64local_SearchCentralDir64().

```
405 {
406     uLong x ;
407     int i = 0;
408     int err;
409
410     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
411     x = (uLong)i;
412
413     if (err==ZIP_OK)
414         err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
415     x += ((uLong)i)<<8;
416
417     if (err==ZIP_OK)
418         err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
419     x += ((uLong)i)<<16;
420
421     if (err==ZIP_OK)
422         err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
423     x += ((uLong)i)<<24;
424
425     if (err==ZIP_OK)
426         *pX = x;
427     else
428         *pX = 0;
429     return err;
430 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.162.4.21 static int zip64local_getLong64 (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* ZPOS64_T * *pX*) [static]

Definition at line 435 of file zip.c.

References x, zip64local_getByte(), and ZIP_OK.

Referenced by LoadCentralDirectoryRecord(), and zip64local_SearchCentralDir64().

```
436 {
437   ZPOS64_T x;
438   int i = 0;
439   int err;
440
441   err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
442   x = (ZPOS64_T)i;
443
444   if (err==ZIP_OK)
445     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
446   x += ((ZPOS64_T)i)<<8;
447
448   if (err==ZIP_OK)
449     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
450   x += ((ZPOS64_T)i)<<16;
451
452   if (err==ZIP_OK)
453     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
454   x += ((ZPOS64_T)i)<<24;
455
456   if (err==ZIP_OK)
457     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
458   x += ((ZPOS64_T)i)<<32;
459
460   if (err==ZIP_OK)
461     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
462   x += ((ZPOS64_T)i)<<40;
463
464   if (err==ZIP_OK)
465     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
466   x += ((ZPOS64_T)i)<<48;
467
468   if (err==ZIP_OK)
469     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
470   x += ((ZPOS64_T)i)<<56;
471
472   if (err==ZIP_OK)
473     *pX = x;
```

```
474   else
475     *pX = 0;
476
477   return err;
478 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

13.162.4.22 static int zip64local_getShort (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* uLong * *pX*) [static]

Definition at line 382 of file zip.c.

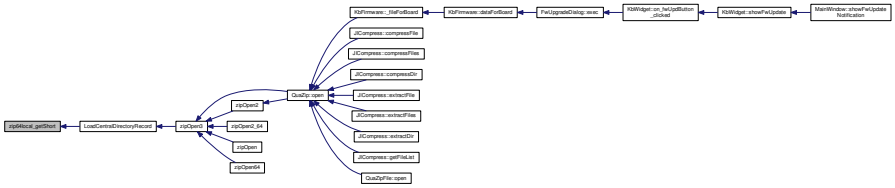
References x, zip64local_getByte(), and ZIP_OK.

Referenced by LoadCentralDirectoryRecord().

```
383 {
384     uLong x ;
385     int i = 0;
386     int err;
387
388     err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
389     x = (uLong)i;
390
391     if (err==ZIP_OK)
392         err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
393     x += ((uLong)i)<<8;
394
395     if (err==ZIP_OK)
396         *pX = x;
397     else
398         *pX = 0;
399     return err;
400 }
```

Here is the call graph for this function:

Here is the caller graph for this function:



13.162.4.23 static int zip64local_putValue (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream,* ZPOS64_T *x,* int *nbByte*) [static]

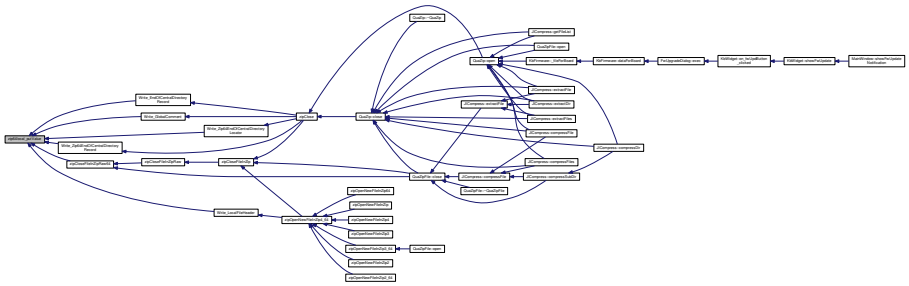
Definition at line 296 of file zip.c.

References ZIP_ERRNO, ZIP_OK, and ZWRITE64.

Referenced by Write_EndOfCentralDirectoryRecord(), Write_GlobalComment(), Write_LocalFileHeader(), Write_-Zip64EndOfCentralDirectoryLocator(), Write_Zip64EndOfCentralDirectoryRecord(), and zipCloseFileInZipRaw64().

```
297 {
298     unsigned char buf[8];
299     int n;
300     for (n = 0; n < nbByte; n++)
301     {
302         buf[n] = (unsigned char)(x & 0xff);
303         x >>= 8;
304     }
305     if (x != 0)
306       {     /* data overflow - hack for ZIP64 (X Roche) */
307       for (n = 0; n < nbByte; n++)
308         {
309           buf[n] = 0xff;
310         }
311       }
312
313     if (ZWRITE64(*pzlib_filefunc_def,filestream,buf,nbByte)!=(uLong)nbByte)
314         return ZIP_ERRNO;
315     else
316         return ZIP_OK;
317 }
```

Here is the caller graph for this function:



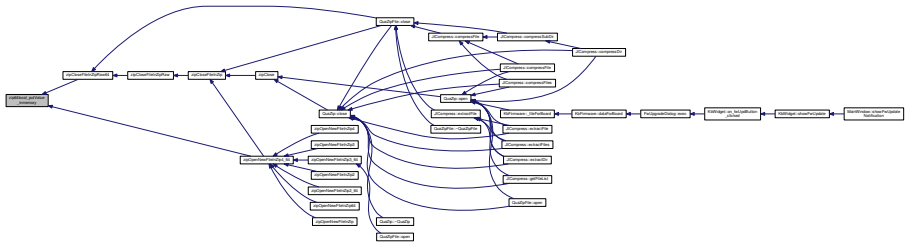
13.162.4.24 static void zip64local_putValue_inmemory (void * *dest*, ZPOS64_T *x*, int *nbByte*) [static]

Definition at line 320 of file zip.c.

Referenced by zipCloseFileInZipRaw64(), and zipOpenNewFileInZip4_64().

```
321 {
322     unsigned char* buf=(unsigned char*)dest;
323     int n;
324     for (n = 0; n < nbByte; n++) {
325         buf[n] = (unsigned char)(x & 0xff);
326         x >>= 8;
327     }
328
329     if (x != 0)
330     {     /* data overflow - hack for ZIP64 */
331        for (n = 0; n < nbByte; n++)
332        {
333           buf[n] = 0xff;
334        }
335     }
336 }
```

Here is the caller graph for this function:



13.162.4.25 static ZPOS64_T zip64local_SearchCentralDir (const zlib_filefunc64_32_def * *pzlib_filefunc_def*, voidpf *filestream*) [static]

Definition at line 489 of file zip.c.

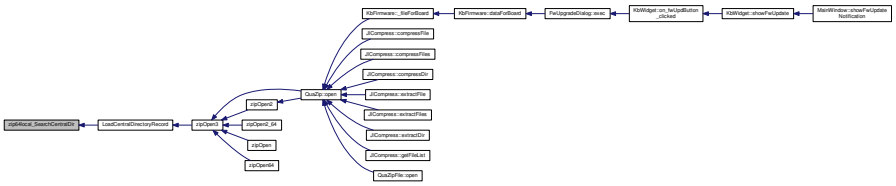
References ALLOC, BUFREADCOMMENT, TRYFREE, ZLIB_FILEFUNC_SEEK_END, ZLIB_FILEFUNC_SEEK_-SET, ZREAD64, ZSEEK64, and ZTELL64.

Referenced by LoadCentralDirectoryRecord().

```
490 {
491   unsigned char* buf;
492   ZPOS64_T uSizeFile;
```

```
493   ZPOS64_T uBackRead;
494   ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
495   ZPOS64_T uPosFound=0;
496
497   if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
498     return 0;
499
500
501   uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);
502
503   if (uMaxBack>uSizeFile)
504     uMaxBack = uSizeFile;
505
506   buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
507   if (buf==NULL)
508     return 0;
509
510   uBackRead = 4;
511   while (uBackRead<uMaxBack)
512   {
513     uLong uReadSize;
514     ZPOS64_T uReadPos ;
515     int i;
516     if (uBackRead+BUFREADCOMMENT>uMaxBack)
517       uBackRead = uMaxBack;
518     else
519       uBackRead+=BUFREADCOMMENT;
520     uReadPos = uSizeFile-uBackRead ;
521
522     uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
523       (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
524     if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
525       break;
526
527     if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
528       break;
529
530     for (i=(int)uReadSize-3; (i--)>0;)
531       if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) &&
532         ((*(buf+i+2))==0x05) && ((*(buf+i+3))==0x06))
533       {
534         uPosFound = uReadPos+i;
535         break;
536       }
537
538     if (uPosFound!=0)
539       break;
540   }
541   TRYFREE(buf);
542   return uPosFound;
543 }
```

Here is the caller graph for this function:



13.162.4.26 static ZPOS64_T zip64local_SearchCentralDir64 (const zlib_filefunc64_32_def * *pzlib_filefunc_def,* voidpf *filestream*) [static]

Definition at line 551 of file zip.c.

References ALLOC, BUFREADCOMMENT, TRYFREE, zip64local_getLong(), zip64local_getLong64(), ZIP_OK, ZLIB_FILEFUNC_SEEK_END, ZLIB_FILEFUNC_SEEK_SET, ZREAD64, ZSEEK64, and ZTELL64.

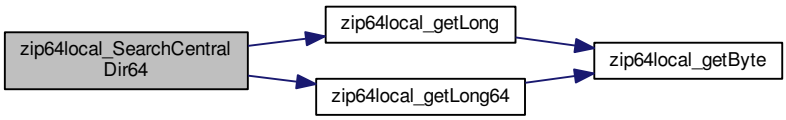
Referenced by LoadCentralDirectoryRecord().

```
552 {
```

```
553   unsigned char* buf;
554   ZPOS64_T uSizeFile;
555   ZPOS64_T uBackRead;
556   ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
557   ZPOS64_T uPosFound=0;
558   uLong uL;
559   ZPOS64_T relativeOffset;
560
561   if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
562     return 0;
563
564   uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);
565
566   if (uMaxBack>uSizeFile)
567     uMaxBack = uSizeFile;
568
569   buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
570   if (buf==NULL)
571     return 0;
572
573   uBackRead = 4;
574   while (uBackRead<uMaxBack)
575   {
576     uLong uReadSize;
577     ZPOS64_T uReadPos;
578     int i;
579     if (uBackRead+BUFREADCOMMENT>uMaxBack)
580       uBackRead = uMaxBack;
581     else
582       uBackRead+=BUFREADCOMMENT;
583     uReadPos = uSizeFile-uBackRead ;
584
585     uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
586       (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
587     if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
588       break;
589
590     if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
591       break;
592
593     for (i=(int)uReadSize-3; (i--)>0;)
594     {
595       /* Signature "0x07064b50" Zip64 end of central directory locater */
596       if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) && ((*(buf+i+2))==0x06) && ((*(buf+i+3))==0x07))
597       {
598         uPosFound = uReadPos+i;
599         break;
600       }
601     }
602
603       if (uPosFound!=0)
604         break;
605   }
606
607   TRYFREE(buf);
608   if (uPosFound == 0)
609     return 0;
610
611   /* Zip64 end of central directory locator */
612   if (ZSEEK64(*pzlib_filefunc_def,filestream, uPosFound,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
613     return 0;
614
615   /* the signature, already checked */
616   if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    ZIP_OK)
617     return 0;
618
619   /* number of the disk with the start of the zip64 end of  central directory */
620   if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    ZIP_OK)
621     return 0;
622   if (uL != 0)
623     return 0;
624
625   /* relative offset of the zip64 end of central directory record */
626   if (zip64local_getLong64(pzlib_filefunc_def,filestream,&relativeOffset)!=
    ZIP_OK)
627     return 0;
628
629   /* total number of disks */
630   if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    ZIP_OK)
631     return 0;
632   if (uL != 1)
633     return 0;
```

```
634
635   /* Goto Zip64 end of central directory record */
636   if (ZSEEK64(*pzlib_filefunc_def,filestream, relativeOffset,
      ZLIB_FILEFUNC_SEEK_SET)!=0)
637     return 0;
638
639   /* the signature */
640   if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
      ZIP_OK)
641     return 0;
642
643   if (uL != 0x06064b50) /* signature of 'Zip64 end of central directory' */
644     return 0;
645
646   return relativeOffset;
647 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.162.4.27 static uLong zip64local_TmzDateToDosDate (const tm_zip * *ptm*) [static]

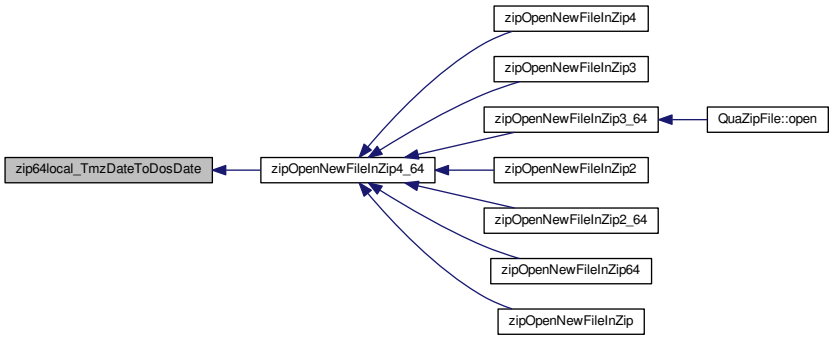
Definition at line 341 of file zip.c.

References tm_zip_s::tm_hour, tm_zip_s::tm_mday, tm_zip_s::tm_min, tm_zip_s::tm_mon, tm_zip_s::tm_sec, and tm_zip_s::tm_year.

Referenced by zipOpenNewFileInZip4_64().

```
342 {
343     uLong year = (uLong)ptm->tm_year;
344     if (year>=1980)
345         year-=1980;
346     else if (year>=80)
347         year-=80;
348     return
349       (uLong) (((ptm->tm_mday) + (32 * (ptm->tm_mon+1)) + (512 * year)) << 16) |
350         ((ptm->tm_sec/2) + (32* ptm->tm_min) + (2048 * (uLong)ptm->
      tm_hour));
351 }
```

Here is the caller graph for this function:



13.162.4.28 int ZEXPORT zipClearFlags (zipFile *file*, unsigned *flags*)

Definition at line 2091 of file zip.c.

References zip64_internal::flags, ZIP_OK, ZIP_PARAMERROR, ZIP_SEQUENTIAL, and ZIP_WRITE_DATA_DESCRIPTOR.

Referenced by QuaZipFile::open().

```
2092 {
2093     zip64_internal* zi;
2094     if (file == NULL)
2095         return ZIP_PARAMERROR;
2096     zi = (zip64_internal*)file;
2097     zi->flags &= ~flags;
2098     // If the data descriptor is not written, we can't use a non-seekable output.
2099     if ((zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) == 0) {
2100         zi->flags &= ~ZIP_SEQUENTIAL;
2101     }
2102     return ZIP_OK;
2103 }
```

Here is the caller graph for this function:

13.162.4.29 int ZEXPORT zipClose (zipFile *file*, const char * *global_comment*)

Definition at line 1942 of file zip.c.

References zip64_internal::add_position_when_writting_offset, zip64_internal::central_dir, linkedlist_datablock_internal_s::data, zip64_internal::filestream, linkedlist_datablock_internal_s::filled_in_this_block, linkedlist_data_s::first_block, zip64_internal::flags, free_linkedlist(), zip64_internal::globalcomment, zip64_internal::in_opened_file_inzip, linkedlist_datablock_internal_s::next_datablock, zip64_internal::number_entry, TRYFREE, Write_EndOf-

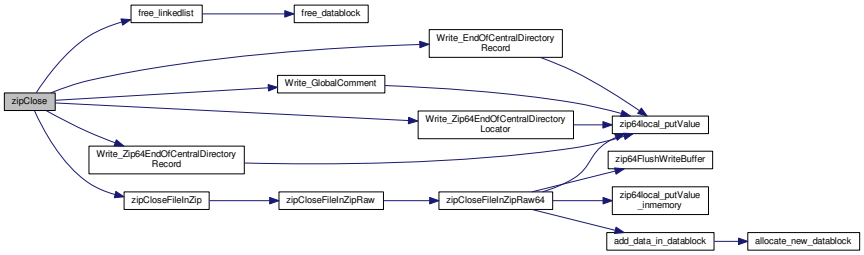
CentralDirectoryRecord(), Write_GlobalComment(), Write_Zip64EndOfCentralDirectoryLocator(), Write_Zip64EndOfCentralDirectoryRecord(), zip64_internal::z_filefunc, ZCLOSE64, ZFAKECLOSE64, ZIP_AUTO_CLOSE, ZIP_ERRNO, ZIP_OK, ZIP_PARAMERROR, zipCloseFileInZip(), ZTELL64, and ZWRITE64.

Referenced by QuaZip::close(), and QuaZip::open().

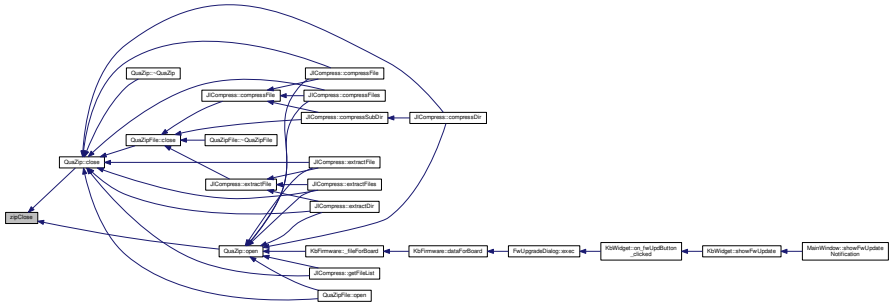
```
1943 {
1944     zip64_internal* zi;
1945     int err = 0;
1946     uLong size_centraldir = 0;
1947     ZPOS64_T centraldir_pos_inzip;
1948     ZPOS64_T pos;
1949 
1950     if (file == NULL)
1951         return ZIP_PARAMERROR;
1952 
1953     zi = (zip64_internal*)file;
1954 
1955     if (zi->in_opened_file_inzip == 1)
1956     {
1957         err = zipCloseFileInZip (file);
1958     }
1959 
1960 #ifndef NO_ADDFILEINEXISTINGZIP
1961     if (global_comment==NULL)
1962         global_comment = zi->globalcomment;
1963 #endif
1964 
1965     centraldir_pos_inzip = ZTELL64(zi->z_filefunc,zi->
     filestream);
1966 
1967     if (err==ZIP_OK)
1968     {
1969         linkedlist_datablock_internal* ldi = zi->
     central_dir.first_block;
1970         while (ldi!=NULL)
1971         {
1972             if ((err==ZIP_OK) && (ldi->filled_in_this_block>0))
1973             {
1974                 if (ZWRITE64(zi->z_filefunc,zi->filestream, ldi->
     data, ldi->filled_in_this_block) != ldi->
     filled_in_this_block)
1975                     err = ZIP_ERRNO;
1976             }
1977 
1978             size_centraldir += ldi->filled_in_this_block;
1979             ldi = ldi->next_datablock;
1980         }
1981     }
1982     free_linkedlist(&(zi->central_dir));
1983 
1984     pos = centraldir_pos_inzip - zi->add_position_when_writting_offset;
1985     if(pos >= 0xffffffff || zi->number_entry > 0xFFFF)
1986     {
1987       ZPOS64_T Zip64EOCDpos = ZTELL64(zi->z_filefunc,zi->
     filestream);
1988       Write_Zip64EndOfCentralDirectoryRecord(zi, size_centraldir,
     centraldir_pos_inzip);
1989 
1990       Write_Zip64EndOfCentralDirectoryLocator(zi, Zip64EOCDpos);
1991     }
1992 
1993     if (err==ZIP_OK)
1994       err = Write_EndOfCentralDirectoryRecord(zi, size_centraldir,
     centraldir_pos_inzip);
1995 
1996     if(err == ZIP_OK)
1997       err = Write_GlobalComment(zi, global_comment);
1998 
1999     if ((zi->flags & ZIP_AUTO_CLOSE) != 0) {
2000         if (ZCLOSE64(zi->z_filefunc,zi->filestream) != 0) {
2001             if (err == ZIP_OK)
2002                 err = ZIP_ERRNO;
2003         }
2004     } else {
2005         if (ZFAKECLOSE64(zi->z_filefunc,zi->filestream) != 0) {
2006             if (err == ZIP_OK)
2007                 err = ZIP_ERRNO;
2008         }
2009     }
2010 
2011 #ifndef NO_ADDFILEINEXISTINGZIP
2012     TRYFREE(zi->globalcomment);
2013 #endif
2014     TRYFREE(zi);
2015 
```

```
2016     return err;
2017 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.30 int ZEXPORT zipCloseFileInZip (zipFile *file*)

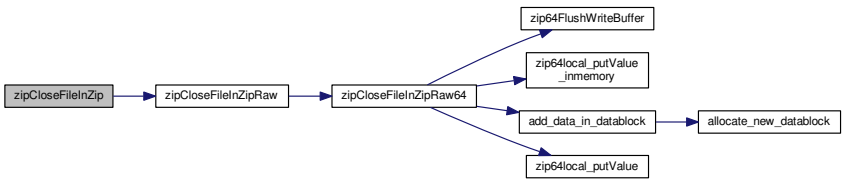
Definition at line 1810 of file zip.c.

References zipCloseFileInZipRaw().

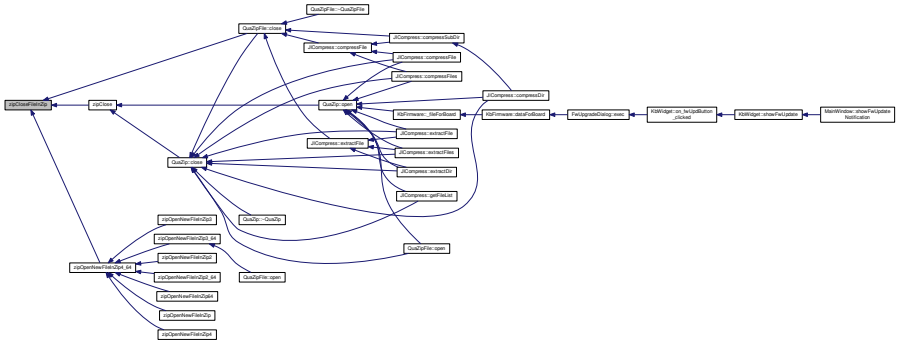
Referenced by QuaZipFile::close(), zipClose(), and zipOpenNewFileInZip4_64().

```
1811 {
1812     return zipCloseFileInZipRaw (file,0,0);
1813 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.31 int ZEXPORT zipCloseFileInZipRaw (zipFile *file*, uLong *uncompressed_size*, uLong *crc32*)

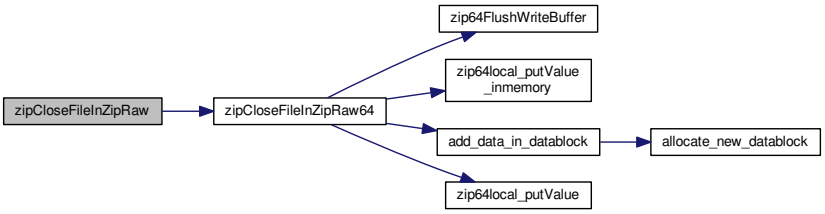
Definition at line 1549 of file zip.c.

References zipCloseFileInZipRaw64().

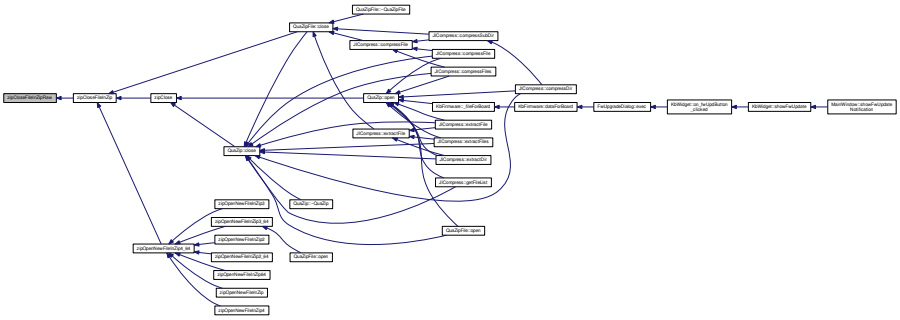
Referenced by zipCloseFileInZip().

```
1550 {
1551     return zipCloseFileInZipRaw64 (file, uncompressed_size, crc32);
1552 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.32 int ZEXPORT zipCloseFileInZipRaw64 (zipFile *file*, ZPOS64_T *uncompressed_size*, uLong *crc32*)

Definition at line 1554 of file zip.c.

References add_data_in_datablock(), curfile64_info::buffered_data, zip64_internal::central_dir, curfile64_info::central_header, zip64_internal::ci, curfile64_info::crc32, curfile64_info::crypt_header_size, DESCRIPTORHEADERMAGIC, zip64_internal::filestream, curfile64_info::flag, zip64_internal::flags, zip64_internal::in_opened_file_inzip, curfile64_info::method, zip64_internal::number_entry, curfile64_info::pos_in_buffered_data, curfile64_info::pos_local_header, curfile64_info::pos_zip64extrainfo, curfile64_info::raw, curfile64_info::size_centralExtra, curfile64_info::size_centralExtraFree, curfile64_info::size_centralheader, curfile64_info::stream, curfile64_info::stream_initialised, curfile64_info::totalCompressedData, curfile64_info::totalUncompressedData, Z_BUFSIZE, Z_BZIP2ED, zip64_internal::z_filefunc, curfile64_info::zip64, zip64FlushWriteBuffer(), zip64local_putValue(), zip64local_putValue_inmemory(), ZIP_BADZIPFILE, ZIP_ERRNO, ZIP_OK, ZIP_PARAMERROR, ZIP_SEQUENTIAL, ZLIB_FILEFUNC_SEEK_SET, ZSEEK64, and ZTELL64.

Referenced by QuaZipFile::close(), and zipCloseFileInZipRaw().

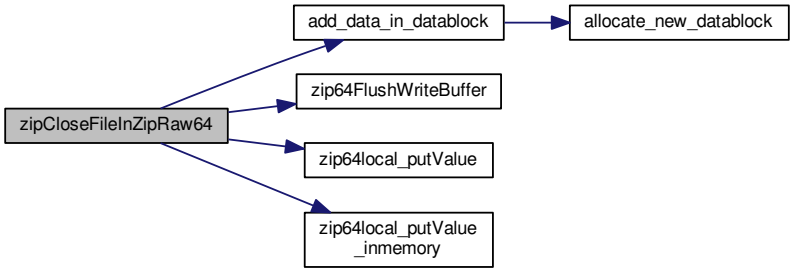
```
1555 {
1556     zip64_internal* zi;
1557     ZPOS64_T compressed_size;
1558     uLong invalidValue = 0xffffffff;
1559     short datasize = 0;
1560     int err=ZIP_OK;
1561
1562     if (file == NULL)
1563         return ZIP_PARAMERROR;
1564     zi = (zip64_internal*)file;
1565
1566     if (zi->in_opened_file_inzip == 0)
1567         return ZIP_PARAMERROR;
1568     zi->ci.stream.avail_in = 0;
1569
1570     if ((zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
1571                 {
1572                         while (err==ZIP_OK)
1573                         {
1574                                 uLong uTotalOutBefore;
1575                                 if (zi->ci.stream.avail_out == 0)
1576                                 {
1577                                         if (zip64FlushWriteBuffer(zi) ==
     ZIP_ERRNO)
1578                                                 err = ZIP_ERRNO;
1579                                         zi->ci.stream.avail_out = (uInt)
     Z_BUFSIZE;
1580                                         zi->ci.stream.next_out = zi->ci.
     buffered_data;
1581                                 }
1582                                 uTotalOutBefore = zi->ci.stream.total_out;
1583                                 err=deflate(&zi->ci.stream,  Z_FINISH);
1584                                 zi->ci.pos_in_buffered_data += (uInt)(zi->
     ci.stream.total_out - uTotalOutBefore) ;
1585                         }
1586                 }
1587     else if ((zi->ci.method == Z_BZIP2ED) && (!zi->ci.raw))
1588     {
1589 #ifdef HAVE_BZIP2
1590       err = BZ_FINISH_OK;
1591       while (err==BZ_FINISH_OK)
1592       {
1593         uLong uTotalOutBefore;
1594         if (zi->ci.bstream.avail_out == 0)
1595         {
1596           if (zip64FlushWriteBuffer(zi) == ZIP_ERRNO)
1597             err = ZIP_ERRNO;
1598           zi->ci.bstream.avail_out = (uInt)Z_BUFSIZE;
1599           zi->ci.bstream.next_out = (char*)zi->ci.buffered_data;
1600         }
1601         uTotalOutBefore = zi->ci.bstream.total_out_lo32;
1602         err=BZ2_bzCompress(&zi->ci.bstream,  BZ_FINISH);
1603         if(err == BZ_STREAM_END)
1604           err = Z_STREAM_END;
1605
1606         zi->ci.pos_in_buffered_data += (uInt)(zi->ci.bstream.total_out_lo32 -
     uTotalOutBefore);
1607       }
1608
1609       if(err == BZ_FINISH_OK)
1610         err = ZIP_OK;
1611 #endif
1612     }
1613
```

```
1614     if (err==Z_STREAM_END)
1615         err=ZIP_OK; /* this is normal */
1616
1617     if ((zi->ci.pos_in_buffered_data>0) && (err==ZIP_OK))
1618                 {
1619         if (zip64FlushWriteBuffer(zi)==ZIP_ERRNO)
1620             err = ZIP_ERRNO;
1621                 }
1622
1623     if ((zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
1624     {
1625         int tmp_err = deflateEnd(&zi->ci.stream);
1626         if (err == ZIP_OK)
1627             err = tmp_err;
1628         zi->ci.stream_initialised = 0;
1629     }
1630 #ifdef HAVE_BZIP2
1631     else if((zi->ci.method == Z_BZIP2ED) && (!zi->ci.raw))
1632     {
1633       int tmperr = BZ2_bzCompressEnd(&zi->ci.bstream);
1634                         if (err==ZIP_OK)
1635                                 err = tmperr;
1636                         zi->ci.stream_initialised = 0;
1637     }
1638 #endif
1639
1640     if (!zi->ci.raw)
1641     {
1642         crc32 = (uLong)zi->ci.crc32;
1643         uncompressed_size = zi->ci.totalUncompressedData;
1644     }
1645     compressed_size = zi->ci.totalCompressedData;
1646
1647 #    ifndef NOCRYPT
1648     compressed_size += zi->ci.crypt_header_size;
1649 #    endif
1650
1651     /* update Current Item crc and sizes, */
1652     if(compressed_size >= 0xffffffff || uncompressed_size >= 0xffffffff || zi->
     ci.pos_local_header >= 0xffffffff)
1653     {
1654       /*version Made by*/
1655       zip64local_putValue_inmemory(zi->ci.
     central_header+4,(uLong)45,2);
1656       /*version needed*/
1657       zip64local_putValue_inmemory(zi->ci.
     central_header+6,(uLong)45,2);
1658
1659     }
1660
1661     zip64local_putValue_inmemory(zi->ci.
     central_header+16,crc32,4); /*crc*/
1662
1663
1664     if(compressed_size >= 0xffffffff)
1665       zip64local_putValue_inmemory(zi->ci.
     central_header+20, invalidValue,4); /*compr size*/
1666     else
1667       zip64local_putValue_inmemory(zi->ci.
     central_header+20, compressed_size,4); /*compr size*/
1668
1669     /* set internal file attributes field */
1670     if (zi->ci.stream.data_type == Z_ASCII)
1671         zip64local_putValue_inmemory(zi->ci.
     central_header+36,(uLong)Z_ASCII,2);
1672
1673     if(uncompressed_size >= 0xffffffff)
1674       zip64local_putValue_inmemory(zi->ci.
     central_header+24, invalidValue,4); /*uncompr size*/
1675     else
1676       zip64local_putValue_inmemory(zi->ci.
     central_header+24, uncompressed_size,4); /*uncompr size*/
1677
1678     /* Add ZIP64 extra info field for uncompressed size */
1679     if(uncompressed_size >= 0xffffffff)
1680       datasize += 8;
1681
1682     /* Add ZIP64 extra info field for compressed size */
1683     if(compressed_size >= 0xffffffff)
1684       datasize += 8;
1685
1686     /* Add ZIP64 extra info field for relative offset to local file header of current file */
1687     if(zi->ci.pos_local_header >= 0xffffffff)
1688       datasize += 8;
1689
1690     if(datasize > 0)
1691     {
```

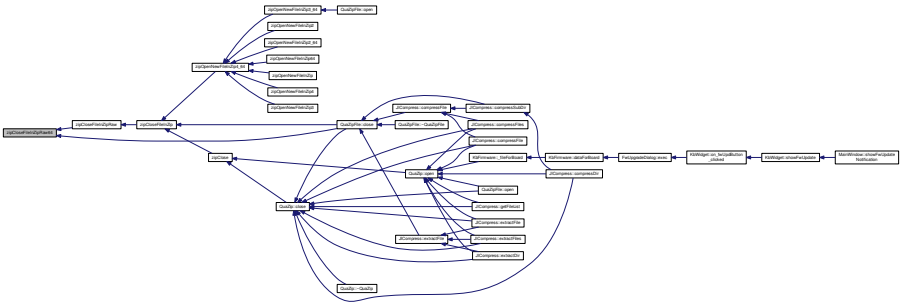
```
1692        char* p = NULL;
1693
1694        if((uLong)(datasize + 4) > zi->ci.size_centralExtraFree)
1695        {
1696          /* we can not write more data to the buffer that we have room for. */
1697          return ZIP_BADZIPFILE;
1698        }
1699
1700        p = zi->ci.central_header + zi->ci.size_centralheader;
1701
1702        /* Add Extra Information Header for 'ZIP64 information' */
1703        zip64local_putValue_inmemory(p, 0x0001, 2); /* HeaderID */
1704        p += 2;
1705        zip64local_putValue_inmemory(p, datasize, 2); /* DataSize */
1706        p += 2;
1707
1708        if(uncompressed_size >= 0xffffffff)
1709        {
1710          zip64local_putValue_inmemory(p, uncompressed_size, 8);
1711          p += 8;
1712        }
1713
1714        if(compressed_size >= 0xffffffff)
1715        {
1716          zip64local_putValue_inmemory(p, compressed_size, 8);
1717          p += 8;
1718        }
1719
1720        if(zi->ci.pos_local_header >= 0xffffffff)
1721        {
1722          zip64local_putValue_inmemory(p, zi->ci.
     pos_local_header, 8);
1723          p += 8;
1724        }
1725
1726        /* Update how much extra free space we got in the memory buffer */
1727        /* and increase the centralheader size so the new ZIP64 fields are included */
1728        /* ( 4 below is the size of HeaderID and DataSize field ) */
1729        zi->ci.size_centralExtraFree -= datasize + 4;
1730        zi->ci.size_centralheader += datasize + 4;
1731
1732        /* Update the extra info size field */
1733        zi->ci.size_centralExtra += datasize + 4;
1734        zip64local_putValue_inmemory(zi->ci.
     central_header+30,(uLong)zi->ci.size_centralExtra,2);
1735      }
1736
1737      if (err==ZIP_OK)
1738          err = add_data_in_datablock(&zi->central_dir, zi->
     ci.central_header, (uLong)zi->ci.size_centralheader);
1739
1740      free(zi->ci.central_header);
1741
1742      if (err==ZIP_OK)
1743      {
1744          if ((zi->flags & ZIP_SEQUENTIAL) == 0) {
1745              /* Update the LocalFileHeader with the new values. */
1746
1747              ZPOS64_T cur_pos_inzip = ZTELL64(zi->z_filefunc,zi->
     filestream);
1748
1749              if (ZSEEK64(zi->z_filefunc,zi->filestream, zi->
     ci.pos_local_header + 14,ZLIB_FILEFUNC_SEEK_SET)!=0)
1750                  err = ZIP_ERRNO;
1751
1752              if (err==ZIP_OK)
1753                  err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,crc32,4); /* crc 32, unknown */
1754
1755              if(uncompressed_size >= 0xffffffff || compressed_size >= 0xffffffff)
1756              {
1757                  if(zi->ci.pos_zip64extrainfo > 0)
1758                  {
1759                      /* Update the size in the ZIP64 extended field. */
1760                      if (ZSEEK64(zi->z_filefunc,zi->filestream, zi->
     ci.pos_zip64extrainfo + 4,ZLIB_FILEFUNC_SEEK_SET)!=0)
1761                          err = ZIP_ERRNO;
1762
1763                      if (err==ZIP_OK) /* compressed size, unknown */
1764                          err = zip64local_putValue(&zi->
     z_filefunc, zi->filestream, uncompressed_size, 8);
1765
1766                      if (err==ZIP_OK) /* uncompressed size, unknown */
1767                          err = zip64local_putValue(&zi->
     z_filefunc, zi->filestream, compressed_size, 8);
1768                  }
1769              }
```

```
1770            else
1771            {
1772                if (err==ZIP_OK) /* compressed size, unknown */
1773                    err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,compressed_size,4);
1774
1775                if (err==ZIP_OK) /* uncompressed size, unknown */
1776                    err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,uncompressed_size,4);
1777            }
1778
1779            if (ZSEEK64(zi->z_filefunc,zi->filestream, cur_pos_inzip,
     ZLIB_FILEFUNC_SEEK_SET)!=0)
1780                err = ZIP_ERRNO;
1781        }
1782
1783        if ((zi->ci.flag & 8) != 0) {
1784            /* Write local Descriptor after file data */
1785            if (err==ZIP_OK)
1786                err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,(uLong)DESCRIPTORHEADERMAGIC,4);
1787            if (err==ZIP_OK)
1788                err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,crc32,4); /* crc 32, unknown */
1789            if (zi->ci.zip64) {
1790                if (err==ZIP_OK) /* compressed size, unknown */
1791                    err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,compressed_size,8);
1792
1793                if (err==ZIP_OK) /* uncompressed size, unknown */
1794                    err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,uncompressed_size,8);
1795            } else {
1796                if (err==ZIP_OK) /* compressed size, unknown */
1797                    err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,compressed_size,4);
1798                if (err==ZIP_OK) /* uncompressed size, unknown */
1799                    err = zip64local_putValue(&zi->z_filefunc,zi->
     filestream,uncompressed_size,4);
1800            }
1801        }
1802    }
1803
1804    zi->number_entry ++;
1805    zi->in_opened_file_inzip = 0;
1806
1807    return err;
1808 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



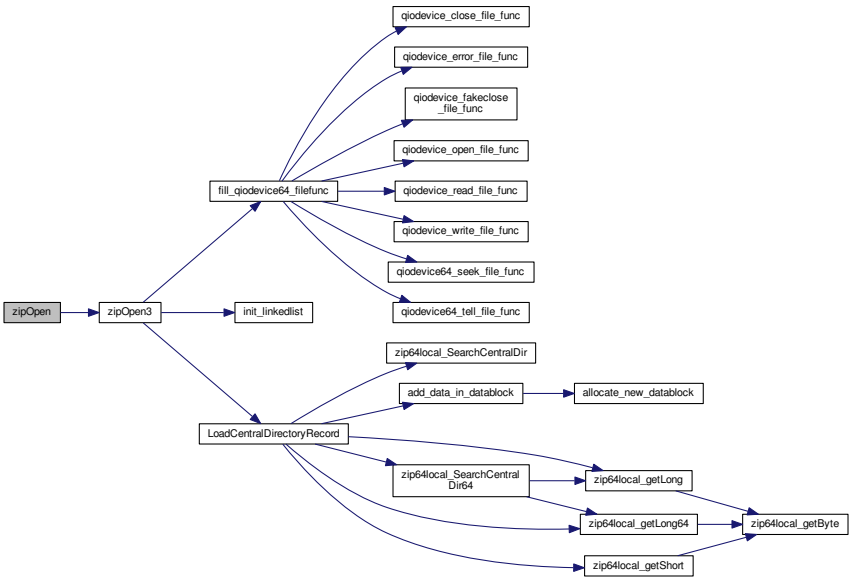
13.162.4.33 zipFile ZEXPORT zipOpen (voidpf *file*, int *append*)

Definition at line 967 of file zip.c.

References ZIP_DEFAULT_FLAGS, and zipOpen3().

```
968 {
969     return zipOpen3(file,append,NULL,NULL, ZIP_DEFAULT_FLAGS);
970 }
```

Here is the call graph for this function:



13.162.4.34 zipFile ZEXPORT zipOpen2 (voidpf *file*, int *append*, zipcharpc * *globalcomment*, zlib_filefunc_def * *pzlib_filefunc32_def*)

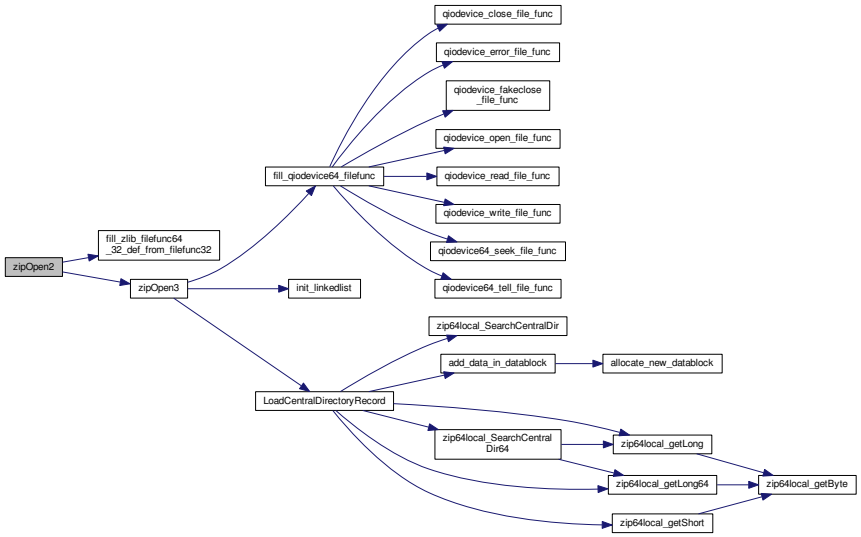
Definition at line 939 of file zip.c.

References fill_zlib_filefunc64_32_def_from_filefunc32(), ZIP_DEFAULT_FLAGS, and zipOpen3().

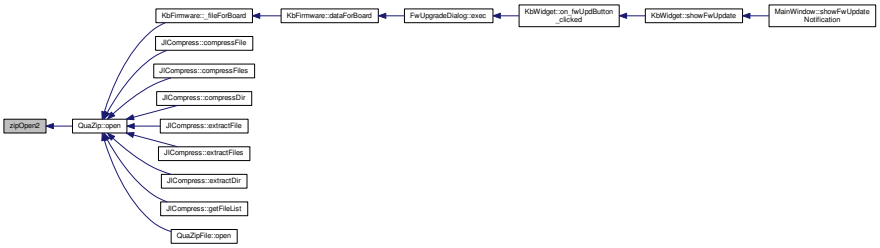
Referenced by QuaZip::open().

```
940 {
941     if (pzlib_filefunc32_def != NULL)
942     {
943         zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
944         fill_zlib_filefunc64_32_def_from_filefunc32(&
    zlib_filefunc64_32_def_fill,pzlib_filefunc32_def);
945         return zipOpen3(file, append, globalcomment, &zlib_filefunc64_32_def_fill,
    ZIP_DEFAULT_FLAGS);
946     }
947     else
948         return zipOpen3(file, append, globalcomment, NULL,
    ZIP_DEFAULT_FLAGS);
949 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.162.4.35 zipFile ZEXPORT zipOpen2_64 (voidpf *file*, int *append*, zipcharpc * *globalcomment*, zlib_filefunc64_def * *pzlib_filefunc_def*)

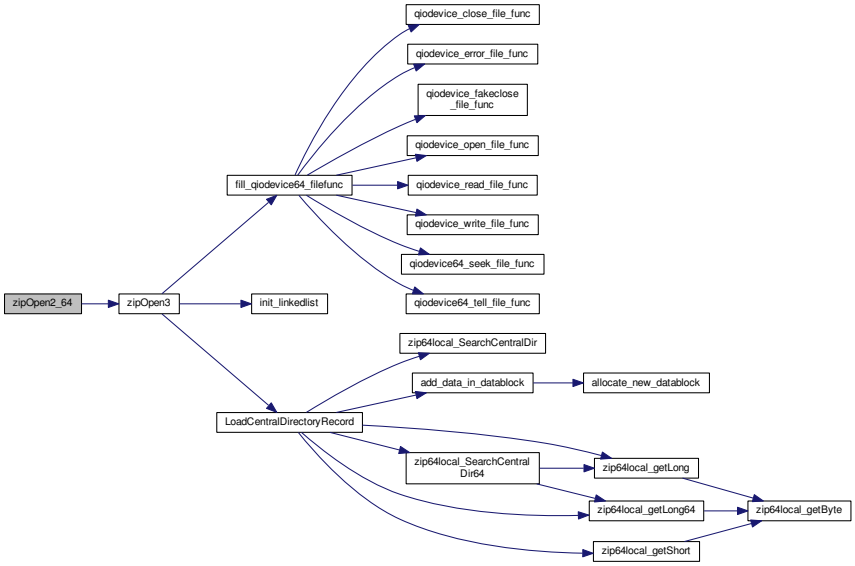
Definition at line 951 of file zip.c.

References zlib_filefunc64_32_def_s::zfile_func64, ZIP_DEFAULT_FLAGS, zipOpen3(), zlib_filefunc64_32_def_s::zseek32_file, and zlib_filefunc64_32_def_s::ztell32_file.

```
952 {
953     if (pzlib_filefunc_def != NULL)
954     {
```

```
955         zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
956         zlib_filefunc64_32_def_fill.zfile_func64 = *pzlib_filefunc_def;
957         zlib_filefunc64_32_def_fill.ztell32_file = NULL;
958         zlib_filefunc64_32_def_fill.zseek32_file = NULL;
959         return zipOpen3(file, append, globalcomment, &zlib_filefunc64_32_def_fill,
    ZIP_DEFAULT_FLAGS);
960     }
961     else
962         return zipOpen3(file, append, globalcomment, NULL,
    ZIP_DEFAULT_FLAGS);
963 }
```

Here is the call graph for this function:



13.162.4.36 zipFile ZEXPORT zipOpen3 (voidpf *file*, int *append*, zipcharpc * *globalcomment*, zlib_filefunc64_32_def * *pzlib_filefunc64_32_def*, unsigned *flags*)

Definition at line 862 of file zip.c.

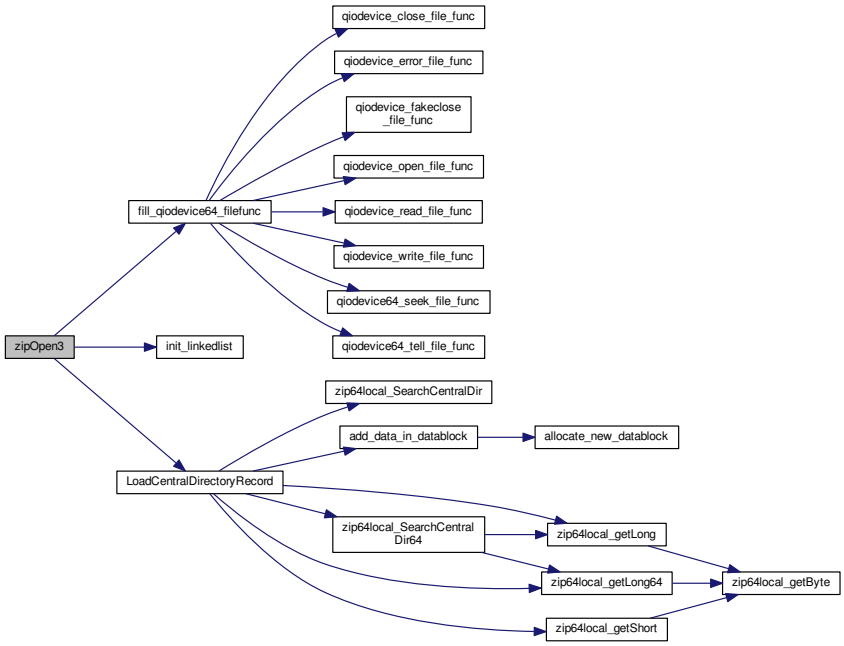
References zip64_internal::add_position_when_writting_offset, ALLOC, APPEND_STATUS_ADDINZIP, APPEND_STATUS_CREATE, APPEND_STATUS_CREATEAFTER, zip64_internal::begin_pos, zip64_internal::central_dir, zip64_internal::ci, zip64_internal::filestream, fill_qiodevice64_filefunc(), zip64_internal::flags, zip64_internal::globalcomment, zip64_internal::in_opened_file_inzip, init_linkedlist(), LoadCentralDirectoryRecord(), zip64_internal::number_entry, SEEK_END, curfile64_info::stream_initialised, TRYFREE, zip64_internal::z_filefunc, ZCLOSE64, ZFAKECLOSE64, zlib_filefunc64_32_def_s::zfile_func64, ZIP_AUTO_CLOSE, ZIP_OK, ZLIB_FILEFUNC_MODE_CREATE, ZLIB_FILEFUNC_MODE_EXISTING, ZLIB_FILEFUNC_MODE_READ, ZLIB_FILEFUNC_MODE_WRITE, ZOPEN64, zlib_filefunc64_32_def_s::zseek32_file, ZSEEK64, zlib_filefunc64_32_def_s::ztell32_file, and ZTELL64.

Referenced by QuaZip::open(), zipOpen(), zipOpen2(), zipOpen2_64(), and zipOpen64().

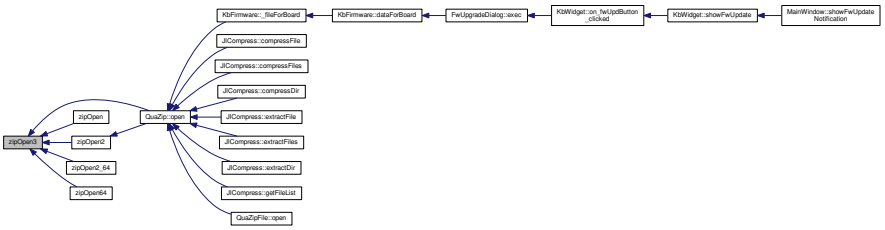
```
864 {
865     zip64_internal ziinit;
866     zip64_internal* zi;
867     int err=ZIP_OK;
868
869     ziinit.flags = flags;
870     ziinit.z_filefunc.zseek32_file = NULL;
871     ziinit.z_filefunc.ztell32_file = NULL;
872     if (pzlib_filefunc64_32_def==NULL)
873         fill_qiodevice64_filefunc(&ziinit.z_filefunc.
    zfile_func64);
```

```
874     else
875         ziinit.z_filefunc = *pzlib_filefunc64_32_def;
876
877     ziinit.filestream = ZOPEN64(ziinit.z_filefunc,
878                   file,
879                   (append == APPEND_STATUS_CREATE) ?
880                   (ZLIB_FILEFUNC_MODE_READ |
    ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_CREATE) :
881                     (ZLIB_FILEFUNC_MODE_READ |
    ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_EXISTING))
    ;
882
883     if (ziinit.filestream == NULL)
884         return NULL;
885
886     if (append == APPEND_STATUS_CREATEAFTER)
887         ZSEEK64(ziinit.z_filefunc,ziinit.filestream,0,
    SEEK_END);
888
889     ziinit.begin_pos = ZTELL64(ziinit.z_filefunc,ziinit.
    filestream);
890     ziinit.in_opened_file_inzip = 0;
891     ziinit.ci.stream_initialised = 0;
892     ziinit.number_entry = 0;
893     ziinit.add_position_when_writting_offset = 0;
894     init_linkedlist(&(ziinit.central_dir));
895
896
897
898     zi = (zip64_internal*)ALLOC(sizeof(zip64_internal));
899     if (zi==NULL)
900     {
901         if ((ziinit.flags & ZIP_AUTO_CLOSE) != 0) {
902             ZCLOSE64(ziinit.z_filefunc,ziinit.filestream);
903         } else {
904             ZFAKECLOSE64(ziinit.z_filefunc,ziinit.
    filestream);
905         }
906         return NULL;
907     }
908
909     /* now we add file in a zipfile */
910 #    ifndef NO_ADDFILEINEXISTINGZIP
911     ziinit.globalcomment = NULL;
912     if (append == APPEND_STATUS_ADDINZIP)
913     {
914       /* Read and Cache Central Directory Records */
915       err = LoadCentralDirectoryRecord(&ziinit);
916     }
917
918     if (globalcomment)
919     {
920       *globalcomment = ziinit.globalcomment;
921     }
922 #    endif /* !NO_ADDFILEINEXISTINGZIP*/
923
924     if (err != ZIP_OK)
925     {
926 #    ifndef NO_ADDFILEINEXISTINGZIP
927         TRYFREE(ziinit.globalcomment);
928 #    endif /* !NO_ADDFILEINEXISTINGZIP*/
929         TRYFREE(zi);
930         return NULL;
931     }
932     else
933     {
934         *zi = ziinit;
935         return (zipFile)zi;
936     }
937 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



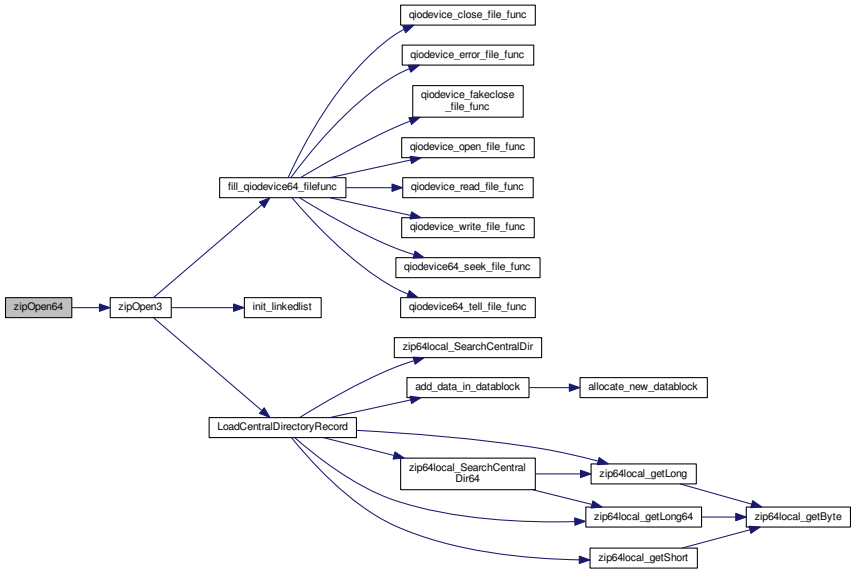
13.162.4.37 zipFile ZEXPORT zipOpen64 (voidpf *file,* int *append*)

Definition at line 972 of file zip.c.

References ZIP_DEFAULT_FLAGS, and zipOpen3().

```
973 {
974     return zipOpen3(file,append,NULL,NULL, ZIP_DEFAULT_FLAGS);
975 }
```

Here is the call graph for this function:



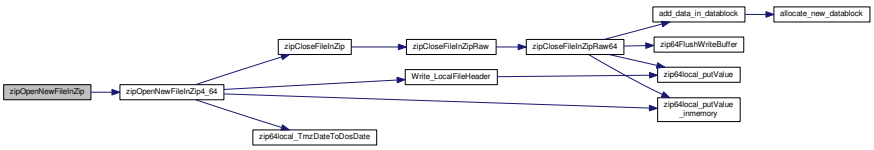
13.162.4.38 int ZEXPORT zipOpenNewFileInZip (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*)

Definition at line 1390 of file zip.c.

References DEF_MEM_LEVEL, VERSIONMADEBY, and zipOpenNewFileInZip4_64().

```
1394 {
1395     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1396                                     extrafield_local, size_extrafield_local,
1397                                     extrafield_global, size_extrafield_global,
1398                                     comment, method, level, 0,
1399                                     -MAX_WBITS, DEF_MEM_LEVEL, Z_DEFAULT_STRATEGY,
1400                                     NULL, 0, VERSIONMADEBY, 0, 0);
1401 }
```

Here is the call graph for this function:



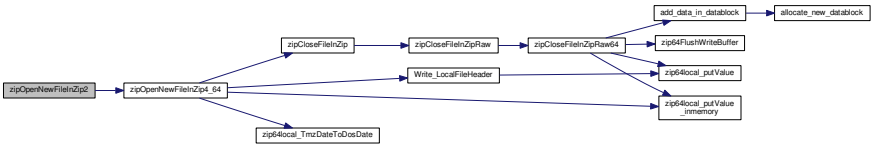
13.162.4.39 int ZEXPORT zipOpenNewFileInZip2 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *raw*)

Definition at line 1351 of file zip.c.

References DEF_MEM_LEVEL, VERSIONMADEBY, and zipOpenNewFileInZip4_64().

```
1355 {
1356     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1357                                     extrafield_local, size_extrafield_local,
1358                                     extrafield_global, size_extrafield_global,
1359                                     comment, method, level, raw,
1360                                     -MAX_WBITS, DEF_MEM_LEVEL, Z_DEFAULT_STRATEGY,
1361                                     NULL, 0, VERSIONMADEBY, 0, 0);
1362 }
```

Here is the call graph for this function:



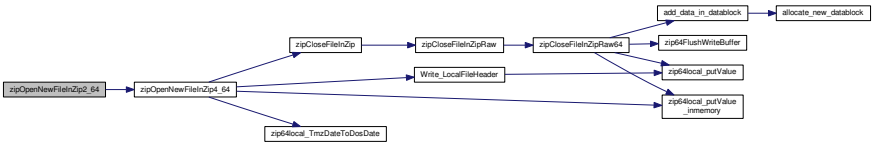
13.162.4.40 int ZEXPORT zipOpenNewFileInZip2_64 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *raw*, int *zip64*)

Definition at line 1364 of file zip.c.

References DEF_MEM_LEVEL, VERSIONMADEBY, and zipOpenNewFileInZip4_64().

```
1368 {
1369     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1370                                     extrafield_local, size_extrafield_local,
1371                                     extrafield_global, size_extrafield_global,
1372                                     comment, method, level, raw,
1373                                     -MAX_WBITS, DEF_MEM_LEVEL, Z_DEFAULT_STRATEGY,
1374                                     NULL, 0, VERSIONMADEBY, 0, zip64);
1375 }
```

Here is the call graph for this function:



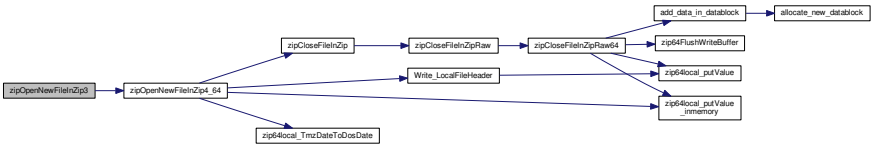
13.162.4.41 int ZEXPORT zipOpenNewFileInZip3 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *raw*, int *windowBits*, int *memLevel*, int *strategy*, const char * *password*, uLong *crcForCrypting*)

Definition at line 1321 of file zip.c.

References VERSIONMADEBY, and zipOpenNewFileInZip4_64().

```
1327 {
1328     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1329                                     extrafield_local, size_extrafield_local,
1330                                     extrafield_global, size_extrafield_global,
1331                                     comment, method, level, raw,
1332                                     windowBits, memLevel, strategy,
1333                                     password, crcForCrypting, VERSIONMADEBY, 0, 0);
1334 }
```

Here is the call graph for this function:



13.162.4.42 int ZEXPORT zipOpenNewFileInZip3_64 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *raw*, int *windowBits*, int *memLevel*, int *strategy*, const char * *password*, uLong *crcForCrypting*, int *zip64*)

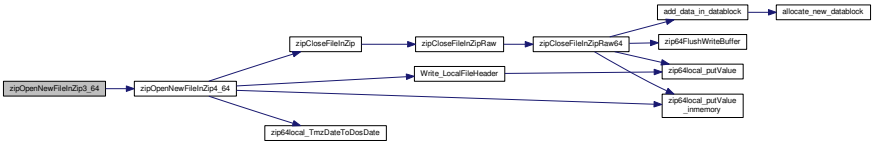
Definition at line 1336 of file zip.c.

References VERSIONMADEBY, and zipOpenNewFileInZip4_64().

Referenced by QuaZipFile::open().

```
1342 {
1343     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1344                                     extrafield_local, size_extrafield_local,
1345                                     extrafield_global, size_extrafield_global,
1346                                     comment, method, level, raw,
1347                                     windowBits, memLevel, strategy,
1348                                     password, crcForCrypting, VERSIONMADEBY, 0, zip64);
1349 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

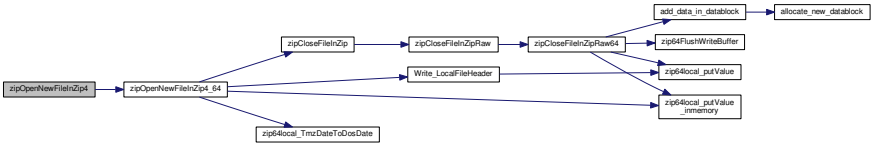
13.162.4.43 int ZEXPORT zipOpenNewFileInZip4 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *raw*, int *windowBits*, int *memLevel*, int *strategy*, const char * *password*, uLong *crcForCrypting*, uLong *versionMadeBy*, uLong *flagBase*)

Definition at line 1305 of file zip.c.

References zipOpenNewFileInZip4_64().

```
1312 {
1313     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1314                                     extrafield_local, size_extrafield_local,
1315                                     extrafield_global, size_extrafield_global,
1316                                     comment, method, level, raw,
1317                                     windowBits, memLevel, strategy,
1318                                     password, crcForCrypting, versionMadeBy, flagBase, 0);
1319 }
```

Here is the call graph for this function:



13.162.4.44 int ZEXPORT zipOpenNewFileInZip4_64 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *raw*, int *windowBits*, int *memLevel*, int *strategy*, const char * *password*, uLong *crcForCrypting*, uLong *versionMadeBy*, uLong *flagBase*, int *zip64*)

Definition at line 1077 of file zip.c.

References zip64_internal::add_position_when_writting_offset, ALLOC, curfile64_info::buffered_data, curfile64_info::central_header, CENTRALHEADERMAGIC, zip64_internal::ci, curfile64_info::crc32, curfile64_info::crypt_header_size, zip_fileinfo::dosDate, curfile64_info::dosDate, curfile64_info::encrypt, zip_fileinfo::external_fa, zip64_internal::filestream, curfile64_info::flag, zip64_internal::flags, zip64_internal::in_opened_file_inzip, zip_fileinfo::internal_fa, curfile64_info::keys, curfile64_info::method, curfile64_info::pcrc_32_tab, curfile64_info::pos_in_buffered_data, curfile64_info::pos_local_header, curfile64_info::pos_zip64extrainfo, curfile64_info::raw, curfile64_info::size_centralExtra, curfile64_info::size_centralExtraFree, curfile64_info::size_centralheader, SIZECENTRALHEADER, curfile64_info::stream, curfile64_info::stream_initialised, zip_fileinfo::tmz_date, curfile64_info::totalCompressedData, curfile64_info::totalUncompressedData, Write_LocalFileHeader(), Z_BUFSIZE, Z_BZIP2ED, zip64_internal::z_filefunc, curfile64_info::zip64, zip64local_putValue_inmemory(), zip64local_TmzDateToDosDate(), ZIP_ERRNO, ZIP_INTERNALERROR, ZIP_OK, ZIP_PARAMERROR, ZIP_SEQUENTIAL, ZIP_WRITE_DATA_DESCRIPTOR, zipCloseFileInZip(), ZTELL64, and ZWRITE64.

Referenced by zipOpenNewFileInZip(), zipOpenNewFileInZip2(), zipOpenNewFileInZip2_64(), zipOpenNewFileInZip3(), zipOpenNewFileInZip3_64(), zipOpenNewFileInZip4(), and zipOpenNewFileInZip64().

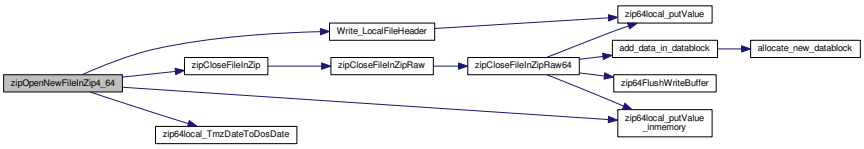
```
1084 {
1085     zip64_internal* zi;
1086     uInt size_filename;
1087     uInt size_comment;
1088     uInt i;
1089     int err = ZIP_OK;
1090     uLong version_to_extract;
1091
1092 #    ifdef NOCRYPT
1093     if (password != NULL)
1094         return ZIP_PARAMERROR;
1095 #    endif
1096
1097     if (file == NULL)
1098         return ZIP_PARAMERROR;
1099
1100 #ifdef HAVE_BZIP2
1101     if ((method!=0) && (method!=Z_DEFLATED) && (method!=Z_BZIP2ED))
1102       return ZIP_PARAMERROR;
1103 #else
1104     if ((method!=0) && (method!=Z_DEFLATED))
1105       return ZIP_PARAMERROR;
1106 #endif
```

```
1107
1108    zi = (zip64_internal*)file;
1109
1110    if (zi->in_opened_file_inzip == 1)
1111    {
1112        err = zipCloseFileInZip (file);
1113        if (err != ZIP_OK)
1114            return err;
1115    }
1116
1117    if (method == 0
1118            && (level == 0 || (zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) == 0)
1119            && (zi->flags & ZIP_SEQUENTIAL) == 0)
1120    {
1121        version_to_extract = 10;
1122    }
1123    else
1124    {
1125        version_to_extract = 20;
1126    }
1127
1128    if (filename==NULL)
1129        filename="-";
1130
1131    if (comment==NULL)
1132        size_comment = 0;
1133    else
1134        size_comment = (uInt)strlen(comment);
1135
1136    size_filename = (uInt)strlen(filename);
1137
1138    if (zipfi == NULL)
1139        zi->ci.dosDate = 0;
1140    else
1141    {
1142        if (zipfi->dosDate != 0)
1143            zi->ci.dosDate = zipfi->dosDate;
1144        else
1145          zi->ci.dosDate = zip64local_TmzDateToDosDate(&zipfi->
    tmz_date);
1146    }
1147
1148    zi->ci.flag = flagBase;
1149    if ((level==8) || (level==9))
1150      zi->ci.flag |= 2;
1151    if (level==2)
1152      zi->ci.flag |= 4;
1153    if (level==1)
1154      zi->ci.flag |= 6;
1155    if (password != NULL)
1156      zi->ci.flag |= 1;
1157    if (version_to_extract >= 20
1158            && ((zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) != 0
1159                || (zi->flags & ZIP_SEQUENTIAL) != 0))
1160      zi->ci.flag |= 8;
1161
1162    zi->ci.crc32 = 0;
1163    zi->ci.method = method;
1164    zi->ci.encrypt = 0;
1165    zi->ci.stream_initialised = 0;
1166    zi->ci.pos_in_buffered_data = 0;
1167    zi->ci.raw = raw;
1168    zi->ci.pos_local_header = ZTELL64(zi->z_filefunc,zi->
    filestream);
1169
1170    zi->ci.size_centralheader = SIZECENTRALHEADER + size_filename +
    size_extrafield_global + size_comment;
1171    zi->ci.size_centralExtraFree = 32; /* Extra space we have reserved in case we
     need to add ZIP64 extra info data */
1172
1173    zi->ci.central_header = (char*)ALLOC((uInt)zi->ci.
    size_centralheader + zi->ci.size_centralExtraFree);
1174
1175    zi->ci.size_centralExtra = size_extrafield_global;
1176    zip64local_putValue_inmemory(zi->ci.
    central_header,(uLong)CENTRALHEADERMAGIC,4);
1177    /* version info */
1178    zip64local_putValue_inmemory(zi->ci.
    central_header+4,(uLong)versionMadeBy,2);
1179    zip64local_putValue_inmemory(zi->ci.
    central_header+6,(uLong)version_to_extract,2);
1180    zip64local_putValue_inmemory(zi->ci.
    central_header+8,(uLong)zi->ci.flag,2);
1181    zip64local_putValue_inmemory(zi->ci.
    central_header+10,(uLong)zi->ci.method,2);
1182    zip64local_putValue_inmemory(zi->ci.
    central_header+12,(uLong)zi->ci.dosDate,4);
```

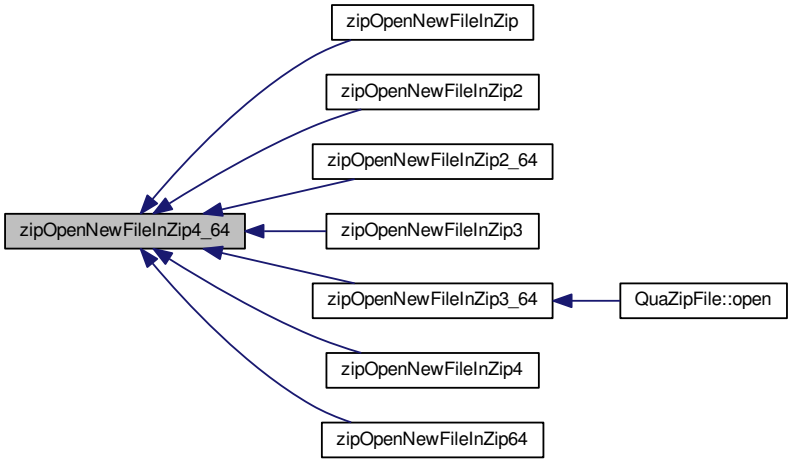
```
1183     zip64local_putValue_inmemory(zi->ci.
     central_header+16,(uLong)0,4); /*crc*/
1184     zip64local_putValue_inmemory(zi->ci.
     central_header+20,(uLong)0,4); /*compr size*/
1185     zip64local_putValue_inmemory(zi->ci.
     central_header+24,(uLong)0,4); /*uncompr size*/
1186     zip64local_putValue_inmemory(zi->ci.
     central_header+28,(uLong)size_filename,2);
1187     zip64local_putValue_inmemory(zi->ci.
     central_header+30,(uLong)size_extrafield_global,2);
1188     zip64local_putValue_inmemory(zi->ci.
     central_header+32,(uLong)size_comment,2);
1189     zip64local_putValue_inmemory(zi->ci.
     central_header+34,(uLong)0,2); /*disk nm start*/
1190
1191     if (zipfi==NULL)
1192         zip64local_putValue_inmemory(zi->ci.
     central_header+36,(uLong)0,2);
1193     else
1194         zip64local_putValue_inmemory(zi->ci.
     central_header+36,(uLong)zipfi->internal_fa,2);
1195
1196     if (zipfi==NULL)
1197         zip64local_putValue_inmemory(zi->ci.
     central_header+38,(uLong)0,4);
1198     else
1199         zip64local_putValue_inmemory(zi->ci.
     central_header+38,(uLong)zipfi->external_fa,4);
1200
1201     if(zi->ci.pos_local_header >= 0xffffffff)
1202       zip64local_putValue_inmemory(zi->ci.
     central_header+42,(uLong)0xffffffff,4);
1203     else
1204       zip64local_putValue_inmemory(zi->ci.
     central_header+42,(uLong)zi->ci.pos_local_header - zi->
     add_position_when_writting_offset,4);
1205
1206     for (i=0;i<size_filename;i++)
1207         *(zi->ci.central_header+SIZECENTRALHEADER+i) = *(filename+i);
1208
1209     for (i=0;i<size_extrafield_global;i++)
1210         *(zi->ci.central_header+SIZECENTRALHEADER+size_filename+i) =
1211               *(((const char*)extrafield_global)+i);
1212
1213     for (i=0;i<size_comment;i++)
1214         *(zi->ci.central_header+SIZECENTRALHEADER+size_filename+
1215               size_extrafield_global+i) = *(comment+i);
1216     if (zi->ci.central_header == NULL)
1217         return ZIP_INTERNALERROR;
1218
1219     zi->ci.zip64 = zip64;
1220     zi->ci.totalCompressedData = 0;
1221     zi->ci.totalUncompressedData = 0;
1222     zi->ci.pos_zip64extrainfo = 0;
1223
1224     err = Write_LocalFileHeader(zi, filename, size_extrafield_local,
1225                                 extrafield_local, version_to_extract);
1226
1227 #ifdef HAVE_BZIP2
1228     zi->ci.bstream.avail_in = (uInt)0;
1229     zi->ci.bstream.avail_out = (uInt)Z_BUFSIZE;
1230     zi->ci.bstream.next_out = (char*)zi->ci.buffered_data;
1231     zi->ci.bstream.total_in_hi32 = 0;
1232     zi->ci.bstream.total_in_lo32 = 0;
1233     zi->ci.bstream.total_out_hi32 = 0;
1234     zi->ci.bstream.total_out_lo32 = 0;
1235 #endif
1236
1237     zi->ci.stream.avail_in = (uInt)0;
1238     zi->ci.stream.avail_out = (uInt)Z_BUFSIZE;
1239     zi->ci.stream.next_out = zi->ci.buffered_data;
1240     zi->ci.stream.total_in = 0;
1241     zi->ci.stream.total_out = 0;
1242     zi->ci.stream.data_type = Z_BINARY;
1243
1244 #ifdef HAVE_BZIP2
1245     if ((err==ZIP_OK) && (zi->ci.method == Z_DEFLATED || zi->ci.
     method == Z_BZIP2ED) && (!zi->ci.raw))
1246 #else
1247     if ((err==ZIP_OK) && (zi->ci.method == Z_DEFLATED) && (!zi->ci.
     raw))
1248 #endif
1249     {
1250         if(zi->ci.method == Z_DEFLATED)
1251         {
1252           zi->ci.stream.zalloc = (alloc_func)0;
1253           zi->ci.stream.zfree = (free_func)0;
```

```
1254           zi->ci.stream.opaque = (voidpf)0;
1255
1256           if (windowBits>0)
1257               windowBits = -windowBits;
1258
1259           err = deflateInit2(&zi->ci.stream, level, Z_DEFLATED, windowBits, memLevel, strategy);
1260
1261           if (err==Z_OK)
1262               zi->ci.stream_initialised = Z_DEFLATED;
1263         }
1264         else if(zi->ci.method == Z_BZIP2ED)
1265         {
1266 #ifdef HAVE_BZIP2
1267             /* Init BZip stuff here */
1268           zi->ci.bstream.bzalloc = 0;
1269           zi->ci.bstream.bzfree = 0;
1270           zi->ci.bstream.opaque = (voidpf)0;
1271
1272           err = BZ2_bzCompressInit(&zi->ci.bstream, level, 0,35);
1273           if(err == BZ_OK)
1274             zi->ci.stream_initialised = Z_BZIP2ED;
1275 #endif
1276         }
1277
1278     }
1279
1280 #    ifndef NOCRYPT
1281     zi->ci.crypt_header_size = 0;
1282     if ((err==Z_OK) && (password != NULL))
1283     {
1284         unsigned char bufHead[RAND_HEAD_LEN];
1285         unsigned int sizeHead;
1286         zi->ci.encrypt = 1;
1287         zi->ci.pcrc_32_tab = get_crc_table();
1288         /*init_keys(password,zi->ci.keys,zi->ci.pcrc_32_tab);*/
1289         if (crcForCrypting == 0) {
1290             crcForCrypting = (uLong)zi->ci.dosDate << 16; /* ATTANTION! Without this row, you
     don't unpack your password protected archive in other app. */
1291         }
1292         sizeHead=crypthead(password,bufHead,RAND_HEAD_LEN,zi->ci.keys,zi->
      ci.pcrc_32_tab,crcForCrypting);
1293         zi->ci.crypt_header_size = sizeHead;
1294
1295         if (ZWRITE64(zi->z_filefunc,zi->filestream,bufHead,sizeHead) !=
     sizeHead)
1296                 err = ZIP_ERRNO;
1297     }
1298 #    endif
1299
1300     if (err==Z_OK)
1301         zi->in_opened_file_inzip = 1;
1302     return err;
1303 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



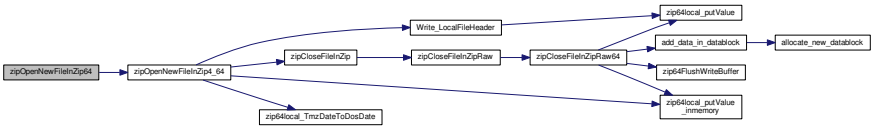
13.162.4.45 int ZEXPORT zipOpenNewFileInZip64 (zipFile *file*, const char * *filename*, const zip_fileinfo * *zipfi*, const void * *extrafield_local*, uInt *size_extrafield_local*, const void * *extrafield_global*, uInt *size_extrafield_global*, const char * *comment*, int *method*, int *level*, int *zip64*)

Definition at line 1377 of file zip.c.

References DEF_MEM_LEVEL, VERSIONMADEBY, and zipOpenNewFileInZip4_64().

```
1381 {
1382     return zipOpenNewFileInZip4_64 (file, filename, zipfi,
1383                                     extrafield_local, size_extrafield_local,
1384                                     extrafield_global, size_extrafield_global,
1385                                     comment, method, level, 0,
1386                                     -MAX_WBITS, DEF_MEM_LEVEL, Z_DEFAULT_STRATEGY,
1387                                     NULL, 0, VERSIONMADEBY, 0, zip64);
1388 }
```

Here is the call graph for this function:



13.162.4.46 int ZEXPORT zipRemoveExtraInfoBlock (char * *pData*, int * *dataLen*, short *sHeader*)

Definition at line 2019 of file zip.c.

References ALLOC, TRYFREE, ZIP_ERRNO, ZIP_OK, and ZIP_PARAMERROR.

```
2020 {
2021   char* p = pData;
2022   int size = 0;
```

```
2023   char* pNewHeader;
2024   char* pTmp;
2025   short header;
2026   short dataSize;
2027
2028   int retVal = ZIP_OK;
2029
2030   if(pData == NULL || *dataLen < 4)
2031     return ZIP_PARAMERROR;
2032
2033   pNewHeader = (char*)ALLOC(*dataLen);
2034   pTmp = pNewHeader;
2035
2036   while(p < (pData + *dataLen))
2037   {
2038     header = *(short*)p;
2039     dataSize = *(((short*)p)+1);
2040
2041     if( header == sHeader ) /* Header found. */
2042     {
2043       p += dataSize + 4; /* skip it. do not copy to temp buffer */
2044     }
2045     else
2046     {
2047       /* Extra Info block should not be removed, So copy it to the temp buffer. */
2048       memcpy(pTmp, p, dataSize + 4);
2049       p += dataSize + 4;
2050       size += dataSize + 4;
2051     }
2052
2053   }
2054
2055   if(size < *dataLen)
2056   {
2057     /* clean old extra info block. */
2058     memset(pData,0, *dataLen);
2059
2060     /* copy the new extra info block over the old */
2061     if(size > 0)
2062       memcpy(pData, pNewHeader, size);
2063
2064     /* set the new extra info size */
2065     *dataLen = size;
2066
2067     retVal = ZIP_OK;
2068   }
2069   else
2070     retVal = ZIP_ERRNO;
2071
2072   TRYFREE(pNewHeader);
2073
2074   return retVal;
2075 }
```

13.162.4.47 int ZEXPORT zipSetFlags (zipFile *file*, unsigned *flags*)

Definition at line 2077 of file zip.c.

References zip64_internal::flags, ZIP_OK, ZIP_PARAMERROR, ZIP_SEQUENTIAL, and ZIP_WRITE_DATA_DESCRIPTOR.

Referenced by QuaZip::open(), and QuaZipFile::open().

```
2078 {
2079     zip64_internal* zi;
2080     if (file == NULL)
2081         return ZIP_PARAMERROR;
2082     zi = (zip64_internal*)file;
2083     zi->flags |= flags;
2084     // If the output is non-seekable, the data descriptor is needed.
2085     if ((zi->flags & ZIP_SEQUENTIAL) != 0) {
2086         zi->flags |= ZIP_WRITE_DATA_DESCRIPTOR;
2087     }
2088     return ZIP_OK;
2089 }
```

Here is the caller graph for this function:

13.162.4.48 int ZEXPORT zipWriteInFileInZip (zipFile *file*, const void * *buf*, unsigned int *len*)

Definition at line 1442 of file zip.c.

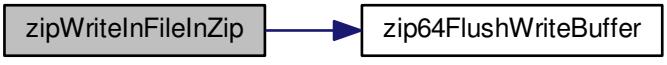
References curfile64_info::buffered_data, zip64_internal::ci, curfile64_info::crc32, zip64_internal::in_opened_file_inzip, curfile64_info::method, curfile64_info::pos_in_buffered_data, curfile64_info::raw, curfile64_info::stream, Z_BUFSIZE, Z_BZIP2ED, zip64FlushWriteBuffer(), ZIP_ERRNO, ZIP_OK, and ZIP_PARAMERROR.

Referenced by QuaZipFile::writeData().

```
1443 {
1444     zip64_internal* zi;
1445     int err=ZIP_OK;
1446
1447     if (file == NULL)
1448         return ZIP_PARAMERROR;
1449     zi = (zip64_internal*)file;
1450
1451     if (zi->in_opened_file_inzip == 0)
1452         return ZIP_PARAMERROR;
1453
1454     zi->ci.crc32 = crc32(zi->ci.crc32,buf,(uInt)len);
1455
1456 #ifdef HAVE_BZIP2
1457     if(zi->ci.method == Z_BZIP2ED && (!zi->ci.raw))
1458     {
1459       zi->ci.bstream.next_in = (void*)buf;
1460       zi->ci.bstream.avail_in = len;
1461       err = BZ_RUN_OK;
1462
1463       while ((err==BZ_RUN_OK) && (zi->ci.bstream.avail_in>0))
1464       {
1465         if (zi->ci.bstream.avail_out == 0)
1466         {
1467           if (zip64FlushWriteBuffer(zi) == ZIP_ERRNO)
1468             err = ZIP_ERRNO;
1469           zi->ci.bstream.avail_out = (uInt)Z_BUFSIZE;
1470           zi->ci.bstream.next_out = (char*)zi->ci.buffered_data;
1471         }
1472
1473
1474         if(err != BZ_RUN_OK)
1475           break;
1476
1477         if ((zi->ci.method == Z_BZIP2ED) && (!zi->ci.raw))
1478         {
1479           uLong uTotalOutBefore_lo = zi->ci.bstream.total_out_lo32;
1480 /*          uLong uTotalOutBefore_hi = zi->ci.bstream.total_out_hi32; */
1481           err=BZ2_bzCompress(&zi->ci.bstream,  BZ_RUN);
1482
1483           zi->ci.pos_in_buffered_data += (uInt)(zi->ci.bstream.total_out_lo32 -
      uTotalOutBefore_lo) ;
1484         }
1485       }
1486
1487       if(err == BZ_RUN_OK)
1488         err = ZIP_OK;
1489     }
1490     else
1491 #endif
1492     {
1493       zi->ci.stream.next_in = (Bytef*)buf;
```

```
1494        zi->ci.stream.avail_in = len;
1495
1496        while ((err==ZIP_OK) && (zi->ci.stream.avail_in>0))
1497        {
1498            if (zi->ci.stream.avail_out == 0)
1499            {
1500                if (zip64FlushWriteBuffer(zi) == ZIP_ERRNO)
1501                    err = ZIP_ERRNO;
1502                zi->ci.stream.avail_out = (uInt)Z_BUFSIZE;
1503                zi->ci.stream.next_out = zi->ci.buffered_data;
1504            }
1505
1506
1507            if(err != ZIP_OK)
1508                break;
1509
1510            if ((zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
1511            {
1512                uLong uTotalOutBefore = zi->ci.stream.total_out;
1513                err=deflate(&zi->ci.stream,  Z_NO_FLUSH);
1514                if(uTotalOutBefore > zi->ci.stream.total_out)
1515                {
1516                  int bBreak = 0;
1517                  bBreak++;
1518                }
1519
1520                zi->ci.pos_in_buffered_data += (uInt)(zi->ci.
    stream.total_out - uTotalOutBefore) ;
1521            }
1522            else
1523            {
1524                uInt copy_this,i;
1525                if (zi->ci.stream.avail_in < zi->ci.stream.avail_out)
1526                    copy_this = zi->ci.stream.avail_in;
1527                else
1528                    copy_this = zi->ci.stream.avail_out;
1529
1530                for (i = 0; i < copy_this; i++)
1531                    *(((char*)zi->ci.stream.next_out)+i) =
1532                        *(((const char*)zi->ci.stream.next_in)+i);
1533                {
1534                    zi->ci.stream.avail_in -= copy_this;
1535                    zi->ci.stream.avail_out-= copy_this;
1536                    zi->ci.stream.next_in+= copy_this;
1537                    zi->ci.stream.next_out+= copy_this;
1538                    zi->ci.stream.total_in+= copy_this;
1539                    zi->ci.stream.total_out+= copy_this;
1540                    zi->ci.pos_in_buffered_data += copy_this;
1541                }
1542            }
1543        }/* while(...) */
1544    }
1545
1546    return err;
1547 }
```

Here is the call graph for this function:



Here is the caller graph for this function:

13.162.5 Variable Documentation

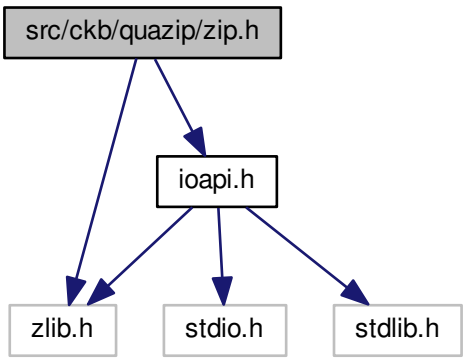
13.162.5.1 const char zip_copyright[ ] =" zip 1.01 Copyright 1998-2004 Gilles Vollant - http://www.winimage.com/zLibDll"

Definition at line 104 of file zip.c.

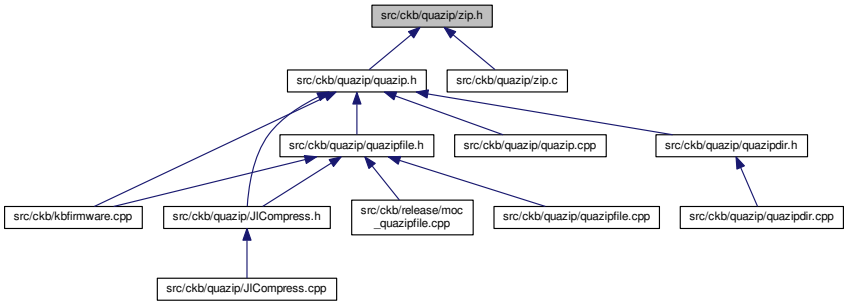
13.163 src/ckb/quazip/zip.h File Reference

```
#include "zlib.h"
#include "ioapi.h"
```

Include dependency graph for zip.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct tm_zip_s
- struct zip_fileinfo

Macros

- #define Z_BZIP2ED 12
- #define ZIP_OK (0)
- #define ZIP_EOF (0)
- #define ZIP_ERRNO (Z_ERRNO)
- #define ZIP_PARAMERROR (-102)
- #define ZIP_BADZIPFILE (-103)
- #define ZIP_INTERNALERROR (-104)
- #define ZIP_WRITE_DATA_DESCRIPTOR 0x8u
- #define ZIP_AUTO_CLOSE 0x1u
- #define ZIP_SEQUENTIAL 0x2u
- #define ZIP_DEFAULT_FLAGS (ZIP_AUTO_CLOSE | ZIP_WRITE_DATA_DESCRIPTOR)
- #define DEF_MEM_LEVEL MAX_MEM_LEVEL
- #define APPEND_STATUS_CREATE (0)
- #define APPEND_STATUS_CREATEAFTER (1)
- #define APPEND_STATUS_ADDINZIP (2)

Typedefs

- typedef voidp zipFile
- typedef struct tm_zip_s tm_zip
- typedef const char * zipcharpc

Functions

- zipFile ZEXPORT zipOpen _Z_OF ((voidpf file, int append))
- zipFile ZEXPORT zipOpen2 _Z_OF ((voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc_def *pzlib_filefunc_def))
- zipFile ZEXPORT zipOpen2_64 _Z_OF ((voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc64_def *pzlib_filefunc_def))
- zipFile ZEXPORT zipOpen3 (voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc64_32_def *pzlib_filefunc64_32_def, unsigned flags)

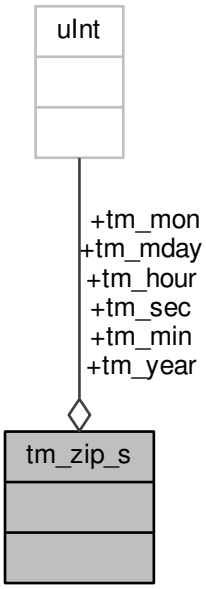
- int ZEXPORT zipOpenNewFileInZip _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level))
- int ZEXPORT zipOpenNewFileInZip64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int zip64))
- int ZEXPORT zipOpenNewFileInZip2 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw))
- int ZEXPORT zipOpenNewFileInZip2_64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int zip64))
- int ZEXPORT zipOpenNewFileInZip3 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting))
- int ZEXPORT zipOpenNewFileInZip3_64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting, int zip64))
- int ZEXPORT zipOpenNewFileInZip4 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting, uLong versionMadeBy, uLong flagBase))
- int ZEXPORT zipOpenNewFileInZip4_64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting, uLong versionMadeBy, uLong flagBase, int zip64))
- int ZEXPORT zipWriteInFileInZip _Z_OF ((zipFile file, const void ∗buf, unsigned len))
- int ZEXPORT zipCloseFileInZip _Z_OF ((zipFile file))
- int ZEXPORT zipCloseFileInZipRaw _Z_OF ((zipFile file, uLong uncompressed_size, uLong crc32))
- int ZEXPORT zipCloseFileInZipRaw64 _Z_OF ((zipFile file, ZPOS64_T uncompressed_size, uLong crc32))
- int ZEXPORT zipClose _Z_OF ((zipFile file, const char ∗global_comment))
- int ZEXPORT zipRemoveExtraInfoBlock _Z_OF ((char ∗pData, int ∗dataLen, short sHeader))
- int ZEXPORT zipSetFlags (zipFile file, unsigned flags)
- int ZEXPORT zipClearFlags (zipFile file, unsigned flags)

13.163.1 Data Structure Documentation

13.163.1.1 struct tm_zip_s

Definition at line 100 of file zip.h.

Collaboration diagram for tm_zip_s:



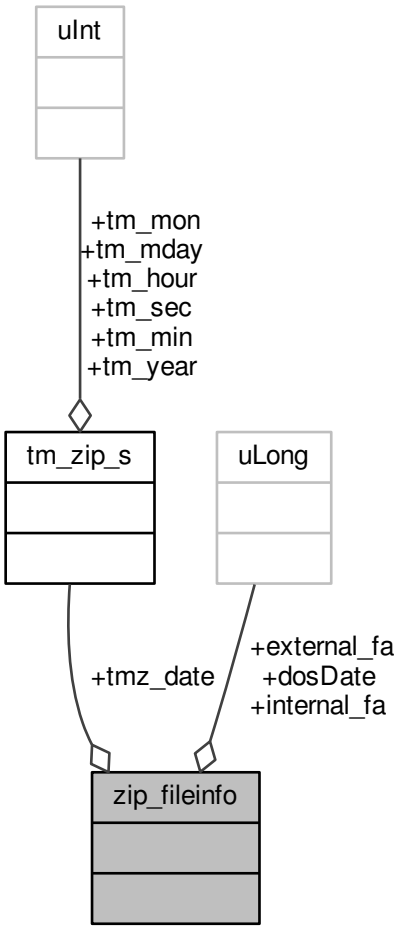
Data Fields

uInt	tm_hour	
uInt	tm_mday	
uInt	tm_min	
uInt	tm_mon	
uInt	tm_sec	
uInt	tm_year	

13.163.1.2 struct zip_fileinfo

Definition at line 110 of file zip.h.

Collaboration diagram for zip_fileinfo:



Data Fields

uLong	dosDate	
uLong	external_fa	
uLong	internal_fa	
tm_zip	tmz_date	

13.163.2 Macro Definition Documentation

13.163.2.1 #define APPEND_STATUS_ADDINZIP (2)

Definition at line 125 of file zip.h.

Referenced by QuaZip::open(), and zipOpen3().

13.163.2.2 #define APPEND_STATUS_CREATE (0)

Definition at line 123 of file zip.h.

Referenced by QuaZip::open(), and zipOpen3().

13.163.2.3 #define APPEND_STATUS_CREATEAFTER (1)

Definition at line 124 of file zip.h.

Referenced by QuaZip::open(), and zipOpen3().

13.163.2.4 #define DEF_MEM_LEVEL MAX_MEM_LEVEL

Definition at line 94 of file zip.h.

Referenced by zipOpenNewFileInZip(), zipOpenNewFileInZip2(), zipOpenNewFileInZip2_64(), and zipOpenNewFileInZip64().

13.163.2.5 #define Z_BZIP2ED 12

Definition at line 67 of file zip.h.

13.163.2.6 #define ZIP_AUTO_CLOSE 0x1u

Definition at line 86 of file zip.h.

Referenced by LoadCentralDirectoryRecord(), QuaZip::open(), zipClose(), and zipOpen3().

13.163.2.7 #define ZIP_BADZIPFILE (-103)

Definition at line 82 of file zip.h.

Referenced by LoadCentralDirectoryRecord(), and zipCloseFileInZipRaw64().

13.163.2.8 #define ZIP_DEFAULT_FLAGS (ZIP_AUTO_CLOSE | ZIP_WRITE_DATA_DESCRIPTOR)

Definition at line 88 of file zip.h.

Referenced by zipOpen(), zipOpen2(), zipOpen2_64(), and zipOpen64().

13.163.2.9 #define ZIP_EOF (0)

Definition at line 79 of file zip.h.

Referenced by zip64local_getByte().

13.163.2.10 #define ZIP_ERRNO (Z_ERRNO)

Definition at line 80 of file zip.h.

Referenced by LoadCentralDirectoryRecord(), Write_GlobalComment(), Write_LocalFileHeader(), zip64FlushWriteBuffer(), zip64local_getByte(), zip64local_putValue(), zipClose(), zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), zipRemoveExtraInfoBlock(), and zipWriteInFileInZip().

13.163.2.11 #define ZIP_INTERNALERROR (-104)

Definition at line 83 of file zip.h.

Referenced by add_data_in_datablock(), and zipOpenNewFileInZip4_64().

13.163.2.12 #define ZIP_OK (0)

Definition at line 78 of file zip.h.

Referenced by add_data_in_datablock(), LoadCentralDirectoryRecord(), Write_EndOfCentralDirectoryRecord(), Write_GlobalComment(), Write_LocalFileHeader(), Write_Zip64EndOfCentralDirectoryLocator(), Write_Zip64EndOfCentralDirectoryRecord(), QuaZipFile::writeData(), zip64FlushWriteBuffer(), zip64local_getByte(), zip64local_getLong(), zip64local_getLong64(), zip64local_getShort(), zip64local_putValue(), zip64local_SearchCentralDir64(), zipClearFlags(), zipClose(), zipCloseFileInZipRaw64(), zipOpen3(), zipOpenNewFileInZip4_64(), zipRemoveExtraInfoBlock(), zipSetFlags(), and zipWriteInFileInZip().

13.163.2.13 #define ZIP_PARAMERROR (-102)

Definition at line 81 of file zip.h.

Referenced by zipClearFlags(), zipClose(), zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), zipRemoveExtraInfoBlock(), zipSetFlags(), and zipWriteInFileInZip().

13.163.2.14 #define ZIP_SEQUENTIAL 0x2u

Definition at line 87 of file zip.h.

Referenced by QuaZip::open(), zipClearFlags(), zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), and zipSetFlags().

13.163.2.15 #define ZIP_WRITE_DATA_DESCRIPTOR 0x8u

Definition at line 85 of file zip.h.

Referenced by QuaZip::open(), QuaZipFile::open(), zipClearFlags(), zipOpenNewFileInZip4_64(), and zipSetFlags().

13.163.3 Typedef Documentation

13.163.3.1 typedef struct tm_zip_s tm_zip

13.163.3.2 typedef const char* zipcharpc

Definition at line 120 of file zip.h.

13.163.3.3 typedef voidp zipFile

Definition at line 75 of file zip.h.

13.163.4 Function Documentation

13.163.4.1 zipFile ZEXPORT zipOpen _Z_OF ((voidpf file, int append))

13.163.4.2 zipFile ZEXPORT zipOpen2 _Z_OF ((voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc_def *pzlib_filefunc_def))

13.163.4.3 zipFile ZEXPORT zipOpen2_64 _Z_OF ((voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc64_def *pzlib_filefunc_def))

13.163.4.4 int ZEXPORT zipOpenNewFileInZip _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level))

13.163.4.5 int ZEXPORT zipOpenNewFileInZip64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int zip64))

13.163.4.6 int ZEXPORT zipOpenNewFileInZip2 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw))

13.163.4.7 int ZEXPORT zipOpenNewFileInZip2_64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int zip64))

13.163.4.8 int ZEXPORT zipOpenNewFileInZip3 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting))

13.163.4.9 int ZEXPORT zipOpenNewFileInZip3_64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting, int zip64))

13.163.4.10 int ZEXPORT zipOpenNewFileInZip4 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting, uLong versionMadeBy, uLong flagBase))

13.163.4.11 int ZEXPORT zipOpenNewFileInZip4_64 _Z_OF ((zipFile file, const char ∗filename, const zip_fileinfo ∗zipfi, const void ∗extrafield_local, uInt size_extrafield_local, const void ∗extrafield_global, uInt size_extrafield_global, const char ∗comment, int method, int level, int raw, int windowBits, int memLevel, int strategy, const char ∗password, uLong crcForCrypting, uLong versionMadeBy, uLong flagBase, int zip64))

13.163.4.12 int ZEXPORT zipWriteInFileInZip _Z_OF ((zipFile file, const void ∗buf, unsigned len))

13.163.4.13 int ZEXPORT zipCloseFileInZip _Z_OF ((zipFile file))

13.163.4.14 int ZEXPORT zipCloseFileInZipRaw _Z_OF ((zipFile file, uLong uncompressed_size, uLong crc32))

13.163.4.15 int ZEXPORT zipCloseFileInZipRaw64 _Z_OF ((zipFile file, ZPOS64_T uncompressed_size, uLong crc32))

13.163.4.16 int ZEXPORT zipClose _Z_OF ((zipFile file, const char ∗global_comment))

13.163.4.17 int ZEXPORT zipRemoveExtraInfoBlock _Z_OF ((char ∗pData, int ∗dataLen, short sHeader))

13.163.4.18 int ZEXPORT zipClearFlags (zipFile *file,* unsigned *flags*)

Definition at line 2091 of file zip.c.

References zip64_internal::flags, ZIP_OK, ZIP_PARAMERROR, ZIP_SEQUENTIAL, and ZIP_WRITE_DATA_DESCRIPTOR.

Referenced by QuaZipFile::open().

```
2092 {
2093     zip64_internal* zi;
```

```
2094     if (file == NULL)
2095         return ZIP_PARAMERROR;
2096     zi = (zip64_internal*)file;
2097     zi->flags &= ~flags;
2098     // If the data descriptor is not written, we can't use a non-seekable output.
2099     if ((zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) == 0) {
2100         zi->flags &= ~ZIP_SEQUENTIAL;
2101     }
2102     return ZIP_OK;
2103 }
```

Here is the caller graph for this function:

13.163.4.19 zipFile ZEXPORT zipOpen3 (voidpf *file*, int *append*, zipcharpc * *globalcomment*, zlib_filefunc64_32_def * *pzlib_filefunc64_32_def*, unsigned *flags*)

Definition at line 862 of file zip.c.

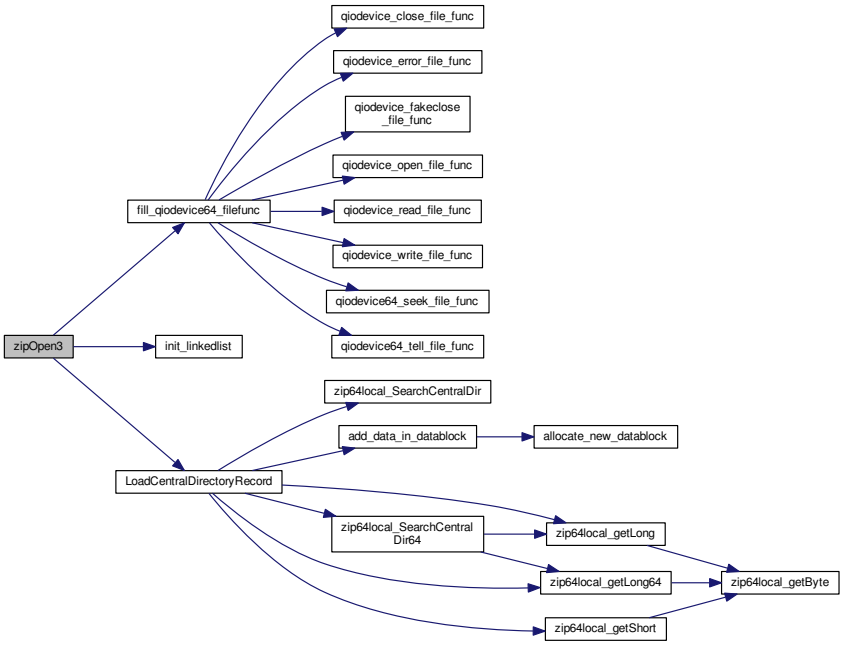
References zip64_internal::add_position_when_writting_offset, ALLOC, APPEND_STATUS_ADDINZIP, APPEND_STATUS_CREATE, APPEND_STATUS_CREATEAFTER, zip64_internal::begin_pos, zip64_internal::central_dir, zip64_internal::ci, zip64_internal::filestream, fill_qiodevice64_filefunc(), zip64_internal::flags, zip64_internal::globalcomment, zip64_internal::in_opened_file_inzip, init_linkedlist(), LoadCentralDirectoryRecord(), zip64_internal::number_entry, SEEK_END, curfile64_info::stream_initialised, TRYFREE, zip64_internal::z_filefunc, ZCLOSE64, ZFAKECLOSE64, zlib_filefunc64_32_def_s::zfile_func64, ZIP_AUTO_CLOSE, ZIP_OK, ZLIB_FILEFUNC_MODE_CREATE, ZLIB_FILEFUNC_MODE_EXISTING, ZLIB_FILEFUNC_MODE_READ, ZLIB_FILEFUNC_MODE_WRITE, ZOPEN64, zlib_filefunc64_32_def_s::zseek32_file, ZSEEK64, zlib_filefunc64_32_def_s::ztell32_file, and ZTELL64.

Referenced by QuaZip::open(), zipOpen(), zipOpen2(), zipOpen2_64(), and zipOpen64().

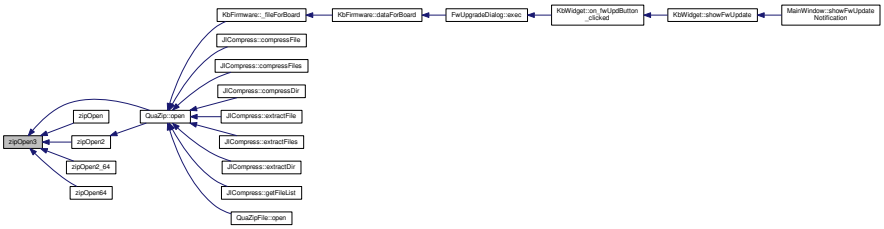
```
864 {
865     zip64_internal ziinit;
866     zip64_internal* zi;
867     int err=ZIP_OK;
868 
869     ziinit.flags = flags;
870     ziinit.z_filefunc.zseek32_file = NULL;
871     ziinit.z_filefunc.ztell32_file = NULL;
872     if (pzlib_filefunc64_32_def==NULL)
873         fill_qiodevice64_filefunc(&ziinit.z_filefunc.
      zfile_func64);
874     else
875         ziinit.z_filefunc = *pzlib_filefunc64_32_def;
876 
877     ziinit.filestream = ZOPEN64(ziinit.z_filefunc,
878                   file,
879                   (append == APPEND_STATUS_CREATE) ?
880                   (ZLIB_FILEFUNC_MODE_READ |
      ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_CREATE) :
881                     (ZLIB_FILEFUNC_MODE_READ |
      ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_EXISTING))
      ;
882 
883     if (ziinit.filestream == NULL)
884         return NULL;
885 
886     if (append == APPEND_STATUS_CREATEAFTER)
887         ZSEEK64(ziinit.z_filefunc,ziinit.filestream,0,
      SEEK_END);
888 
```

```
889     ziinit.begin_pos = ZTELL64(ziinit.z_filefunc,ziinit.
      filestream);
890     ziinit.in_opened_file_inzip = 0;
891     ziinit.ci.stream_initialised = 0;
892     ziinit.number_entry = 0;
893     ziinit.add_position_when_writting_offset = 0;
894     init_linkedlist(&(ziinit.central_dir));
895
896
897
898     zi = (zip64_internal*)ALLOC(sizeof(zip64_internal));
899     if (zi==NULL)
900     {
901         if ((ziinit.flags & ZIP_AUTO_CLOSE) != 0) {
902             ZCLOSE64(ziinit.z_filefunc,ziinit.filestream);
903         } else {
904             ZFAKECLOSE64(ziinit.z_filefunc,ziinit.
      filestream);
905         }
906         return NULL;
907     }
908
909     /* now we add file in a zipfile */
910 #    ifndef NO_ADDFILEINEXISTINGZIP
911     ziinit.globalcomment = NULL;
912     if (append == APPEND_STATUS_ADDINZIP)
913     {
914       /* Read and Cache Central Directory Records */
915       err = LoadCentralDirectoryRecord(&ziinit);
916     }
917
918     if (globalcomment)
919     {
920       *globalcomment = ziinit.globalcomment;
921     }
922 #    endif /* !NO_ADDFILEINEXISTINGZIP*/
923
924     if (err != ZIP_OK)
925     {
926 #    ifndef NO_ADDFILEINEXISTINGZIP
927         TRYFREE(ziinit.globalcomment);
928 #    endif /* !NO_ADDFILEINEXISTINGZIP*/
929         TRYFREE(zi);
930         return NULL;
931     }
932     else
933     {
934         *zi = ziinit;
935         return (zipFile)zi;
936     }
937 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



13.163.4.20 int ZEXPORT zipSetFlags (zipFile *file*, unsigned *flags*)

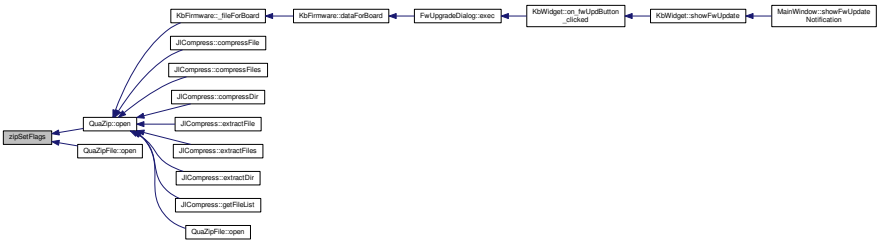
Definition at line 2077 of file zip.c.

References zip64_internal::flags, ZIP_OK, ZIP_PARAMERROR, ZIP_SEQUENTIAL, and ZIP_WRITE_DATA_DESCRIPTOR.

Referenced by QuaZip::open(), and QuaZipFile::open().

```
2078 {
2079     zip64_internal* zi;
2080     if (file == NULL)
2081         return ZIP_PARAMERROR;
2082     zi = (zip64_internal*)file;
2083     zi->flags |= flags;
2084     // If the output is non-seekable, the data descriptor is needed.
2085     if ((zi->flags & ZIP_SEQUENTIAL) != 0) {
2086         zi->flags |= ZIP_WRITE_DATA_DESCRIPTOR;
2087     }
2088     return ZIP_OK;
2089 }
```

Here is the caller graph for this function:



13.164 src/ckb/rebindwidget.cpp File Reference

```
#include <QStandardPaths>
#include "rebindwidget.h"
#include "ui_rebindwidget.h"
#include <qdebug.h>
```

Include dependency graph for rebindwidget.cpp:

Variables

- static const int DPI_OFFSET = -KeyAction::DPI_CYCLE_UP + 1
- static const int DPI_CUST_IDX = KeyAction::DPI_CUSTOM + DPI_OFFSET

13.164.1 Variable Documentation

13.164.1.1 const int DPI_CUST_IDX = KeyAction::DPI_CUSTOM + DPI_OFFSET [static]

Definition at line 7 of file rebindwidget.cpp.

Referenced by RebindWidget::on_dpiBox_currentIndexChanged().

13.164.1.2 const int DPI_OFFSET = -KeyAction::DPI_CYCLE_UP + 1 [static]

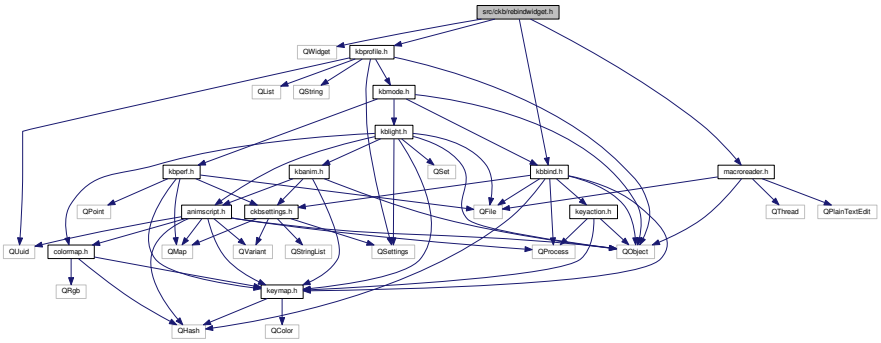
Definition at line 6 of file rebindwidget.cpp.

Referenced by RebindWidget::applyChanges(), RebindWidget::setBind(), and RebindWidget::setSelection().

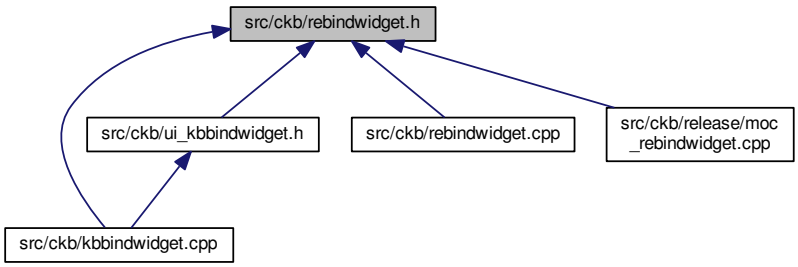
13.165 src/ckb/rebindwidget.h File Reference

```
#include <QWidget>
#include "kbbind.h"
#include "kbprofile.h"
#include "macroreader.h"
```

Include dependency graph for rebindwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class RebindWidget

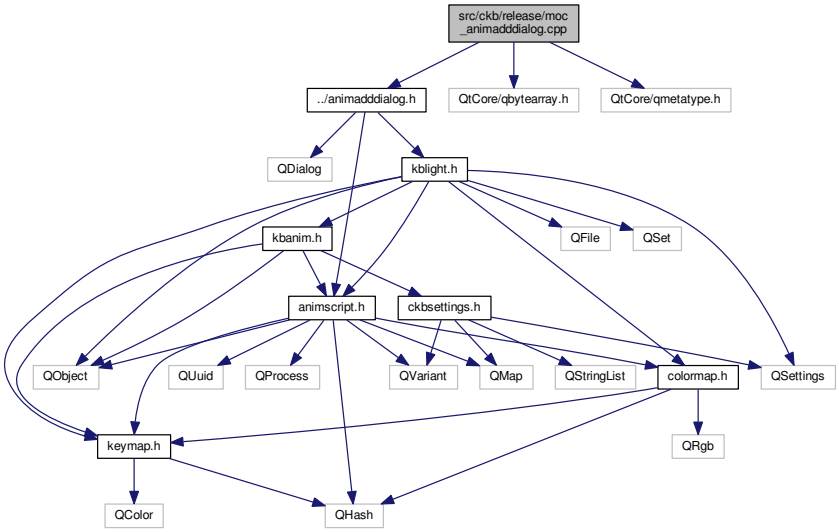
Namespaces

- Ui

13.166 src/ckb/release/moc_animadddialog.cpp File Reference

```
#include "../animadddialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_animadddialog.cpp:



Data Structures

- struct qt_meta_stringdata_AnimAddDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

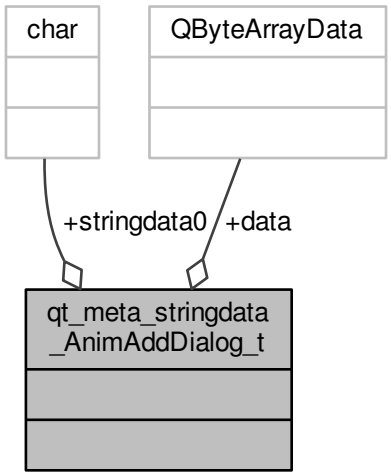
- static const qt_meta_stringdata_AnimAddDialog_t qt_meta_stringdata_AnimAddDialog
- static const uint qt_meta_data_AnimAddDialog []

13.166.1 Data Structure Documentation

13.166.1.1 struct qt_meta_stringdata_AnimAddDialog_t

Definition at line 21 of file moc_animadddialog.cpp.

Collaboration diagram for qt_meta_stringdata_AnimAddDialog_t:



Data Fields

QByteArrayData	data[7]	
char	stringdata0[95]	

13.166.2 Macro Definition Documentation

13.166.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_AnimAddDialog_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_animadddialog.cpp.

13.166.3 Variable Documentation

13.166.3.1 const uint qt_meta_data_AnimAddDialog[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       3,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,


        1,    1,   29,    2, 0x08 ,
        4,    1,   32,    2, 0x08 ,
        5,    1,   35,    2, 0x08 ,


    QMetaType::Void, QMetaType::Int,    3,
    QMetaType::Void, QMetaType::Int,    3,
    QMetaType::Void, QMetaType::Bool,    6,

        0
}
```

Definition at line 47 of file moc_animadddialog.cpp.

13.166.3.2 const qt_meta_stringdata_AnimAddDialog_t qt_meta_stringdata_AnimAddDialog [static]

Initial value:

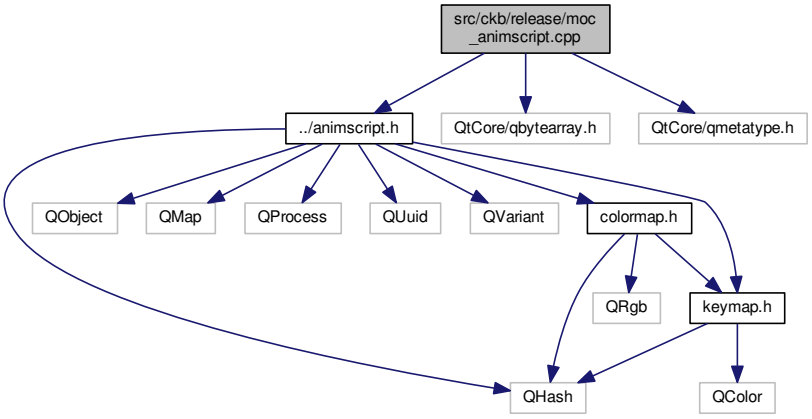
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  14    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  35    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  36    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 22 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  42    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  65    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimAddDialog_t, stringdata0) +  87    -  6  * sizeof(
      QByteArrayData))   )

    },
    "AnimAddDialog\0on_animBox_activated\0\0"
    "index\0on_presetBox_activated\0"
    "on_previewBox_clicked\0checked"
}
```

Definition at line 30 of file moc_animadddialog.cpp.

13.167 src/ckb/release/moc_animscript.cpp File Reference

```
#include "../animscript.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_animscript.cpp:



Data Structures

- struct qt_meta_stringdata_AnimScript_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

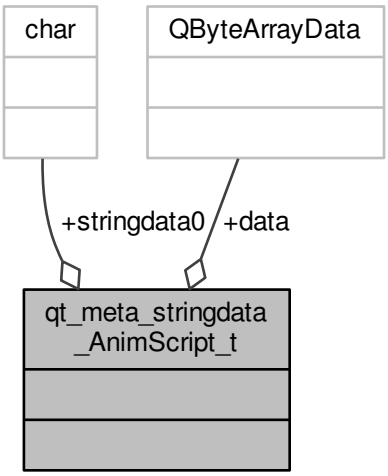
- static const qt_meta_stringdata_AnimScript_t qt_meta_stringdata_AnimScript
- static const uint qt_meta_data_AnimScript [ ]

13.167.1 Data Structure Documentation

13.167.1.1 struct qt_meta_stringdata_AnimScript_t

Definition at line 21 of file moc_animscript.cpp.

Collaboration diagram for qt_meta_stringdata_AnimScript_t:



Data Fields

QByteArrayData	data[3]	
char	stringdata0[24]	

13.167.2 Macro Definition Documentation

13.167.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_AnimScript_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_animscript.cpp.

13.167.3 Variable Documentation

13.167.3.1 const uint qt_meta_data_AnimScript[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       1,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
```

```
        0,

        1,    0,   19,    2, 0x08 ,

    QMetaType::Void,

        0
}
```

Definition at line 41 of file moc_animscript.cpp.

13.167.3.2 const qt_meta_stringdata_AnimScript_t qt_meta_stringdata_AnimScript [static]

Initial value:

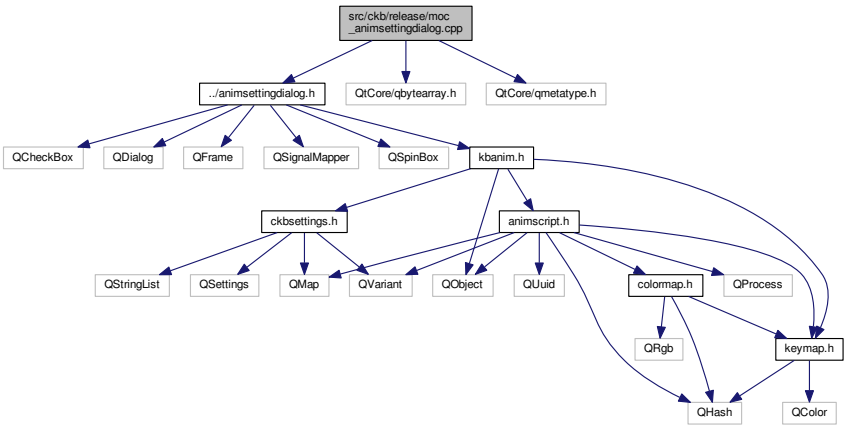
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 10 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimScript_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimScript_t, stringdata0) +  11    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimScript_t, stringdata0) +  23    -  2  * sizeof(
      QByteArrayData))   )

    },
    "AnimScript\0readProcess\0"
}
```

Definition at line 30 of file moc_animscript.cpp.

13.168 src/ckb/release/moc_animsettingdialog.cpp File Reference

```
#include "../animsettingdialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_animsettingdialog.cpp:



Data Structures

- struct qt_meta_stringdata_AnimSettingDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

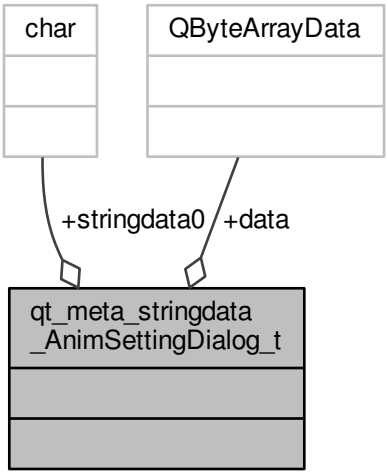
- static const qt_meta_stringdata_AnimSettingDialog_t qt_meta_stringdata_AnimSettingDialog
- static const uint qt_meta_data_AnimSettingDialog []

13.168.1 Data Structure Documentation

13.168.1.1 struct qt_meta_stringdata_AnimSettingDialog_t

Definition at line 21 of file moc_animsettingdialog.cpp.

Collaboration diagram for qt_meta_stringdata_AnimSettingDialog_t:



Data Fields

QByteArrayData	data[15]	
char	stringdata0[246]	

13.168.2 Macro Definition Documentation

13.168.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_AnimSettingDialog_t,
      stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_animsettingdialog.cpp.

13.168.3 Variable Documentation

13.168.3.1 const uint qt_meta_data_AnimSettingDialog[] [static]

Definition at line 60 of file moc_animsettingdialog.cpp.

13.168.3.2 const qt_meta_stringdata_AnimSettingDialog_t qt_meta_stringdata_AnimSettingDialog [static]

Initial value:

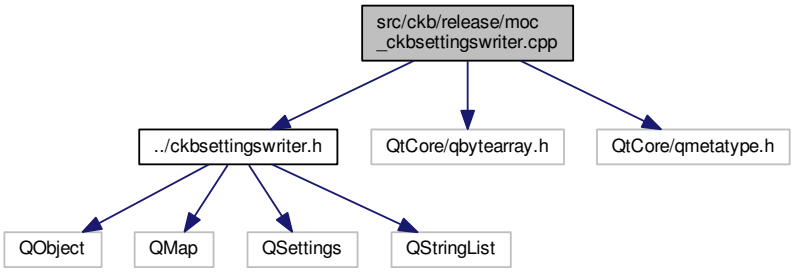
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 17 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  0    -  0  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  18    -  1  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  30    -  2  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  31    -  3  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  40    -  4  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  52    -  5  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  69    -  6  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  74    -  7  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  94    -  8  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 24 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  106    -  9
      * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  131    -  10
       * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 25 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  136    -  11
       * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 26 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  162    -  12
       * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 27 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  189    -  13
       * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 28 ,   qptrdiff(offsetof(
      qt_meta_stringdata_AnimSettingDialog_t, stringdata0) +  217    -  14
       * sizeof(QByteArrayData))   )

    },
    "AnimSettingDialog\0newDuration\0\0duration\0"
    "updateStops\0angleDialChanged\0name\0"
    "angleSpinnerChanged\0updateParam\0"
    "on_delayBox_valueChanged\0arg1\0"
    "on_repeatBox_valueChanged\0"
    "on_kpDelayBox_valueChanged\0"
    "on_kpRepeatBox_valueChanged\0"
    "on_kpReleaseBox_stateChanged"
}
```

Definition at line 30 of file moc_animsettingdialog.cpp.

13.169 src/ckb/release/moc_ckbsettingswriter.cpp File Reference

```
#include "../ckbsettingswriter.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_ckbsettingswriter.cpp:



Data Structures

- struct qt_meta_stringdata_CkbSettingsWriter_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

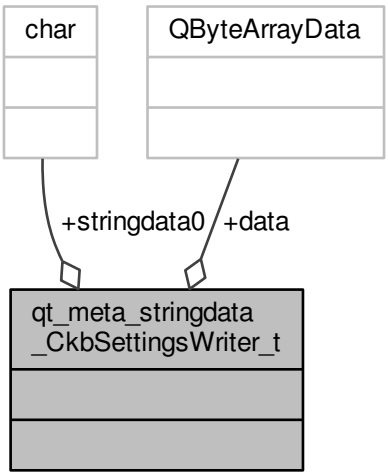
- static const qt_meta_stringdata_CkbSettingsWriter_t qt_meta_stringdata_CkbSettingsWriter
- static const uint qt_meta_data_CkbSettingsWriter [ ]

13.169.1 Data Structure Documentation

13.169.1.1 struct qt_meta_stringdata_CkbSettingsWriter_t

Definition at line 21 of file moc_ckbsettingswriter.cpp.

Collaboration diagram for qt_meta_stringdata_CkbSettingsWriter_t:



Data Fields

QByteArrayData	data[3]	
char	stringdata0[23]	

13.169.2 Macro Definition Documentation

13.169.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_CkbSettingsWriter_t,
      stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_ckbsettingswriter.cpp.

13.169.3 Variable Documentation

13.169.3.1 const uint qt_meta_data_CkbSettingsWriter[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       1,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,

        1,    0,   19,    2, 0x0a ,

    QMetaType::Void,

        0
}
```

Definition at line 41 of file moc_ckbsettingswriter.cpp.

13.169.3.2 const qt_meta_stringdata_CkbSettingsWriter_t qt_meta_stringdata_CkbSettingsWriter [static]

Initial value:

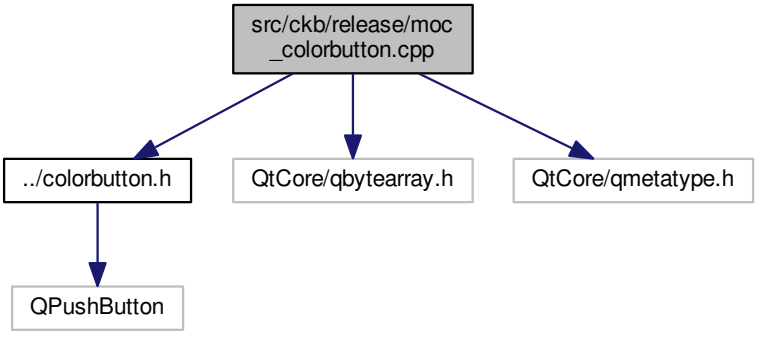
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 17 ,   qptrdiff(offsetof(
      qt_meta_stringdata_CkbSettingsWriter_t, stringdata0) +  0    -  0  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 3 ,   qptrdiff(offsetof(
      qt_meta_stringdata_CkbSettingsWriter_t, stringdata0) +  18    -  1  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_CkbSettingsWriter_t, stringdata0) +  22    -  2  *
       sizeof(QByteArrayData))   )

    },
    "CkbSettingsWriter\0run\0"
}
```

Definition at line 30 of file moc_ckbsettingswriter.cpp.

13.170 src/ckb/release/moc_colorbutton.cpp File Reference

```
#include "../colorbutton.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_colorbutton.cpp:



Data Structures

- struct qt_meta_stringdata_ColorButton_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

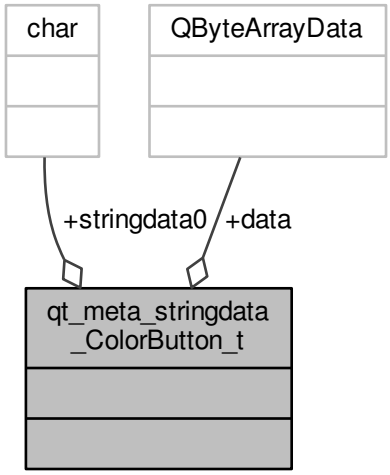
- static const qt_meta_stringdata_ColorButton_t qt_meta_stringdata_ColorButton
- static const uint qt_meta_data_ColorButton [ ]

13.170.1 Data Structure Documentation

13.170.1.1 struct qt_meta_stringdata_ColorButton_t

Definition at line 21 of file moc_colorbutton.cpp.

Collaboration diagram for qt_meta_stringdata_ColorButton_t:



Data Fields

QByteArrayData	data[5]	
char	stringdata0[45]	

13.170.2 Macro Definition Documentation

13.170.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_ColorButton_t, stringdata0) + ofs
      \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_colorbutton.cpp.

13.170.3 Variable Documentation

13.170.3.1 const uint qt_meta_data_ColorButton[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       2,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
       1,


       1,    1,   24,    2, 0x06 ,


       4,    0,   27,    2, 0x08 ,


    QMetaType::Void, QMetaType::QColor,    3,


    QMetaType::Void,

       0
}
```

Definition at line 44 of file moc_colorbutton.cpp.

13.170.3.2 const qt_meta_stringdata_ColorButton_t qt_meta_stringdata_ColorButton [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ColorButton_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ColorButton_t, stringdata0) +  12    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ColorButton_t, stringdata0) +  25    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ColorButton_t, stringdata0) +  26    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ColorButton_t, stringdata0) +  35    -  4  * sizeof(
      QByteArrayData))   )

    },
    "ColorButton\0colorChanged\0\0newColor\0"
    "pickColor"
}
```

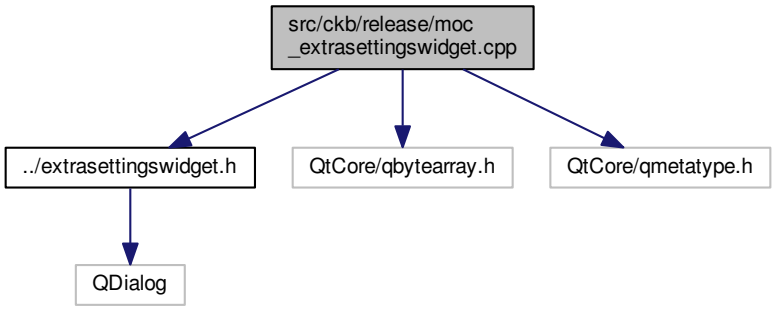
Definition at line 30 of file moc_colorbutton.cpp.

13.171 src/ckb/release/moc_extrasettingswidget.cpp File Reference

```
#include "../extrasettingswidget.h"
```

```
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_extrasettingswidget.cpp:



Data Structures

- struct qt_meta_stringdata_ExtraSettingsWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

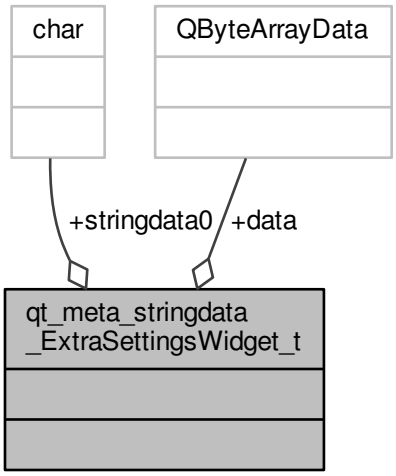
- static const qt_meta_stringdata_ExtraSettingsWidget_t qt_meta_stringdata_ExtraSettingsWidget
- static const uint qt_meta_data_ExtraSettingsWidget [ ]

13.171.1 Data Structure Documentation

13.171.1.1 struct qt_meta_stringdata_ExtraSettingsWidget_t

Definition at line 21 of file moc_extrasettingswidget.cpp.

Collaboration diagram for qt_meta_stringdata_ExtraSettingsWidget_t:



Data Fields

QByteArrayData	data[13]	
char	stringdata0[236]	

13.171.2 Macro Definition Documentation

13.171.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_ExtraSettingsWidget_t,
      stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_extrasettingswidget.cpp.

13.171.3 Variable Documentation

13.171.3.1 const uint qt_meta_data_ExtraSettingsWidget[ ] [static]

Definition at line 57 of file moc_extrasettingswidget.cpp.

13.171.3.2 const qt_meta_stringdata_ExtraSettingsWidget_t qt_meta_stringdata_ExtraSettingsWidget [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
```

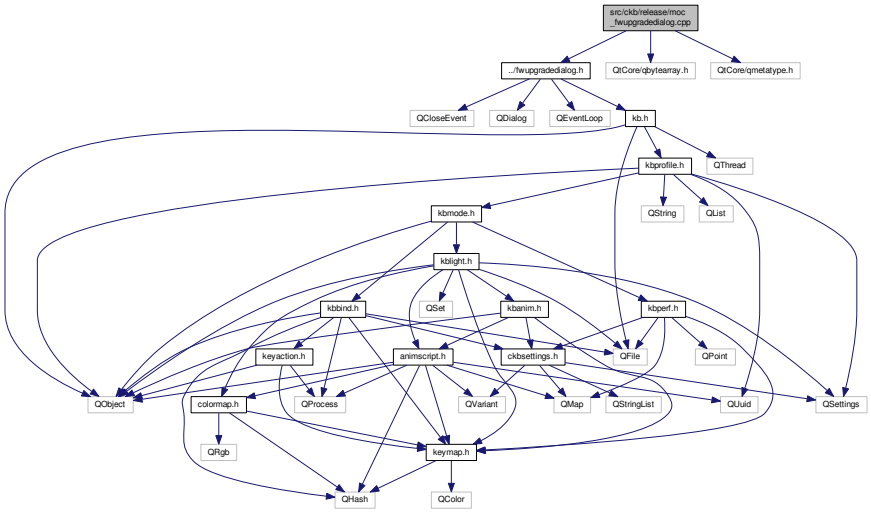
```
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  0    -  0
        * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 18 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  20    -
      1  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  39    -
      2  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  40    -
      3  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 24 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  48    -
      4  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 25 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  73    -
      5  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 22 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  99    -
      6  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  122    -
       7  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  127    -
       8  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  148    -
       9  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  169    -
       10  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 25 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  190    -
       11  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ExtraSettingsWidget_t, stringdata0) +  216    -
       12  * sizeof(QByteArrayData))   )

    },
    "ExtraSettingsWidget\0on_trayBox_clicked\0"
    "\0checked\0on_brightnessBox_clicked\0"
    "on_animScanButton_clicked\0"
    "on_fpsBox_valueChanged\0arg1\0"
    "on_ditherBox_clicked\0on_mAccelBox_clicked\0"
    "on_sAccelBox_clicked\0on_sSpeedBox_valueChanged\0"
    "on_delayBox_clicked"
}
```

Definition at line 30 of file moc_extrasettingswidget.cpp.

13.172 src/ckb/release/moc_fwupgradedialog.cpp File Reference

```
#include "../fwupgradedialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_fwupgradedialog.cpp:



Data Structures

- struct qt_meta_stringdata_FwUpgradeDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

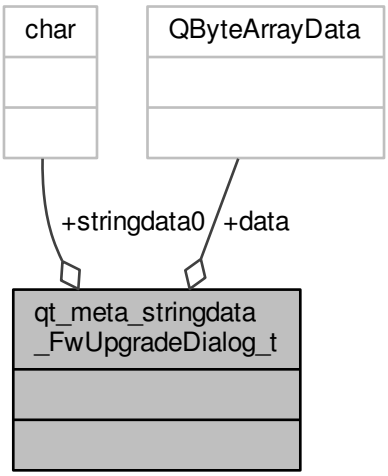
- static const qt_meta_stringdata_FwUpgradeDialog_t qt_meta_stringdata_FwUpgradeDialog
- static const uint qt_meta_data_FwUpgradeDialog [ ]

13.172.1 Data Structure Documentation

13.172.1.1 struct qt_meta_stringdata_FwUpgradeDialog_t

Definition at line 21 of file moc_fwupgradedialog.cpp.

Collaboration diagram for qt_meta_stringdata_FwUpgradeDialog_t:



Data Fields

QByteArrayData	data[10]	
char	stringdata0[133]	

13.172.2 Macro Definition Documentation

13.172.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_FwUpgradeDialog_t, stringdata0
      ) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_fwupgradedialog.cpp.

13.172.3 Variable Documentation

13.172.3.1 const uint qt_meta_data_FwUpgradeDialog[ ] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       5,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
       0,
       0,


       1,    2,   39,    2, 0x08 ,
       5,    1,   44,    2, 0x08 ,
       7,    0,   47,    2, 0x08 ,
       8,    0,   48,    2, 0x08 ,
       9,    0,   49,    2, 0x08 ,


    QMetaType::Void, QMetaType::Int, QMetaType::Int,    3,    4,
    QMetaType::Void, QMetaType::Bool,    6,
    QMetaType::Void,
    QMetaType::Void,
    QMetaType::Void,

       0
}
```

Definition at line 51 of file moc_fwupgradedialog.cpp.

13.172.3.2 const qt_meta_stringdata_FwUpgradeDialog_t qt_meta_stringdata_FwUpgradeDialog [static]

Initial value:

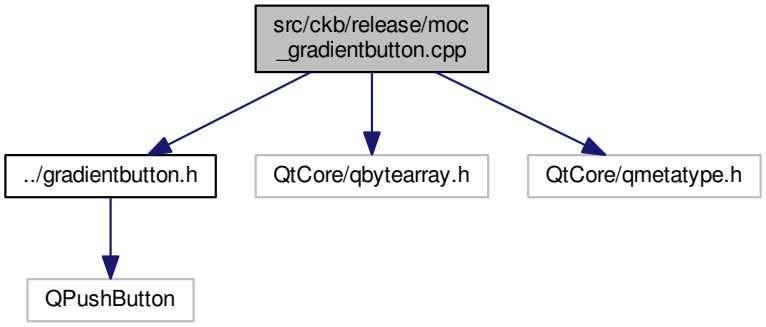
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 15 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  0    -  0  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  16    -  1  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  33    -  2  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  34    -  3  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  42    -  4  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  48    -  5  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  65    -  6  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  75    -  7  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  85    -  8  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_FwUpgradeDialog_t, stringdata0) +  109    -  9  *
      sizeof(QByteArrayData))   )

    },
    "FwUpgradeDialog\0fwUpdateProgress\0\0"
    "current\0total\0fwUpdateFinished\0succeeded\0"
    "removeDev\0on_cancelButton_clicked\0"
    "on_actionButton_clicked"
}
```

Definition at line 30 of file moc_fwupgradedialog.cpp.

13.173 src/ckb/release/moc_gradientbutton.cpp File Reference

```
#include "../gradientbutton.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_gradientbutton.cpp:



Data Structures

- struct qt_meta_stringdata_GradientButton_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

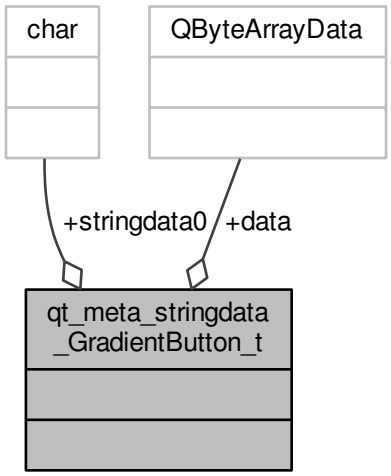
- static const qt_meta_stringdata_GradientButton_t qt_meta_stringdata_GradientButton
- static const uint qt_meta_data_GradientButton []

13.173.1 Data Structure Documentation

13.173.1.1 struct qt_meta_stringdata_GradientButton_t

Definition at line 21 of file moc_gradientbutton.cpp.

Collaboration diagram for qt_meta_stringdata_GradientButton_t:



Data Fields

QByteArrayData	data[4]	
char	stringdata0[45]	

13.173.2 Macro Definition Documentation

13.173.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_GradientButton_t, stringdata0)
      + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_gradientbutton.cpp.

13.173.3 Variable Documentation

13.173.3.1 const uint qt_meta_data_GradientButton[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       2,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        1,

        1,    0,   24,    2, 0x06 ,

        3,    0,   25,    2, 0x08 ,

    QMetaType::Void,

    QMetaType::Void,

        0
}
```

Definition at line 43 of file moc_gradientbutton.cpp.

13.173.3.2 const qt_meta_stringdata_GradientButton_t qt_meta_stringdata_GradientButton [static]

Initial value:

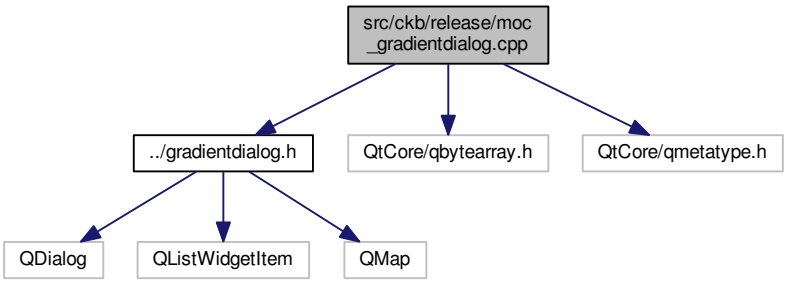
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientButton_t, stringdata0) +  0    -  0  * sizeof
      (QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 15 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientButton_t, stringdata0) +  15    -  1  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientButton_t, stringdata0) +  31    -  2  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientButton_t, stringdata0) +  32    -  3  *
      sizeof(QByteArrayData))   )

    },
    "GradientButton\0gradientChanged\0\0"
    "pickGradient"
}
```

Definition at line 30 of file moc_gradientbutton.cpp.

13.174 src/ckb/release/moc_gradientdialog.cpp File Reference

```
#include "../gradientdialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_gradientdialog.cpp:



Data Structures

- struct qt_meta_stringdata_GradientDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

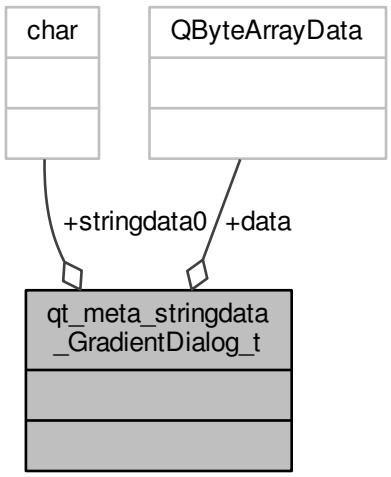
- static const qt_meta_stringdata_GradientDialog_t qt_meta_stringdata_GradientDialog
- static const uint qt_meta_data_GradientDialog [ ]

13.174.1 Data Structure Documentation

13.174.1.1 struct qt_meta_stringdata_GradientDialog_t

Definition at line 21 of file moc_gradientdialog.cpp.

Collaboration diagram for qt_meta_stringdata_GradientDialog_t:



Data Fields

QByteArrayData	data[17]	
char	stringdata0[266]	

13.174.2 Macro Definition Documentation

13.174.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_GradientDialog_t, stringdata0)
      + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_gradientdialog.cpp.

13.174.3 Variable Documentation

13.174.3.1 const uint qt_meta_data_GradientDialog[] [static]

Definition at line 62 of file moc_gradientdialog.cpp.

13.174.3.2 const qt_meta_stringdata_GradientDialog_t qt_meta_stringdata_GradientDialog [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  0    -  0  * sizeof
      (QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  15    -  1  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  30    -  2  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  31    -  3  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  37    -  4  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  49    -  5  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  58    -  6  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 32 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  71    -  7  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  104    -  8  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  121    -  9  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  129    -  10  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 24 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  138    -  11  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  163    -  12  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  168    -  13  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  190    -  14  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  214    -  15  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 27 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialog_t, stringdata0) +  238    -  16  *
      sizeof(QByteArrayData))   )

    },
    "GradientDialog\0currentChanged\0\0color\0"
    "spontaneous\0position\0colorChanged\0"
    "on_presetList_currentItemChanged\0"
    "QListWidgetItem*\0current\0previous\0"
```

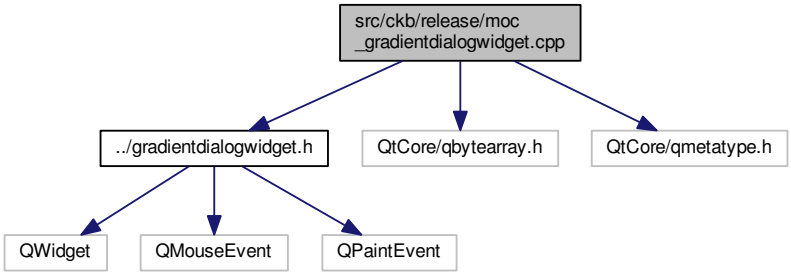
```
    "on_presetName_textEdited\0arg1\0"
    "on_presetSave_clicked\0on_presetDelete_clicked\0"
    "on_stopPos_valueChanged\0"
    "on_stopOpacity_valueChanged"
}
```

Definition at line 30 of file moc_gradientdialog.cpp.

13.175 src/ckb/release/moc_gradientdialogwidget.cpp File Reference

```
#include "../gradientdialogwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_gradientdialogwidget.cpp:



Data Structures

- struct qt_meta_stringdata_GradientDialogWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

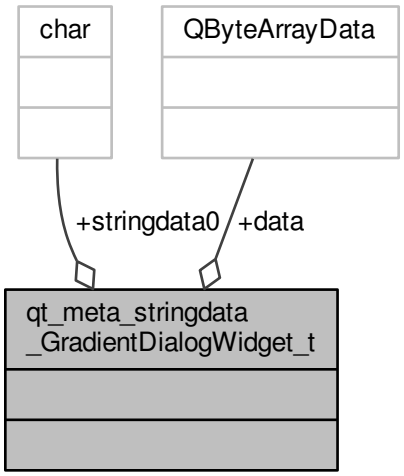
- static const qt_meta_stringdata_GradientDialogWidget_t qt_meta_stringdata_GradientDialogWidget
- static const uint qt_meta_data_GradientDialogWidget [ ]

13.175.1 Data Structure Documentation

13.175.1.1 struct qt_meta_stringdata_GradientDialogWidget_t

Definition at line 21 of file moc_gradientdialogwidget.cpp.

Collaboration diagram for qt_meta_stringdata_GradientDialogWidget_t:



Data Fields

QByteArrayData	data[6]	
char	stringdata0[64]	

13.175.2 Macro Definition Documentation

13.175.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_GradientDialogWidget_t,
      stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_gradientdialogwidget.cpp.

13.175.3 Variable Documentation

13.175.3.1 const uint qt_meta_data_GradientDialogWidget[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       1,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        1,

        1,    3,   19,    2, 0x06 ,

    QMetaType::Void, QMetaType::QColor, QMetaType::Bool, QMetaType::Int,    3,    4,    5,

        0
}
```

Definition at line 45 of file moc_gradientdialogwidget.cpp.

13.175.3.2 const qt_meta_stringdata_GradientDialogWidget_t qt_meta_stringdata_GradientDialogWidget [static]

Initial value:

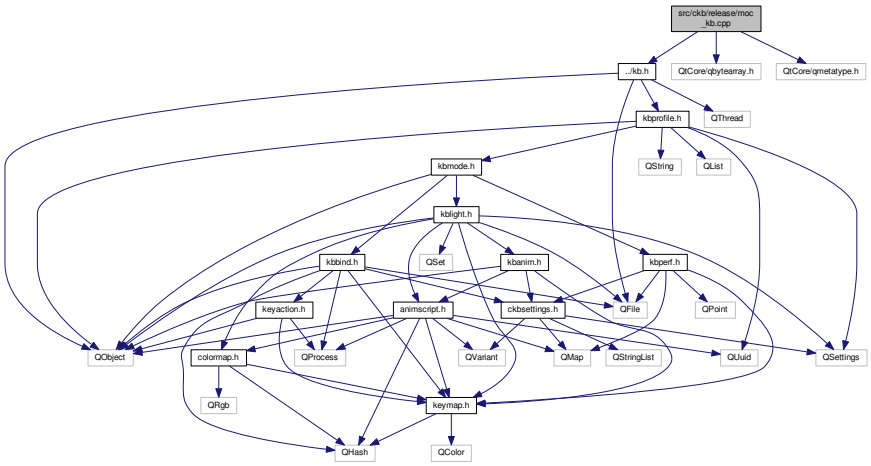
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialogWidget_t, stringdata0) +  0    -
       0  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialogWidget_t, stringdata0) +  21    -
        1  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialogWidget_t, stringdata0) +  36    -
        2  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialogWidget_t, stringdata0) +  37    -
        3  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialogWidget_t, stringdata0) +  43    -
        4  * sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_GradientDialogWidget_t, stringdata0) +  55    -
        5  * sizeof(QByteArrayData))   )

    },
    "GradientDialogWidget\0currentChanged\0"
    "\0color\0spontaneous\0position"
}
```

Definition at line 30 of file moc_gradientdialogwidget.cpp.

13.176 src/ckb/release/moc_kb.cpp File Reference

```
#include "../kb.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kb.cpp:



Data Structures

- struct qt_meta_stringdata_Kb_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

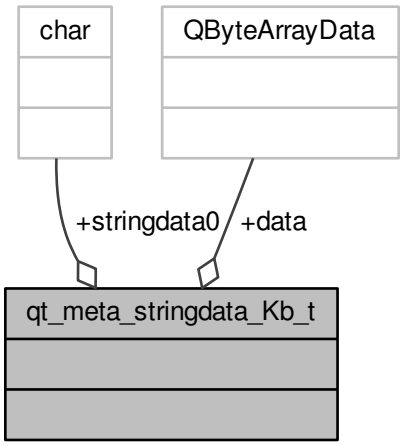
- static const qt_meta_stringdata_Kb_t qt_meta_stringdata_Kb
- static const uint qt_meta_data_Kb [ ]

13.176.1 Data Structure Documentation

13.176.1.1 struct qt_meta_stringdata_Kb_t

Definition at line 21 of file moc_kb.cpp.

Collaboration diagram for qt_meta_stringdata_Kb_t:



Data Fields

QByteArrayData	data[20]	
char	stringdata0[214]	

13.176.2 Macro Definition Documentation

13.176.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_Kb_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kb.cpp.

13.176.3 Variable Documentation

13.176.3.1 const uint qt_meta_data_Kb[] [static]

Definition at line 63 of file moc_kb.cpp.

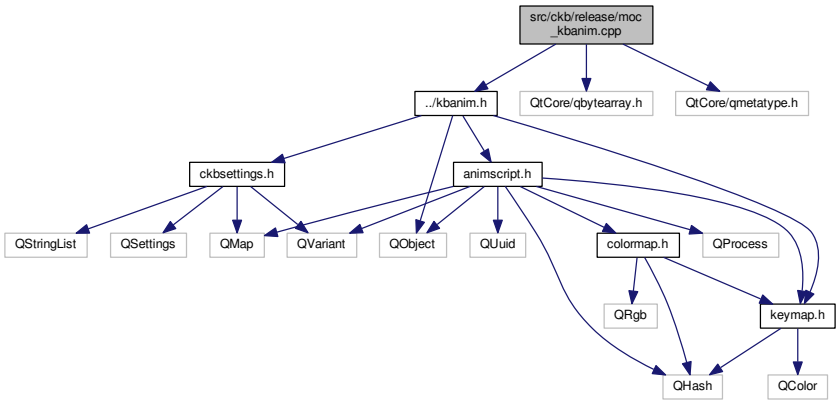
13.176.3.2 const qt_meta_stringdata_Kb_t qt_meta_stringdata_Kb [static]

Definition at line 30 of file moc_kb.cpp.

13.177 src/ckb/release/moc_kbanim.cpp File Reference

```
#include "../kbanim.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbanim.cpp:



Data Structures

- struct qt_meta_stringdata_KbAnim_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

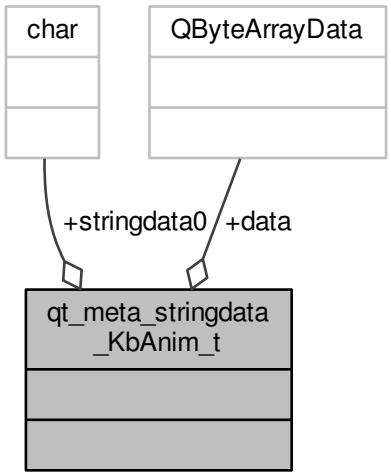
- static const qt_meta_stringdata_KbAnim_t qt_meta_stringdata_KbAnim
- static const uint qt_meta_data_KbAnim [ ]

13.177.1 Data Structure Documentation

13.177.1.1 struct qt_meta_stringdata_KbAnim_t

Definition at line 21 of file moc_kbanim.cpp.

Collaboration diagram for qt_meta_stringdata_KbAnim_t:



Data Fields

QByteArrayData	data[7]	
char	stringdata0[48]	

13.177.2 Macro Definition Documentation

13.177.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbAnim_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbanim.cpp.

13.177.3 Variable Documentation

13.177.3.1 const uint qt_meta_data_KbAnim[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       1,   14,
       0,    0,
       0,
```

```
        0,

        1, 0x0,    5,   18,

        2, uint(KbAnim::Normal),
        3, uint(KbAnim::Add),
        4, uint(KbAnim::Subtract),
        5, uint(KbAnim::Multiply),
        6, uint(KbAnim::Divide),

        0
}
```

Definition at line 46 of file moc_kbanim.cpp.

13.177.3.2 const qt_meta_stringdata_KbAnim_t qt_meta_stringdata_KbAnim [static]

Initial value:

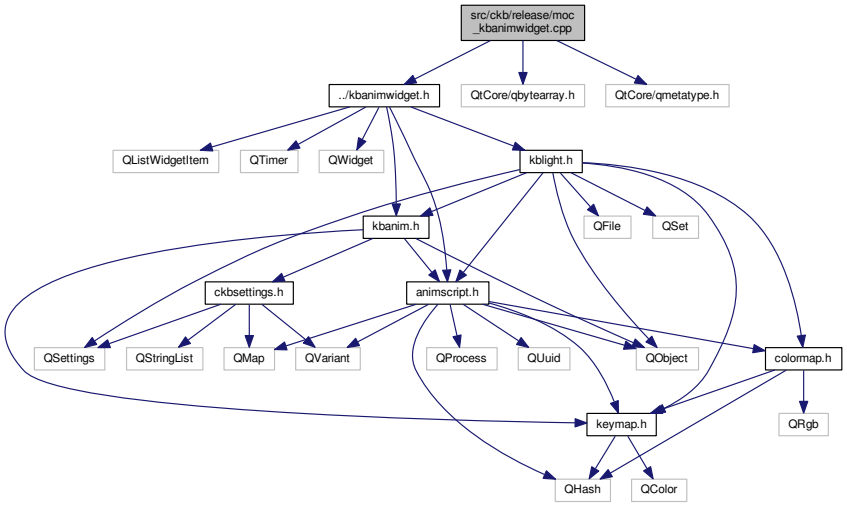
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  0    -  0  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  7    -  1  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  12    -  2  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 3 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  19    -  3  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  23    -  4  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  32    -  5  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbAnim_t, stringdata0) +  41    -  6  * sizeof(QByteArrayData
      ))   )

    },
    "KbAnim\0Mode\0Normal\0Add\0Subtract\0"
    "Multiply\0Divide"
}
```

Definition at line 30 of file moc_kbanim.cpp.

13.178 src/ckb/release/moc_kbanimwidget.cpp File Reference

```
#include "../kbanimwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbanimwidget.cpp:



Data Structures

- struct qt_meta_stringdata_KbAnimWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

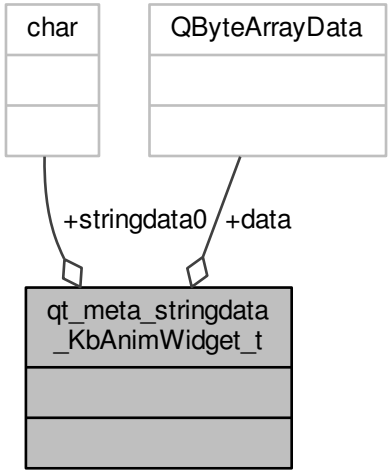
- static const qt_meta_stringdata_KbAnimWidget_t qt_meta_stringdata_KbAnimWidget
- static const uint qt_meta_data_KbAnimWidget [ ]

13.178.1 Data Structure Documentation

13.178.1.1 struct qt_meta_stringdata_KbAnimWidget_t

Definition at line 21 of file moc_kbanimwidget.cpp.

Collaboration diagram for qt_meta_stringdata_KbAnimWidget_t:



Data Fields

QByteArrayData	data[25]	
char	stringdata0[383]	

13.178.2 Macro Definition Documentation

13.178.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbAnimWidget_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbanimwidget.cpp.

13.178.3 Variable Documentation

13.178.3.1 const uint qt_meta_data_KbAnimWidget[] [static]

Definition at line 74 of file moc_kbanimwidget.cpp.

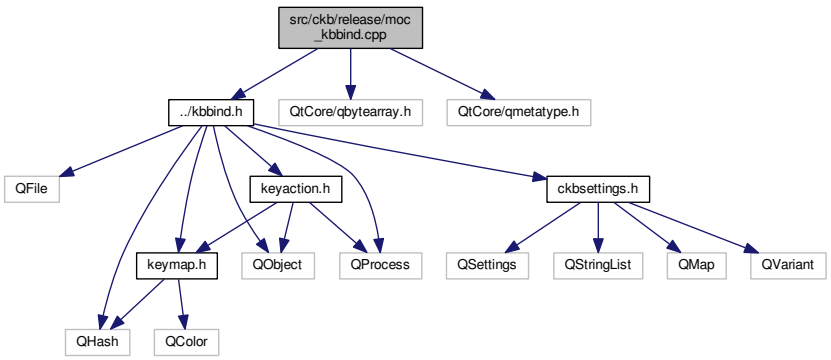
13.178.3.2 const qt_meta_stringdata_KbAnimWidget_t qt_meta_stringdata_KbAnimWidget [static]

Definition at line 30 of file moc_kbanimwidget.cpp.

13.179 src/ckb/release/moc_kbbind.cpp File Reference

```
#include "../kbbind.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbbind.cpp:



Data Structures

- struct qt_meta_stringdata_KbBind_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

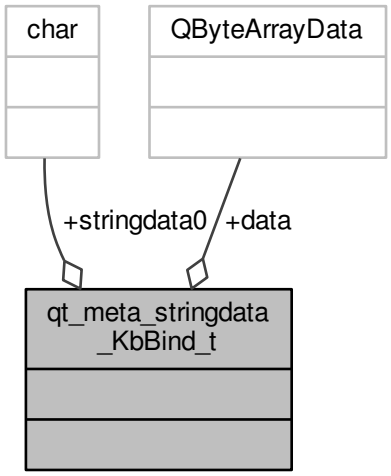
- static const qt_meta_stringdata_KbBind_t qt_meta_stringdata_KbBind
- static const uint qt_meta_data_KbBind []

13.179.1 Data Structure Documentation

13.179.1.1 struct qt_meta_stringdata_KbBind_t

Definition at line 21 of file moc_kbbind.cpp.

Collaboration diagram for qt_meta_stringdata_KbBind_t:



Data Fields

QByteArrayData	data[8]	
char	stringdata0[56]	

13.179.2 Macro Definition Documentation

13.179.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbBind_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbbind.cpp.

13.179.3 Variable Documentation

13.179.3.1 const uint qt_meta_data_KbBind[] [static]

Definition at line 47 of file moc_kbbind.cpp.

13.179.3.2 const qt_meta_stringdata_KbBind_t qt_meta_stringdata_KbBind [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  0    -  0  * sizeof(QByteArrayData)
```

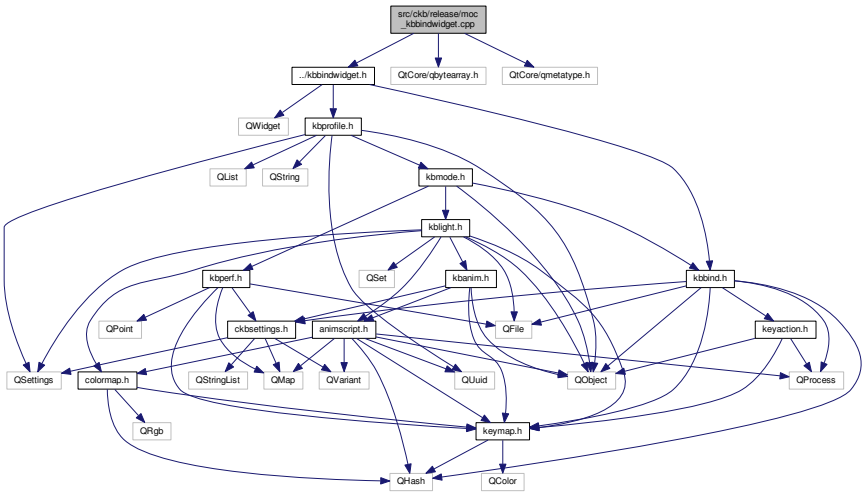
```
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  7    -  1  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  15    -  2  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  16    -  3  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  30    -  4  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  38    -  5  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 3 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  47    -  6  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBind_t, stringdata0) +  51    -  7  * sizeof(QByteArrayData
      ))   )

    },
    "KbBind\0didLoad\0\0layoutChanged\0updated\0"
    "keyEvent\0key\0down"
}
```

Definition at line 30 of file moc_kbbind.cpp.

13.180 src/ckb/release/moc_kbbindwidget.cpp File Reference

```
#include "../kbbindwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbbindwidget.cpp:



Data Structures

- struct qt_meta_stringdata_KbBindWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

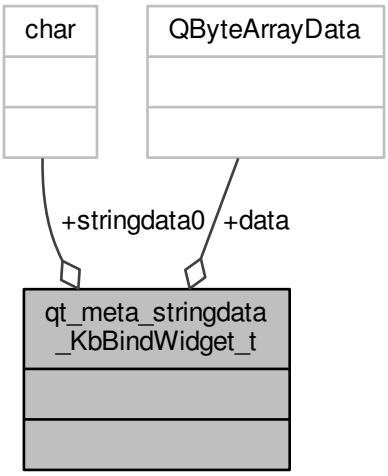
- static const qt_meta_stringdata_KbBindWidget_t qt_meta_stringdata_KbBindWidget
- static const uint qt_meta_data_KbBindWidget [ ]

13.180.1 Data Structure Documentation

13.180.1.1 struct qt_meta_stringdata_KbBindWidget_t

Definition at line 21 of file moc_kbbindwidget.cpp.

Collaboration diagram for qt_meta_stringdata_KbBindWidget_t:



Data Fields

QByteArrayData	data[9]	
char	stringdata0[120]	

13.180.2 Macro Definition Documentation

13.180.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbBindWidget_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbbindwidget.cpp.

13.180.3 Variable Documentation

13.180.3.1 const uint qt_meta_data_KbBindWidget[] [static]

Definition at line 49 of file moc_kbbindwidget.cpp.

13.180.3.2 const qt_meta_stringdata_KbBindWidget_t qt_meta_stringdata_KbBindWidget [static]

Initial value:

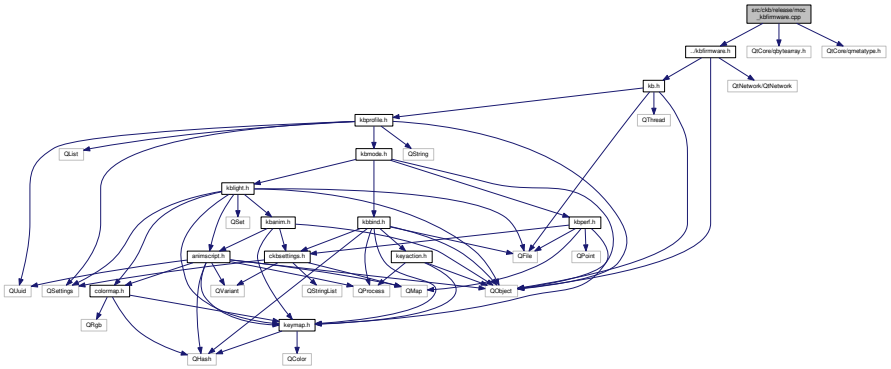
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 10 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  13    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  24    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  25    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  35    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  48    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  58    -  6  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 22 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  75    -  7  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbBindWidget_t, stringdata0) +  98    -  8  * sizeof(
      QByteArrayData))   )

    },
    "KbBindWidget\0updateBind\0\0newLayout\0"
    "newSelection\0selection\0updateSelDisplay\0"
    "on_resetButton_clicked\0on_copyButton_clicked"
}
```

Definition at line 30 of file moc_kbbindwidget.cpp.

13.181 src/ckb/release/moc_kbfirmware.cpp File Reference

```
#include "../kbfirmware.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbfirmware.cpp:



Data Structures

- struct qt_meta_stringdata_KbFirmware_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

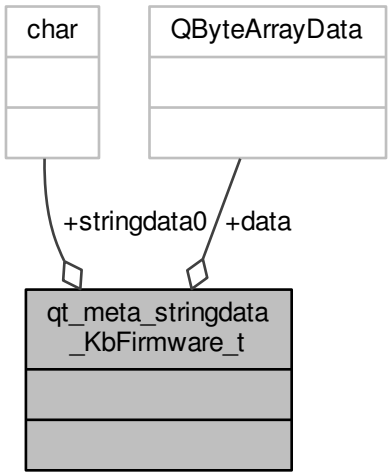
- static const qt_meta_stringdata_KbFirmware_t qt_meta_stringdata_KbFirmware
- static const uint qt_meta_data_KbFirmware [ ]

13.181.1 Data Structure Documentation

13.181.1.1 struct qt_meta_stringdata_KbFirmware_t

Definition at line 21 of file moc_kbfirmware.cpp.

Collaboration diagram for qt_meta_stringdata_KbFirmware_t:



Data Fields

QByteArrayData	data[7]	
char	stringdata0[77]	

13.181.2 Macro Definition Documentation

13.181.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbFirmware_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbfirmware.cpp.

13.181.3 Variable Documentation

13.181.3.1 const uint qt_meta_data_KbFirmware[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       3,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
```

```
       1,

       1,    0,   29,    2, 0x06 ,

       3,    1,   30,    2, 0x08 ,
       6,    0,   33,    2, 0x08 ,

    QMetaType::Void,

    QMetaType::Void, 0x80000000 | 4,    5,
    QMetaType::Void,

       0
}
```

Definition at line 46 of file moc_kbfirmware.cpp.

13.181.3.2 const qt_meta_stringdata_KbFirmware_t qt_meta_stringdata_KbFirmware [static]

Initial value:

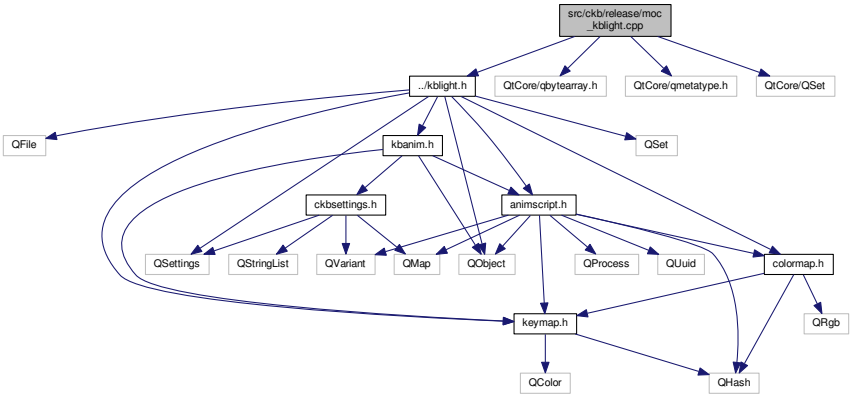
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 10 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 10 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  11    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  22    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 15 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  23    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  39    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  54    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbFirmware_t, stringdata0) +  60    -  6  * sizeof(
      QByteArrayData))   )

    },
    "KbFirmware\0downloaded\0\0processDownload\0"
    "QNetworkReply*\0reply\0downloadFinished"
}
```

Definition at line 30 of file moc_kbfirmware.cpp.

13.182 src/ckb/release/moc_kblight.cpp File Reference

```
#include "../kblight.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
#include <QtCore/QSet>
```

Include dependency graph for moc_kblight.cpp:



Data Structures

- struct qt_meta_stringdata_KbLight_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

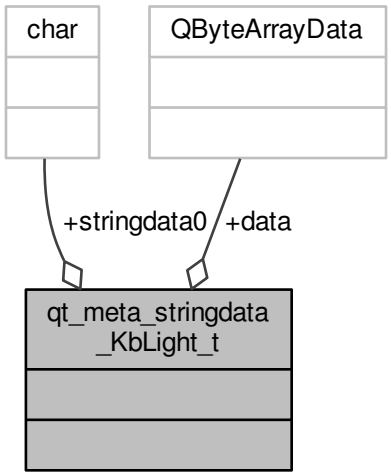
- static const qt_meta_stringdata_KbLight_t qt_meta_stringdata_KbLight
- static const uint qt_meta_data_KbLight [ ]

13.182.1 Data Structure Documentation

13.182.1.1 struct qt_meta_stringdata_KbLight_t

Definition at line 22 of file moc_kblight.cpp.

Collaboration diagram for qt_meta_stringdata_KbLight_t:



Data Fields

QByteArrayData	data[9]	
char	stringdata0[92]	

13.182.2 Macro Definition Documentation

13.182.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbLight_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 26 of file moc_kblight.cpp.

13.182.3 Variable Documentation

13.182.3.1 const uint qt_meta_data_KbLight[ ] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       3,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
```

```
       3,

       1,    0,   29,    2, 0x06 ,
       3,    0,   30,    2, 0x06 ,
       4,    2,   31,    2, 0x06 ,

    QMetaType::Void,
    QMetaType::Void,
    QMetaType::Void, 0x80000000 | 5, 0x80000000 | 7,    6,    8,

       0
}
```

Definition at line 50 of file moc_kblight.cpp.

13.182.3.2 const qt_meta_stringdata_KbLight_t qt_meta_stringdata_KbLight [static]

Initial value:

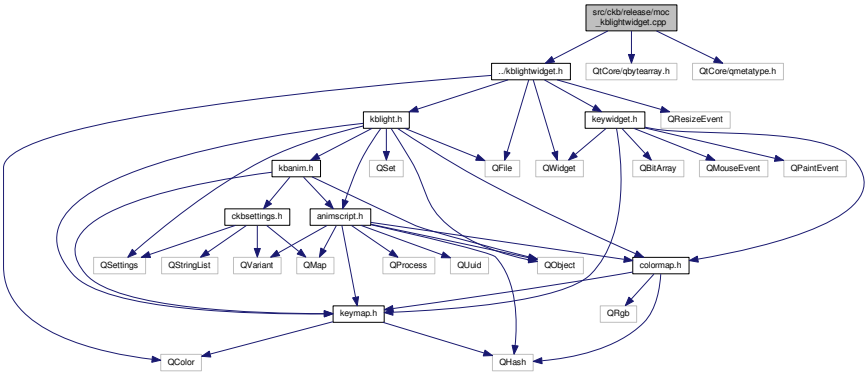
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  8    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  16    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  17    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  25    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  40    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  49    -  6  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  64    -  7  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbLight_t, stringdata0) +  78    -  8  * sizeof(
      QByteArrayData))   )

    },
    "KbLight\0didLoad\0\0updated\0frameDisplayed\0"
    "ColorMap\0animatedColors\0QSet<QString>\0"
    "indicatorList"
}
```

Definition at line 31 of file moc_kblight.cpp.

13.183 src/ckb/release/moc_kblightwidget.cpp File Reference

```
#include "../kblightwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kblightwidget.cpp:



Data Structures

- struct qt_meta_stringdata_KbLightWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

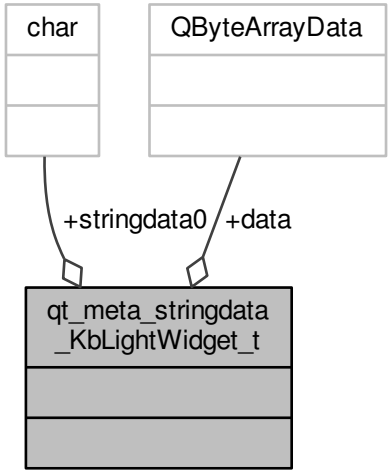
- static const qt_meta_stringdata_KbLightWidget_t qt_meta_stringdata_KbLightWidget
- static const uint qt_meta_data_KbLightWidget [ ]

13.183.1 Data Structure Documentation

13.183.1.1 struct qt_meta_stringdata_KbLightWidget_t

Definition at line 21 of file moc_kblightwidget.cpp.

Collaboration diagram for qt_meta_stringdata_KbLightWidget_t:



Data Fields

QByteArrayData	data[19]	
char	stringdata0[240]	

13.183.2 Macro Definition Documentation

13.183.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbLightWidget_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kblightwidget.cpp.

13.183.3 Variable Documentation

13.183.3.1 const uint qt_meta_data_KbLightWidget[] [static]

Definition at line 63 of file moc_kblightwidget.cpp.

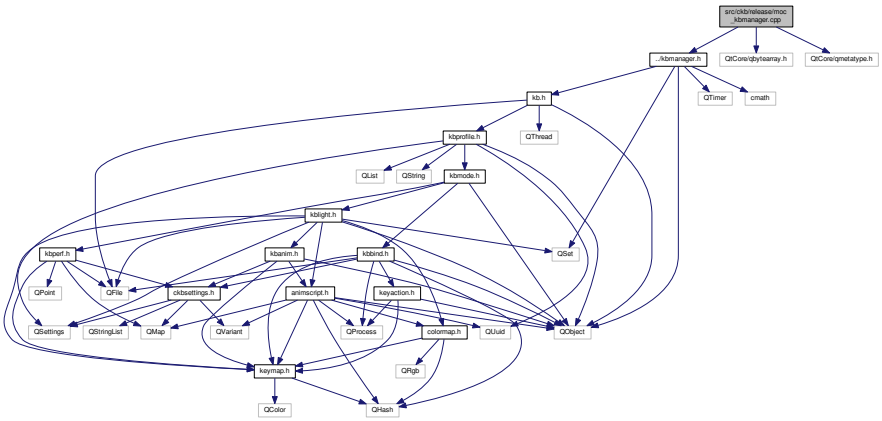
13.183.3.2 const qt_meta_stringdata_KbLightWidget_t qt_meta_stringdata_KbLightWidget [static]

Definition at line 30 of file moc_kblightwidget.cpp.

13.184 src/ckb/release/moc_kbmanager.cpp File Reference

```
#include "../kbmanager.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbmanager.cpp:



Data Structures

- struct qt_meta_stringdata_KbManager_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

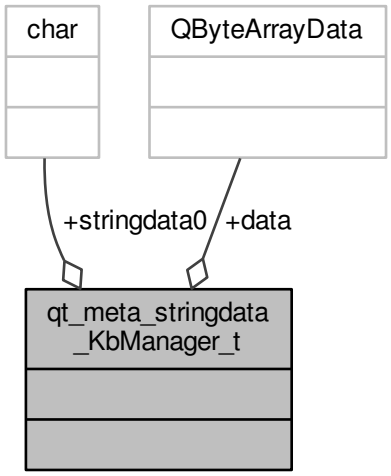
- static const qt_meta_stringdata_KbManager_t qt_meta_stringdata_KbManager
- static const uint qt_meta_data_KbManager [ ]

13.184.1 Data Structure Documentation

13.184.1.1 struct qt_meta_stringdata_KbManager_t

Definition at line 21 of file moc_kbmanager.cpp.

Collaboration diagram for qt_meta_stringdata_KbManager_t:



Data Fields

QByteArrayData	data[8]	
char	stringdata0[78]	

13.184.2 Macro Definition Documentation

13.184.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbManager_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbmanager.cpp.

13.184.3 Variable Documentation

13.184.3.1 const uint qt_meta_data_KbManager[] [static]

Definition at line 48 of file moc_kbmanager.cpp.

13.184.3.2 const qt_meta_stringdata_KbManager_t qt_meta_stringdata_KbManager [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  0    -  0  * sizeof(
```

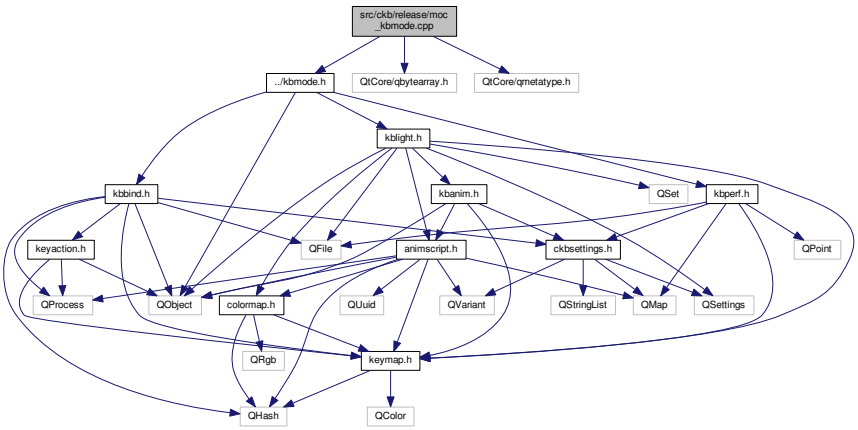
```
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  10    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  22    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 3 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  23    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  27    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  34    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  49    -  6  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbManager_t, stringdata0) +  64    -  7  * sizeof(
      QByteArrayData))   )

    },
    "KbManager\0kbConnected\0\0Kb*\0device\0"
    "kbDisconnected\0versionUpdated\0"
    "scanKeyboards"
}
```

Definition at line 30 of file moc_kbmanager.cpp.

13.185 src/ckb/release/moc_kbmode.cpp File Reference

```
#include "../kbmode.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbmode.cpp:



Data Structures

- struct qt_meta_stringdata_KbMode_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

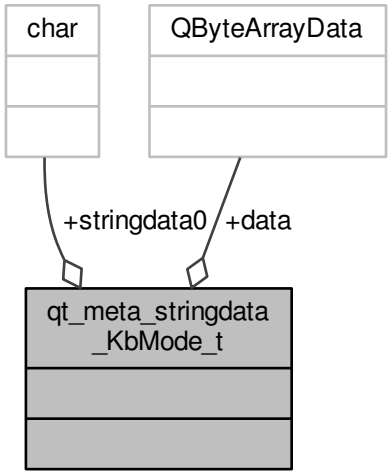
- static const qt_meta_stringdata_KbMode_t qt_meta_stringdata_KbMode
- static const uint qt_meta_data_KbMode [ ]

13.185.1 Data Structure Documentation

13.185.1.1 struct qt_meta_stringdata_KbMode_t

Definition at line 21 of file moc_kbmode.cpp.

Collaboration diagram for qt_meta_stringdata_KbMode_t:



Data Fields

QByteArrayData	data[4]	
char	stringdata0[25]	

13.185.2 Macro Definition Documentation

13.185.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbMode_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbmode.cpp.

13.185.3 Variable Documentation

13.185.3.1 const uint qt_meta_data_KbMode[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       2,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
       1,


       1,    0,   24,    2, 0x06 ,


       3,    0,   25,    2, 0x08 ,


    QMetaType::Void,


    QMetaType::Void,

       0
}
```

Definition at line 42 of file moc_kbmode.cpp.

13.185.3.2 const qt_meta_stringdata_KbMode_t qt_meta_stringdata_KbMode [static]

Initial value:

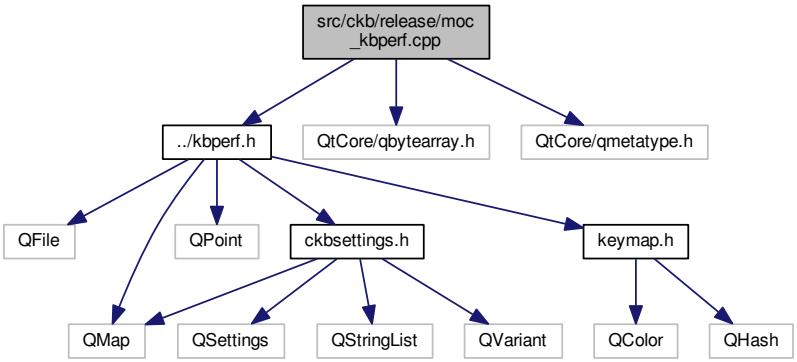
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbMode_t, stringdata0) +  0    -  0  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbMode_t, stringdata0) +  7    -  1  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbMode_t, stringdata0) +  15    -  2  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbMode_t, stringdata0) +  16    -  3  * sizeof(QByteArrayData
      ))   )

    },
    "KbMode\0updated\0\0doUpdate"
}
```

Definition at line 30 of file moc_kbmode.cpp.

13.186 src/ckb/release/moc_kbperf.cpp File Reference

```
#include "../kbperf.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbperf.cpp:



Data Structures

- struct qt_meta_stringdata_KbPerf_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

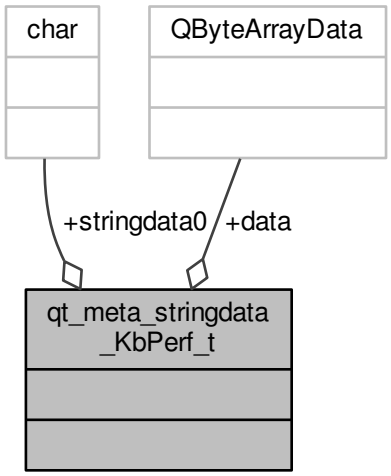
- static const qt_meta_stringdata_KbPerf_t qt_meta_stringdata_KbPerf
- static const uint qt_meta_data_KbPerf [ ]

13.186.1 Data Structure Documentation

13.186.1.1 struct qt_meta_stringdata_KbPerf_t

Definition at line 21 of file moc_kbperf.cpp.

Collaboration diagram for qt_meta_stringdata_KbPerf_t:



Data Fields

QByteArrayData	data[4]	
char	stringdata0[32]	

13.186.2 Macro Definition Documentation

13.186.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbPerf_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbperf.cpp.

13.186.3 Variable Documentation

13.186.3.1 const uint qt_meta_data_KbPerf[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       2,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
```

```
        2,

        1,    0,   24,    2, 0x06 ,
        3,    0,   25,    2, 0x06 ,

    QMetaType::Void,
    QMetaType::Void,

        0
}
```

Definition at line 42 of file moc_kbperf.cpp.

13.186.3.2 const qt_meta_stringdata_KbPerf_t qt_meta_stringdata_KbPerf [static]

Initial value:

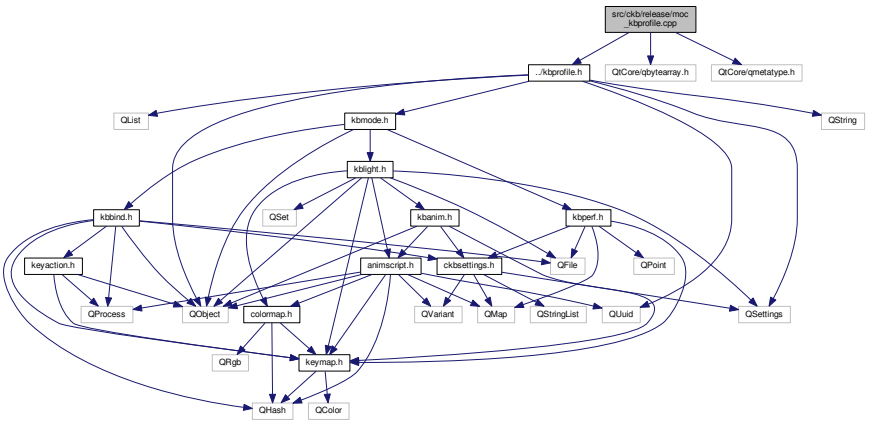
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbPerf_t, stringdata0) +  0    -  0  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbPerf_t, stringdata0) +  7    -  1  * sizeof(QByteArrayData)
      )   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbPerf_t, stringdata0) +  15    -  2  * sizeof(QByteArrayData
      ))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 15 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbPerf_t, stringdata0) +  16    -  3  * sizeof(QByteArrayData
      ))   )

    },
    "KbPerf\0didLoad\0\0settingsUpdated"
}
```

Definition at line 30 of file moc_kbperf.cpp.

13.187 src/ckb/release/moc_kbprofile.cpp File Reference

```
#include "../kbprofile.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbprofile.cpp:



Data Structures

- struct qt_meta_stringdata_KbProfile_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

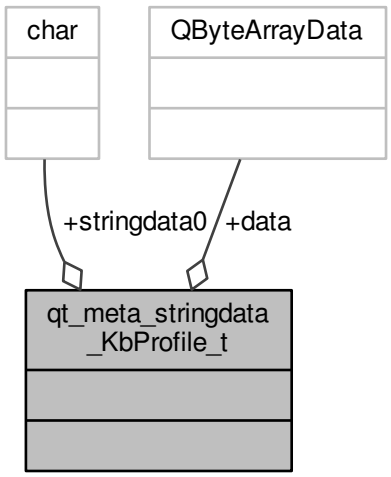
- static const qt_meta_stringdata_KbProfile_t qt_meta_stringdata_KbProfile
- static const uint qt_meta_data_KbProfile []

13.187.1 Data Structure Documentation

13.187.1.1 struct qt_meta_stringdata_KbProfile_t

Definition at line 21 of file moc_kbprofile.cpp.

Collaboration diagram for qt_meta_stringdata_KbProfile_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[10]	

13.187.2 Macro Definition Documentation

13.187.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbProfile_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbprofile.cpp.

13.187.3 Variable Documentation

13.187.3.1 const uint qt_meta_data_KbProfile[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,
       0,

       0
}
```

Definition at line 39 of file moc_kbprofile.cpp.

13.187.3.2 const qt_meta_stringdata_KbProfile_t qt_meta_stringdata_KbProfile [static]

Initial value:

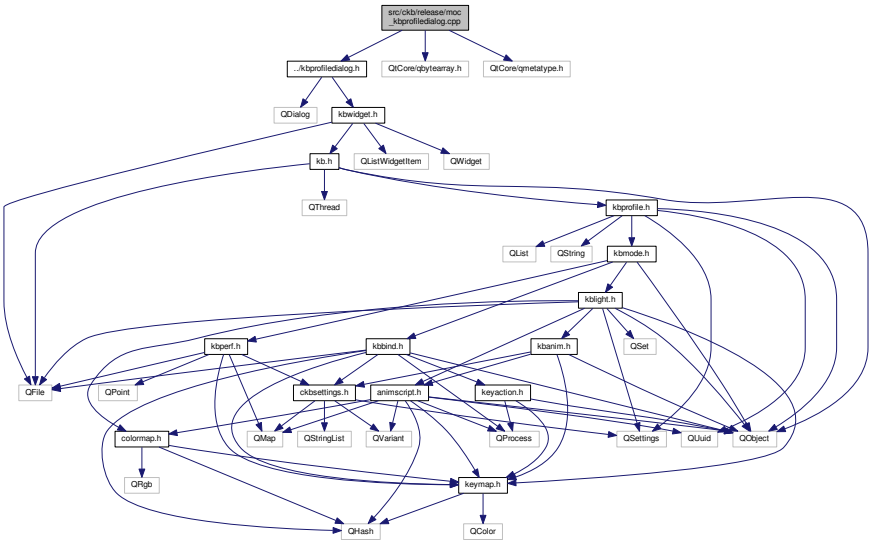
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfile_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "KbProfile"
}
```

Definition at line 30 of file moc_kbprofile.cpp.

13.188 src/ckb/release/moc_kbprofiledialog.cpp File Reference

```
#include "../kbprofiledialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbprofiledialog.cpp:



Data Structures

- struct qt_meta_stringdata_KbProfileDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

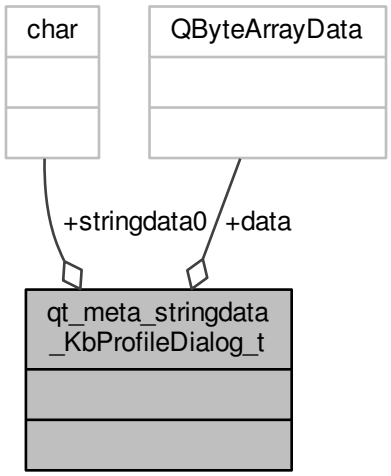
- static const qt_meta_stringdata_KbProfileDialog_t qt_meta_stringdata_KbProfileDialog
- static const uint qt_meta_data_KbProfileDialog []

13.188.1 Data Structure Documentation

13.188.1.1 struct qt_meta_stringdata_KbProfileDialog_t

Definition at line 21 of file moc_kbprofiledialog.cpp.

Collaboration diagram for qt_meta_stringdata_KbProfileDialog_t:



Data Fields

QByteArrayData	data[12]	
char	stringdata0[212]	

13.188.2 Macro Definition Documentation

13.188.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbProfileDialog_t, stringdata0
      ) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbprofiledialog.cpp.

13.188.3 Variable Documentation

13.188.3.1 const uint qt_meta_data_KbProfileDialog[ ] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       5,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,


        1,    0,   39,    2, 0x08 ,
        3,    2,   40,    2, 0x08 ,
        7,    1,   45,    2, 0x08 ,
        9,    1,   48,    2, 0x08 ,
       10,    1,   51,    2, 0x08 ,


    QMetaType::Void,
    QMetaType::Void, 0x80000000 | 4, 0x80000000 | 4,    5,    6,
    QMetaType::Void, 0x80000000 | 4,    8,
    QMetaType::Void, 0x80000000 | 4,    8,
    QMetaType::Void, QMetaType::QPoint,   11,

        0
}
```

Definition at line 56 of file moc_kbprofiledialog.cpp.

13.188.3.2 const qt_meta_stringdata_KbProfileDialog_t qt_meta_stringdata_KbProfileDialog [static]

Initial value:

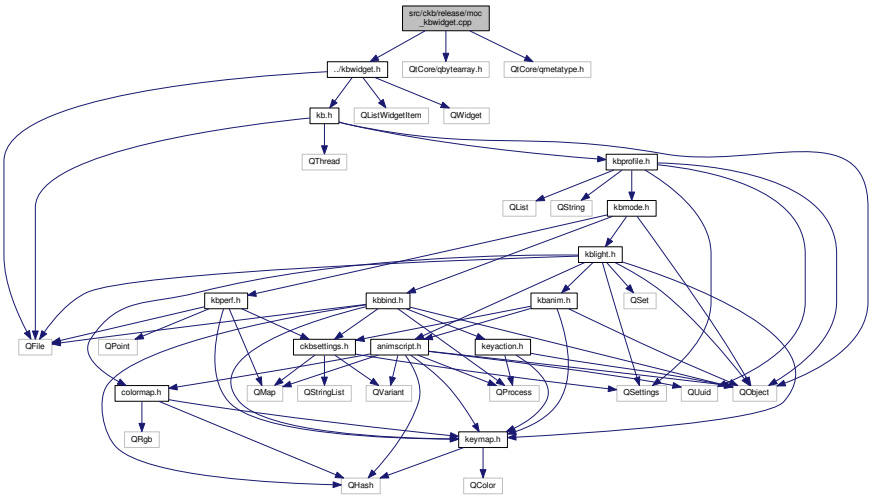
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 15 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  0    -  0  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  16    -  1  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  38    -  2  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 33 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  39    -  3  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  73    -  4  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  90    -  5  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  98    -  6  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 26 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  107    -  7  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  134    -  8  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 26 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  139    -  9  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 41 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  166    -  10  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 3 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KbProfileDialog_t, stringdata0) +  208    -  11  *
      sizeof(QByteArrayData))   )

    },
    "KbProfileDialog\0profileList_reordered\0"
    "\0on_profileList_currentItemChanged\0"
    "QListWidgetItem*\0current\0previous\0"
    "on_profileList_itemClicked\0item\0"
    "on_profileList_itemChanged\0"
    "on_profileList_customContextMenuRequested\0"
    "pos"
}
```

Definition at line 30 of file moc_kbprofiledialog.cpp.

13.189 src/ckb/release/moc_kbwidget.cpp File Reference

```
#include "../kbwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kbwidget.cpp:



Data Structures

- struct qt_meta_stringdata_KbWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

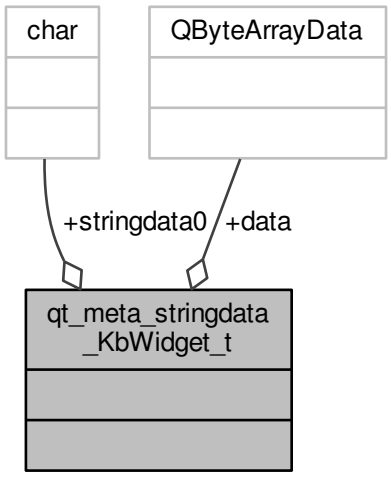
- static const qt_meta_stringdata_KbWidget_t qt_meta_stringdata_KbWidget
- static const uint qt_meta_data_KbWidget [ ]

13.189.1 Data Structure Documentation

13.189.1.1 struct qt_meta_stringdata_KbWidget_t

Definition at line 21 of file moc_kbwidget.cpp.

Collaboration diagram for qt_meta_stringdata_KbWidget_t:



Data Fields

QByteArrayData	data[29]	
char	stringdata0[442]	

13.189.2 Macro Definition Documentation

13.189.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KbWidget_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kbwidget.cpp.

13.189.3 Variable Documentation

13.189.3.1 const uint qt_meta_data_KbWidget[] [static]

Definition at line 80 of file moc_kbwidget.cpp.

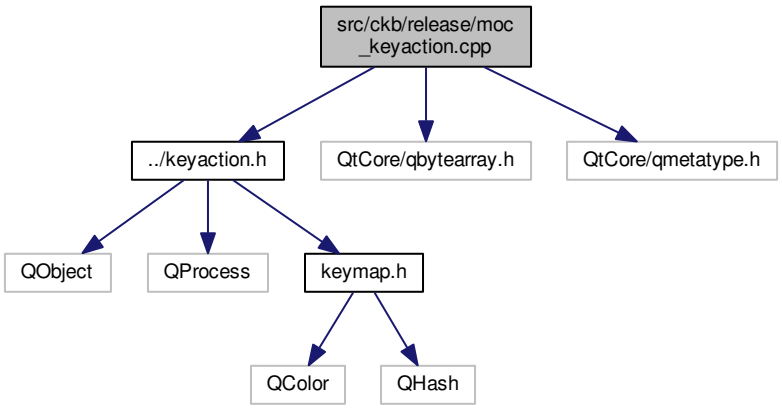
13.189.3.2 const qt_meta_stringdata_KbWidget_t qt_meta_stringdata_KbWidget [static]

Definition at line 30 of file moc_kbwidget.cpp.

13.190 src/ckb/release/moc_keyaction.cpp File Reference

```
#include "../keyaction.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_keyaction.cpp:



Data Structures

- struct qt_meta_stringdata_KeyAction_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

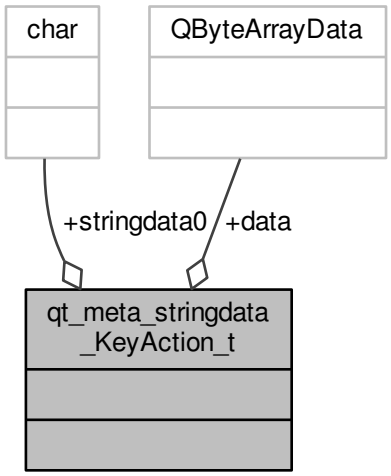
- static const qt_meta_stringdata_KeyAction_t qt_meta_stringdata_KeyAction
- static const uint qt_meta_data_KeyAction [ ]

13.190.1 Data Structure Documentation

13.190.1.1 struct qt_meta_stringdata_KeyAction_t

Definition at line 21 of file moc_keyaction.cpp.

Collaboration diagram for qt_meta_stringdata_KeyAction_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[10]	

13.190.2 Macro Definition Documentation

13.190.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KeyAction_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_keyaction.cpp.

13.190.3 Variable Documentation

13.190.3.1 const uint qt_meta_data_KeyAction[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,
```

```
        0,

        0
}
```

Definition at line 39 of file moc_keyaction.cpp.

13.190.3.2 const qt_meta_stringdata_KeyAction_t qt_meta_stringdata_KeyAction [static]

Initial value:

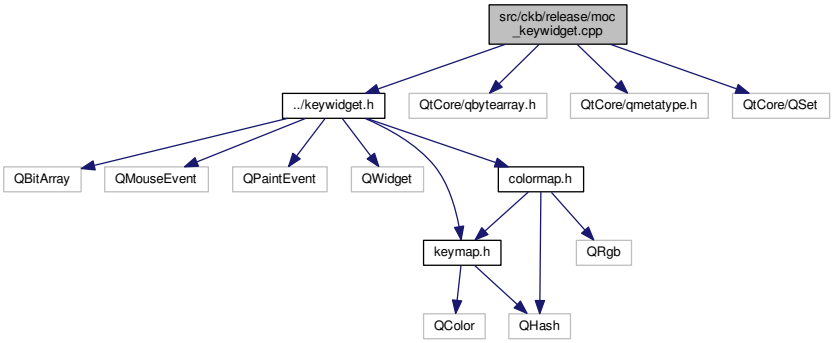
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyAction_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "KeyAction"
}
```

Definition at line 30 of file moc_keyaction.cpp.

13.191 src/ckb/release/moc_keywidget.cpp File Reference

```
#include "../keywidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
#include <QtCore/QSet>
```

Include dependency graph for moc_keywidget.cpp:



Data Structures

- struct qt_meta_stringdata_KeyWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

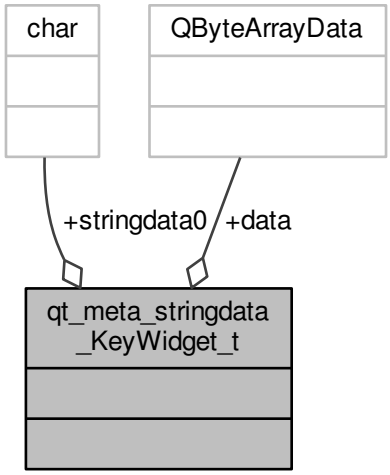
- static const qt_meta_stringdata_KeyWidget_t qt_meta_stringdata_KeyWidget
- static const uint qt_meta_data_KeyWidget [ ]

13.191.1 Data Structure Documentation

13.191.1.1 struct qt_meta_stringdata_KeyWidget_t

Definition at line 22 of file moc_keywidget.cpp.

Collaboration diagram for qt_meta_stringdata_KeyWidget_t:



Data Fields

QByteArrayData	data[10]	
char	stringdata0[118]	

13.191.2 Macro Definition Documentation

13.191.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KeyWidget_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 26 of file moc_keywidget.cpp.

13.191.3 Variable Documentation

13.191.3.1 const uint qt_meta_data_KeyWidget[] [static]

Definition at line 52 of file moc_keywidget.cpp.

13.191.3.2 const qt_meta_stringdata_KeyWidget_t qt_meta_stringdata_KeyWidget [static]

Initial value:

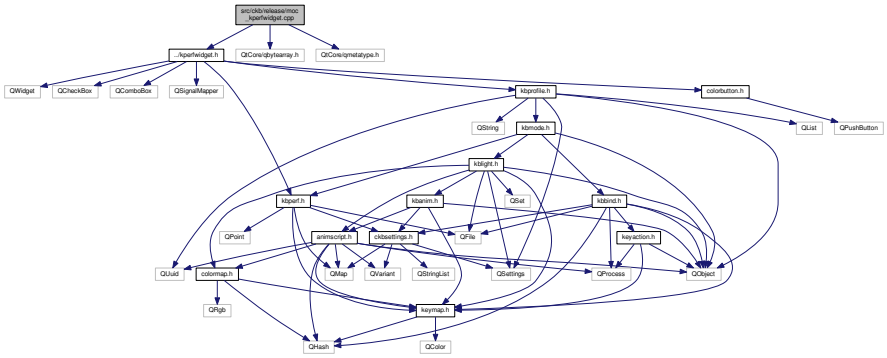
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  10    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  27    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  28    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  37    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 15 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  54    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 8 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  70    -  6  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  79    -  7  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  93    -  8  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 10 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KeyWidget_t, stringdata0) +  107    -  9  * sizeof(
      QByteArrayData))   )

    },
    "KeyWidget\0selectionChanged\0\0selected\0"
    "sidelightToggled\0displayColorMap\0"
    "ColorMap\0newDisplayMap\0QSet<QString>\0"
    "indicators"
}
```

Definition at line 31 of file moc_keywidget.cpp.

13.192 src/ckb/release/moc_kperfwidget.cpp File Reference

```
#include "../kperfwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_kperfwidget.cpp:



Data Structures

- struct qt_meta_stringdata_KPerfWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

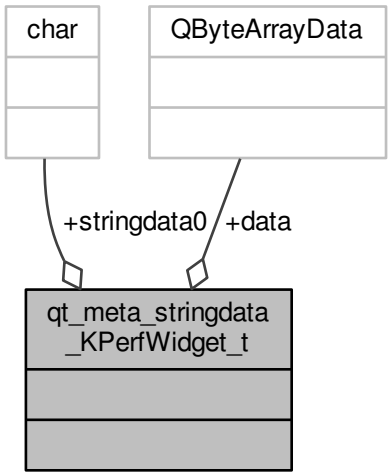
- static const qt_meta_stringdata_KPerfWidget_t qt_meta_stringdata_KPerfWidget
- static const uint qt_meta_data_KPerfWidget [ ]

13.192.1 Data Structure Documentation

13.192.1.1 struct qt_meta_stringdata_KPerfWidget_t

Definition at line 21 of file moc_kperfwidget.cpp.

Collaboration diagram for qt_meta_stringdata_KPerfWidget_t:



Data Fields

QByteArrayData	data[7]	
char	stringdata0[85]	

13.192.2 Macro Definition Documentation

13.192.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_KPerfWidget_t, stringdata0) + ofs
      \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_kperfwidget.cpp.

13.192.3 Variable Documentation

13.192.3.1 const uint qt_meta_data_KPerfWidget[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       3,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,


        1,    1,   29,    2, 0x08 ,
        4,    1,   32,    2, 0x08 ,
        6,    0,   35,    2, 0x08 ,


    QMetaType::Void, QMetaType::Int,    3,
    QMetaType::Void, QMetaType::Int,    5,
    QMetaType::Void,

        0
}
```

Definition at line 47 of file moc_kperfwidget.cpp.

13.192.3.2 const qt_meta_stringdata_KPerfWidget_t qt_meta_stringdata_KPerfWidget [static]

Initial value:

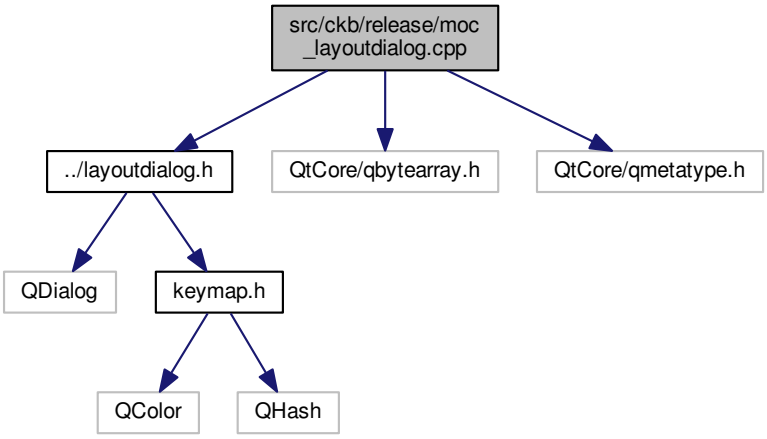
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  12    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  22    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  23    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 28 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  29    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  58    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_KPerfWidget_t, stringdata0) +  63    -  6  * sizeof(
      QByteArrayData))   )

    },
    "KPerfWidget\0uiUpdated\0\0index\0"
    "on_intensityBox_valueChanged\0arg1\0"
    "on_copyButton_clicked"
}
```

Definition at line 30 of file moc_kperfwidget.cpp.

13.193 src/ckb/release/moc_layoutdialog.cpp File Reference

```
#include "../layoutdialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_layoutdialog.cpp:



Data Structures

- struct qt_meta_stringdata_LayoutDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

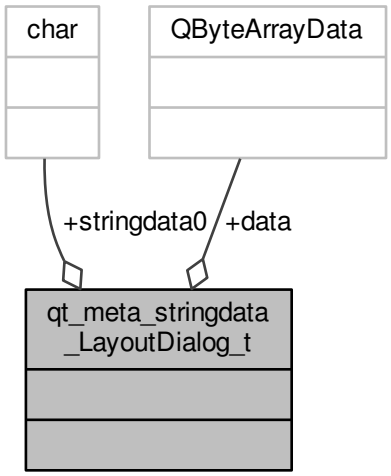
- static const qt_meta_stringdata_LayoutDialog_t qt_meta_stringdata_LayoutDialog
- static const uint qt_meta_data_LayoutDialog [ ]

13.193.1 Data Structure Documentation

13.193.1.1 struct qt_meta_stringdata_LayoutDialog_t

Definition at line 21 of file moc_layoutdialog.cpp.

Collaboration diagram for qt_meta_stringdata_LayoutDialog_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[13]	

13.193.2 Macro Definition Documentation

13.193.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_LayoutDialog_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_layoutdialog.cpp.

13.193.3 Variable Documentation

13.193.3.1 const uint qt_meta_data_LayoutDialog[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,

        0
}
```

Definition at line 39 of file moc_layoutdialog.cpp.

13.193.3.2 const qt_meta_stringdata_LayoutDialog_t qt_meta_stringdata_LayoutDialog [static]

Initial value:

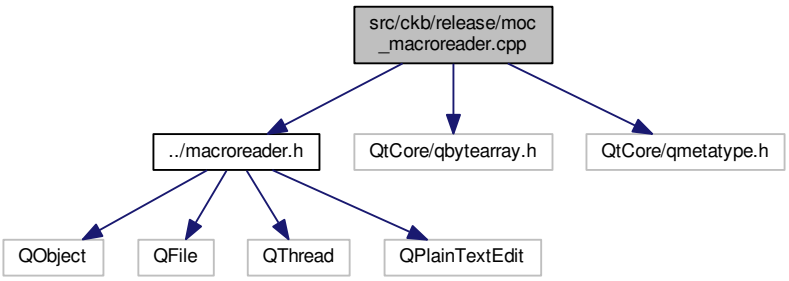
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_LayoutDialog_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "LayoutDialog"
}
```

Definition at line 30 of file moc_layoutdialog.cpp.

13.194 src/ckb/release/moc_macroreader.cpp File Reference

```
#include "../macroreader.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_macroreader.cpp:



Data Structures

- struct qt_meta_stringdata_MacroReaderThread_t
- struct qt_meta_stringdata_MacroReader_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)
- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

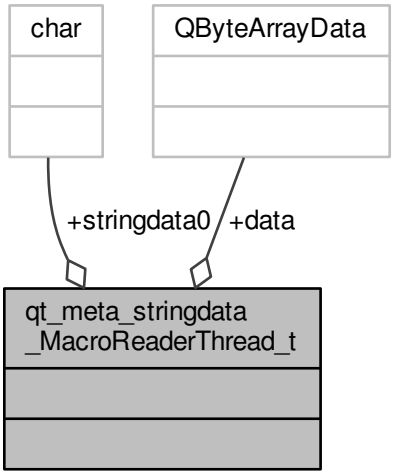
- static const qt_meta_stringdata_MacroReaderThread_t qt_meta_stringdata_MacroReaderThread
- static const uint qt_meta_data_MacroReaderThread []
- static const qt_meta_stringdata_MacroReader_t qt_meta_stringdata_MacroReader
- static const uint qt_meta_data_MacroReader []

13.194.1 Data Structure Documentation

13.194.1.1 struct qt_meta_stringdata_MacroReaderThread_t

Definition at line 21 of file moc_macroreader.cpp.

Collaboration diagram for qt_meta_stringdata_MacroReaderThread_t:



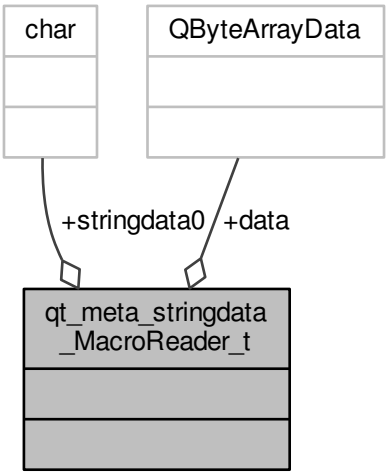
Data Fields

QByteArrayData	data[4]	
char	stringdata0[34]	

13.194.1.2 struct qt_meta_stringdata_MacroReader_t

Definition at line 111 of file moc_macroreader.cpp.

Collaboration diagram for qt_meta_stringdata_MacroReader_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[12]	

13.194.2 Macro Definition Documentation

13.194.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_MacroReaderThread_t,
      stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 115 of file moc_macroreader.cpp.

13.194.2.2 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_MacroReader_t, stringdata0) + ofs
      \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 115 of file moc_macroreader.cpp.

13.194.3 Variable Documentation

13.194.3.1 const uint qt_meta_data_MacroReader[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,
       0,

       0
}
```

Definition at line 129 of file moc_macroreader.cpp.

13.194.3.2 const uint qt_meta_data_MacroReaderThread[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       1,   14,
       0,    0,
       0,    0,
       0,    0,
       0,
       0,


       1,    1,   19,    2, 0x08 ,


    QMetaType::Void, QMetaType::QString,    3,

       0
}
```

Definition at line 42 of file moc_macroreader.cpp.

13.194.3.3 const qt_meta_stringdata_MacroReader_t qt_meta_stringdata_MacroReader [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MacroReader_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "MacroReader"
}
```

Definition at line 120 of file moc_macroreader.cpp.

13.194.3.4 const qt_meta_stringdata_MacroReaderThread_t qt_meta_stringdata_MacroReaderThread [static]

Initial value:

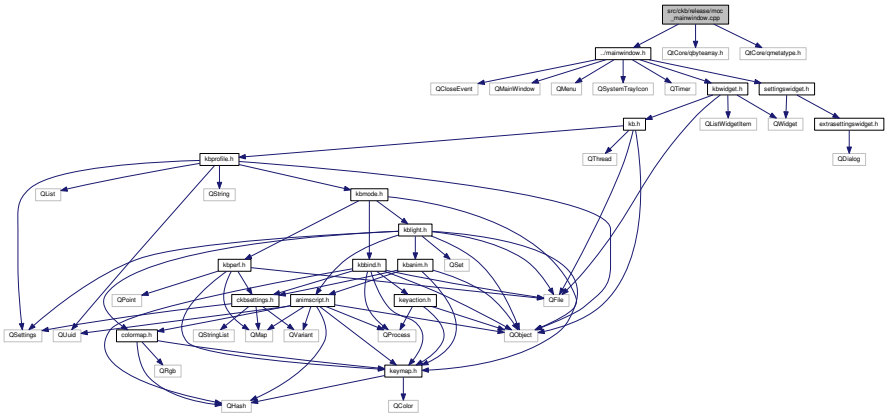
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 17 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MacroReaderThread_t, stringdata0) +  0    -  0  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MacroReaderThread_t, stringdata0) +  18    -  1  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MacroReaderThread_t, stringdata0) +  28    -  2  *
       sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MacroReaderThread_t, stringdata0) +  29    -  3  *
       sizeof(QByteArrayData))   )

    },
    "MacroReaderThread\0readMacro\0\0line"
}
```

Definition at line 30 of file moc_macroreader.cpp.

13.195 src/ckb/release/moc_mainwindow.cpp File Reference

```
#include "../mainwindow.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_mainwindow.cpp:



Data Structures

- struct qt_meta_stringdata_MainWindow_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

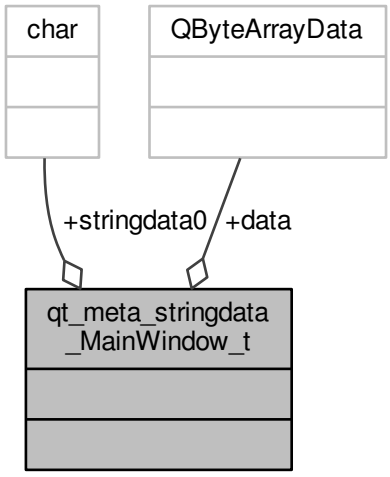
- static const qt_meta_stringdata_MainWindow_t qt_meta_stringdata_MainWindow
- static const uint qt_meta_data_MainWindow []

13.195.1 Data Structure Documentation

13.195.1.1 struct qt_meta_stringdata_MainWindow_t

Definition at line 21 of file moc_mainwindow.cpp.

Collaboration diagram for qt_meta_stringdata_MainWindow_t:



Data Fields

QByteArrayData	data[22]	
char	stringdata0[253]	

13.195.2 Macro Definition Documentation

13.195.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_MainWindow_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_mainwindow.cpp.

13.195.3 Variable Documentation

13.195.3.1 const uint qt_meta_data_MainWindow[] [static]

Definition at line 66 of file moc_mainwindow.cpp.

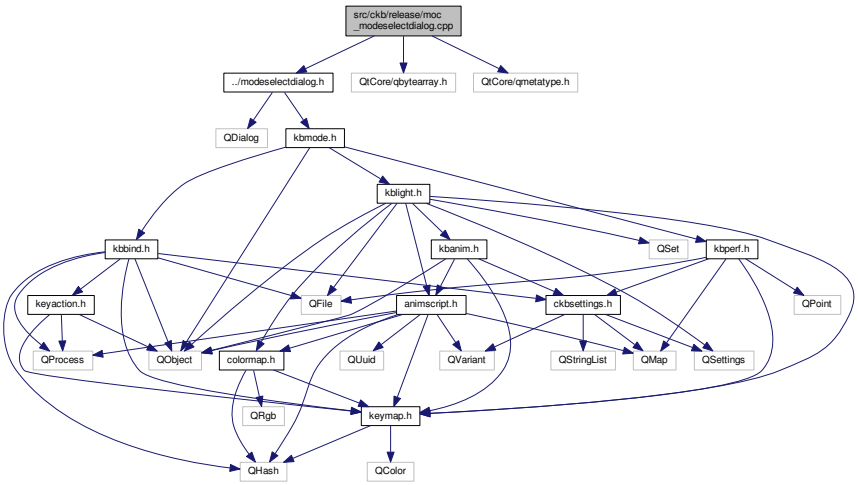
13.195.3.2 const qt_meta_stringdata_MainWindow_t qt_meta_stringdata_MainWindow [static]

Definition at line 30 of file moc_mainwindow.cpp.

13.196 src/ckb/release/moc_modeselectdialog.cpp File Reference

```
#include "../modeselectdialog.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_modeselectdialog.cpp:



Data Structures

- struct qt_meta_stringdata_ModeSelectDialog_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

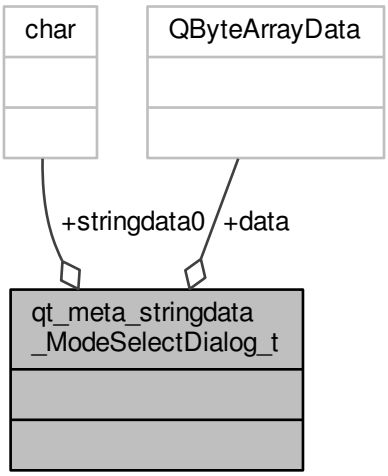
- static const qt_meta_stringdata_ModeSelectDialog_t qt_meta_stringdata_ModeSelectDialog
- static const uint qt_meta_data_ModeSelectDialog []

13.196.1 Data Structure Documentation

13.196.1.1 struct qt_meta_stringdata_ModeSelectDialog_t

Definition at line 21 of file moc_modeselectdialog.cpp.

Collaboration diagram for qt_meta_stringdata_ModeSelectDialog_t:



Data Fields

QByteArrayData	data[4]	
char	stringdata0[67]	

13.196.2 Macro Definition Documentation

13.196.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_ModeSelectDialog_t,
      stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_modeselectdialog.cpp.

13.196.3 Variable Documentation

13.196.3.1 const uint qt_meta_data_ModeSelectDialog[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       2,   14,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,


        1,    0,   24,    2, 0x08 ,
        3,    0,   25,    2, 0x08 ,


    QMetaType::Void,
    QMetaType::Void,

        0
}
```

Definition at line 43 of file moc_modeselectdialog.cpp.

13.196.3.2 const qt_meta_stringdata_ModeSelectDialog_t qt_meta_stringdata_ModeSelectDialog [static]

Initial value:

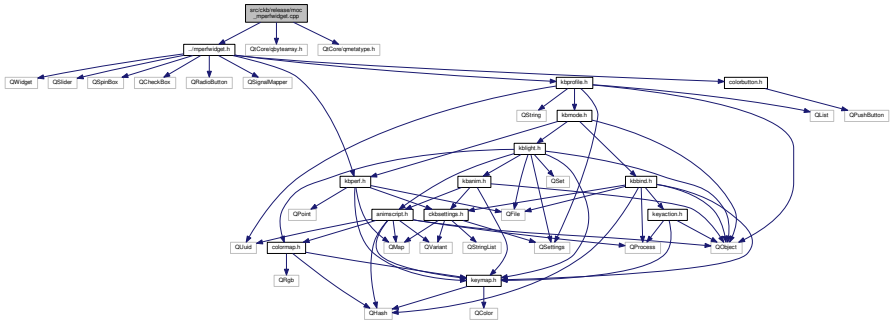
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ModeSelectDialog_t, stringdata0) +  0    -  0  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ModeSelectDialog_t, stringdata0) +  17    -  1  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ModeSelectDialog_t, stringdata0) +  41    -  2  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 24 ,   qptrdiff(offsetof(
      qt_meta_stringdata_ModeSelectDialog_t, stringdata0) +  42    -  3  *
      sizeof(QByteArrayData))   )

    },
    "ModeSelectDialog\0on_selAllButton_clicked\0"
    "\0on_selNoneButton_clicked"
}
```

Definition at line 30 of file moc_modeselectdialog.cpp.

13.197 src/ckb/release/moc_mperfwidget.cpp File Reference

```
#include "../mperfwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_mperfwidget.cpp:



Data Structures

- struct qt_meta_stringdata_MPerfWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

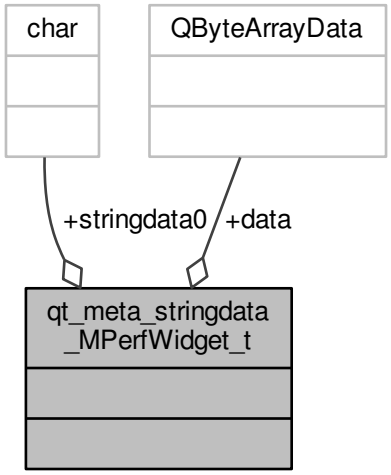
- static const qt_meta_stringdata_MPerfWidget_t qt_meta_stringdata_MPerfWidget
- static const uint qt_meta_data_MPerfWidget []

13.197.1 Data Structure Documentation

13.197.1.1 struct qt_meta_stringdata_MPerfWidget_t

Definition at line 21 of file moc_mperfwidget.cpp.

Collaboration diagram for qt_meta_stringdata_MPerfWidget_t:



Data Fields

QByteArrayData	data[18]	
char	stringdata0[249]	

13.197.2 Macro Definition Documentation

13.197.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_MPerfWidget_t, stringdata0) + ofs
      \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_mperfwidget.cpp.

13.197.3 Variable Documentation

13.197.3.1 const uint qt_meta_data_MPerfWidget[] [static]

Definition at line 62 of file moc_mperfwidget.cpp.

13.197.3.2 const qt_meta_stringdata_MPerfWidget_t qt_meta_stringdata_MPerfWidget [static]

Initial value:

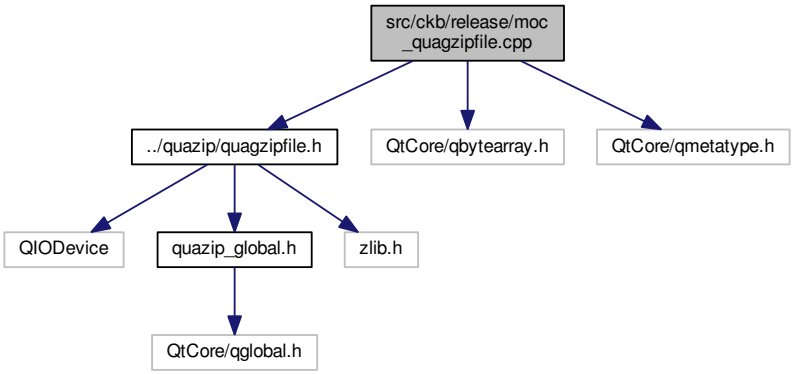
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  12    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  25    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  26    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  32    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  45    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  58    -  6  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  71    -  7  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  83    -  8  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 13 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  95    -  9  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  109    -  10  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  126    -  11  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  134    -  12  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  154    -  13  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  174    -  14  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  198    -  15  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  220    -  16  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_MPerfWidget_t, stringdata0) +  244    -  17  * sizeof(
      QByteArrayData))   )

    },
    "MPerfWidget\0colorClicked\0\0index\0"
    "colorChanged\0sliderXMoved\0sliderYMoved\0"
    "boxXChanged\0boxYChanged\0enableChanged\0"
    "on_xyBox_clicked\0checked\0on_indicBox_clicked\0"
    "on_aSnapBox_clicked\0on_lHeightBox_activated\0"
    "on_copyButton_clicked\0on_spinBox_valueChanged\0"
    "arg1"
}
```

Definition at line 30 of file moc_mperfwidget.cpp.

13.198 src/ckb/release/moc_quagzipfile.cpp File Reference

```
#include "../quazip/quagzipfile.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_quagzipfile.cpp:



Data Structures

- struct qt_meta_stringdata_QuaGzipFile_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

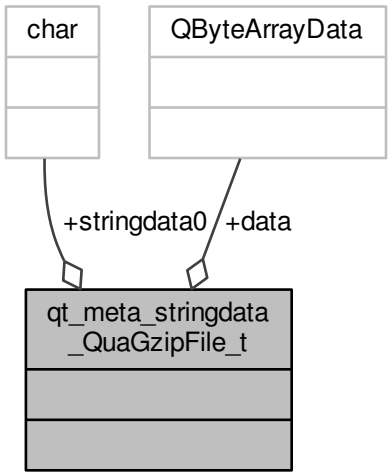
- static const qt_meta_stringdata_QuaGzipFile_t qt_meta_stringdata_QuaGzipFile
- static const uint qt_meta_data_QuaGzipFile []

13.198.1 Data Structure Documentation

13.198.1.1 struct qt_meta_stringdata_QuaGzipFile_t

Definition at line 21 of file moc_quagzipfile.cpp.

Collaboration diagram for qt_meta_stringdata_QuaGzipFile_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[12]	

13.198.2 Macro Definition Documentation

13.198.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_QuaGzipFile_t, stringdata0) + ofs
      \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_quagzipfile.cpp.

13.198.3 Variable Documentation

13.198.3.1 const uint qt_meta_data_QuaGzipFile[ ] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
```

```
        0,
        0,

        0
}
```

Definition at line 39 of file moc_quagzipfile.cpp.

13.198.3.2 const qt_meta_stringdata_QuaGzipFile_t qt_meta_stringdata_QuaGzipFile [static]

Initial value:

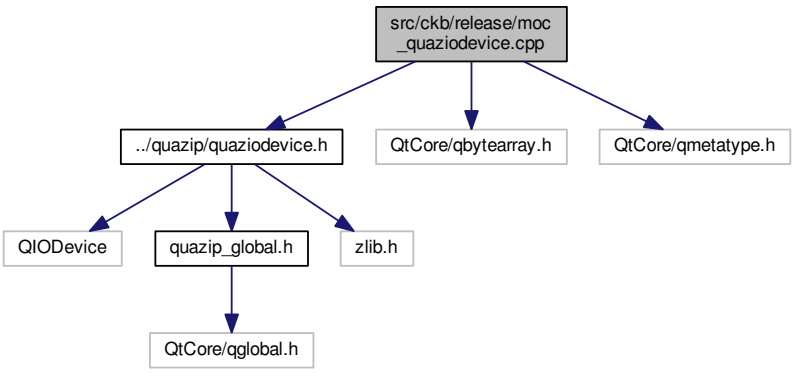
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
      qt_meta_stringdata_QuaGzipFile_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "QuaGzipFile"
}
```

Definition at line 30 of file moc_quagzipfile.cpp.

13.199 src/ckb/release/moc_quaziodevice.cpp File Reference

```
#include "../quazip/quaziodevice.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_quaziodevice.cpp:



Data Structures

- struct qt_meta_stringdata_QuaZIODevice_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

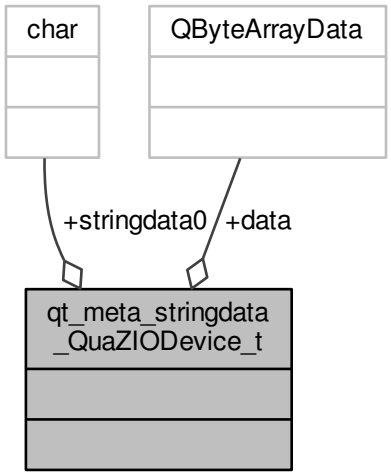
- static const qt_meta_stringdata_QuaZIODevice_t qt_meta_stringdata_QuaZIODevice
- static const uint qt_meta_data_QuaZIODevice []

13.199.1 Data Structure Documentation

13.199.1.1 struct qt_meta_stringdata_QuaZIODevice_t

Definition at line 21 of file moc_quaziodevice.cpp.

Collaboration diagram for qt_meta_stringdata_QuaZIODevice_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[13]	

13.199.2 Macro Definition Documentation

13.199.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_QuaZIODevice_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_quaziodevice.cpp.

13.199.3 Variable Documentation

13.199.3.1 const uint qt_meta_data_QuaZIODevice[] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,
       0,

       0
}
```

Definition at line 39 of file moc_quaziodevice.cpp.

13.199.3.2 const qt_meta_stringdata_QuaZIODevice_t qt_meta_stringdata_QuaZIODevice [static]

Initial value:

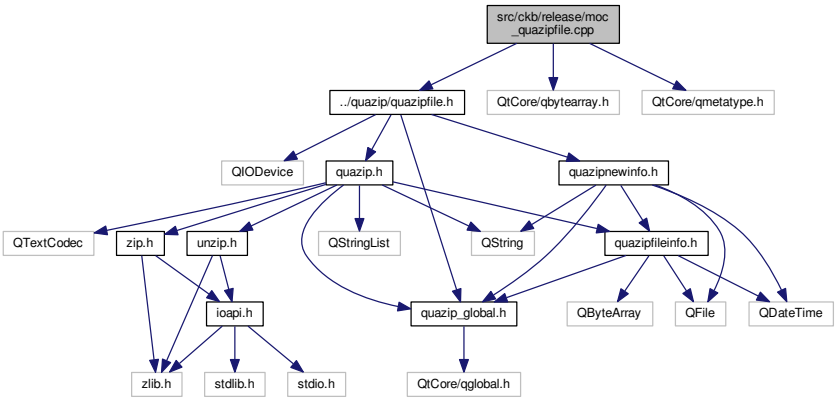
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_QuaZIODevice_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "QuaZIODevice"
}
```

Definition at line 30 of file moc_quaziodevice.cpp.

13.200 src/ckb/release/moc_quazipfile.cpp File Reference

```
#include "../quazip/quazipfile.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_quazipfile.cpp:



Data Structures

- struct qt_meta_stringdata_QuaZipFile_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

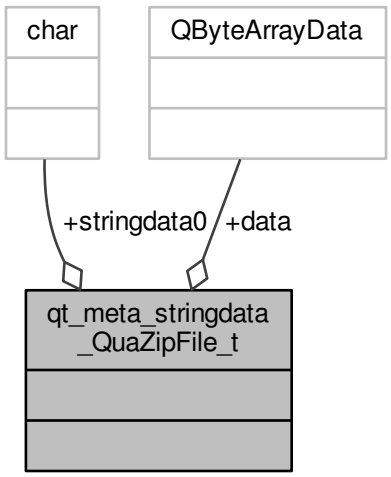
- static const qt_meta_stringdata_QuaZipFile_t qt_meta_stringdata_QuaZipFile
- static const uint qt_meta_data_QuaZipFile []

13.200.1 Data Structure Documentation

13.200.1.1 struct qt_meta_stringdata_QuaZipFile_t

Definition at line 21 of file moc_quazipfile.cpp.

Collaboration diagram for qt_meta_stringdata_QuaZipFile_t:



Data Fields

QByteArrayData	data[1]	
char	stringdata0[11]	

13.200.2 Macro Definition Documentation

13.200.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_QuaZipFile_t, stringdata0) + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_quazipfile.cpp.

13.200.3 Variable Documentation

13.200.3.1 const uint qt_meta_data_QuaZipFile[ ] [static]

Initial value:

```
= {

       7,
       0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,    0,
       0,
       0,

       0
}
```

Definition at line 39 of file moc_quazipfile.cpp.

13.200.3.2 const qt_meta_stringdata_QuaZipFile_t qt_meta_stringdata_QuaZipFile [static]

Initial value:

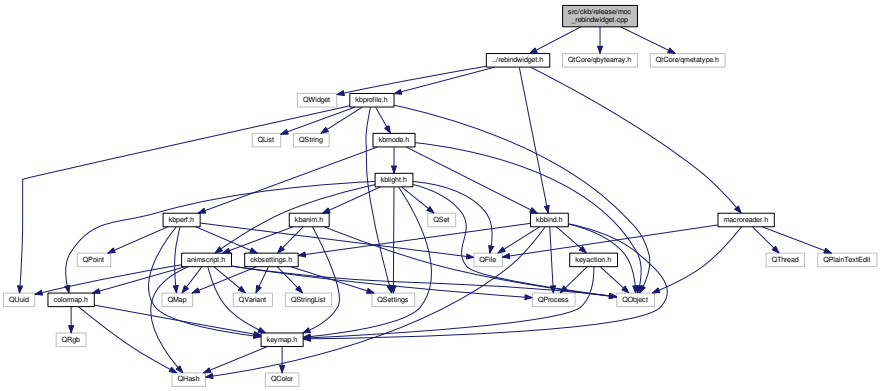
```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 10 ,   qptrdiff(offsetof(
      qt_meta_stringdata_QuaZipFile_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   )

    },
    "QuaZipFile"
}
```

Definition at line 30 of file moc_quazipfile.cpp.

13.201 src/ckb/release/moc_rebindwidget.cpp File Reference

```
#include "../rebindwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_rebindwidget.cpp:



Data Structures

- struct qt_meta_stringdata_RebindWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

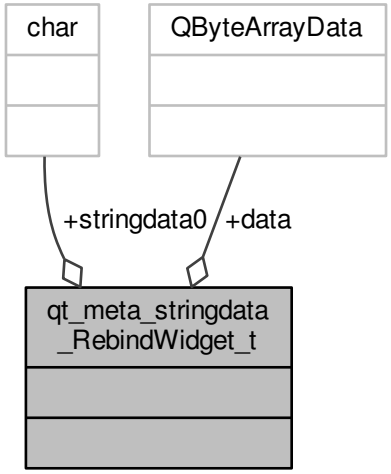
- static const qt_meta_stringdata_RebindWidget_t qt_meta_stringdata_RebindWidget
- static const uint qt_meta_data_RebindWidget [ ]

13.201.1 Data Structure Documentation

13.201.1.1 struct qt_meta_stringdata_RebindWidget_t

Definition at line 21 of file moc_rebindwidget.cpp.

Collaboration diagram for qt_meta_stringdata_RebindWidget_t:



Data Fields

QByteArrayData	data[51]	
char	stringdata0[1179]	

13.201.2 Macro Definition Documentation

13.201.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_RebindWidget_t, stringdata0) +
      ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_rebindwidget.cpp.

13.201.3 Variable Documentation

13.201.3.1 const uint qt_meta_data_RebindWidget[ ] [static]

Definition at line 124 of file moc_rebindwidget.cpp.

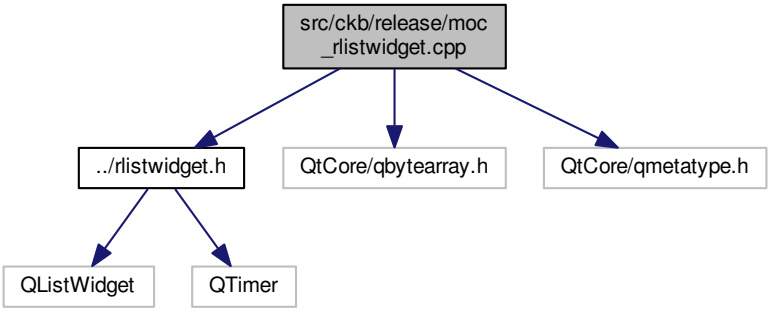
13.201.3.2 const qt_meta_stringdata_RebindWidget_t qt_meta_stringdata_RebindWidget [static]

Definition at line 30 of file moc_rebindwidget.cpp.

13.202 src/ckb/release/moc_rlistwidget.cpp File Reference

```
#include "../rlistwidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_rlistwidget.cpp:



Data Structures

- struct qt_meta_stringdata_RListWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

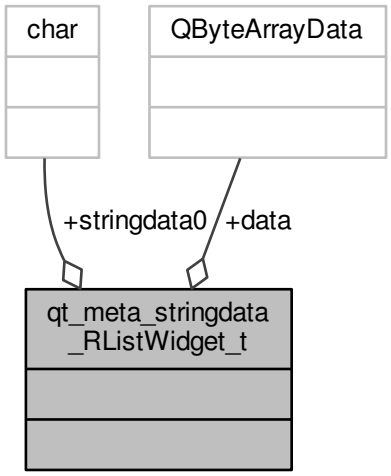
- static const qt_meta_stringdata_RListWidget_t qt_meta_stringdata_RListWidget
- static const uint qt_meta_data_RListWidget []

13.202.1 Data Structure Documentation

13.202.1.1 struct qt_meta_stringdata_RListWidget_t

Definition at line 21 of file moc_rlistwidget.cpp.

Collaboration diagram for qt_meta_stringdata_RListWidget_t:



Data Fields

QByteArrayData	data[8]	
char	stringdata0[71]	

13.202.2 Macro Definition Documentation

13.202.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_RListWidget_t, stringdata0) + ofs
      \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_rlistwidget.cpp.

13.202.3 Variable Documentation

13.202.3.1 const uint qt_meta_data_RListWidget[] [static]

Definition at line 47 of file moc_rlistwidget.cpp.

13.202.3.2 const qt_meta_stringdata_RListWidget_t qt_meta_stringdata_RListWidget [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 11 ,   qptrdiff(offsetof(
```

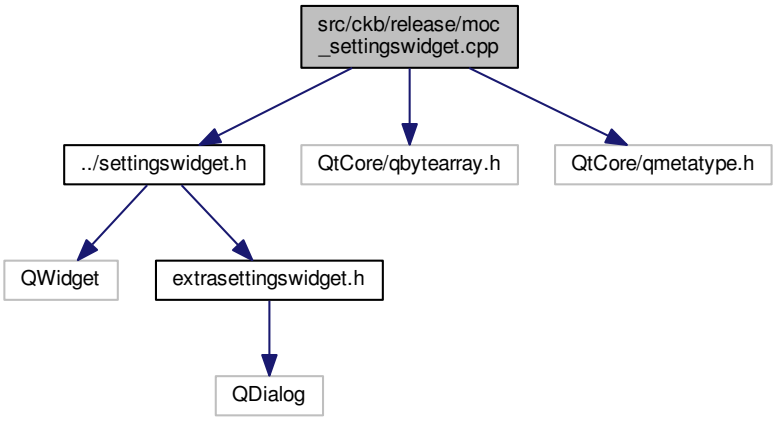
```
      qt_meta_stringdata_RListWidget_t, stringdata0) +  0    -  0  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 12 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  12    -  1  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  25    -  2  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 9 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  26    -  3  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  36    -  4  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  42    -  5  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 4 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  59    -  6  * sizeof(
      QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 6 ,   qptrdiff(offsetof(
      qt_meta_stringdata_RListWidget_t, stringdata0) +  64    -  7  * sizeof(
      QByteArrayData))   )

    },
    "RListWidget\0orderChanged\0\0timerTick\0"
    "enter\0QListWidgetItem*\0item\0change"
}
```

Definition at line 30 of file moc_rlistwidget.cpp.

13.203 src/ckb/release/moc_settingswidget.cpp File Reference

```
#include "../settingswidget.h"
#include <QtCore/qbytearray.h>
#include <QtCore/qmetatype.h>
```

Include dependency graph for moc_settingswidget.cpp:



Data Structures

- struct qt_meta_stringdata_SettingsWidget_t

Macros

- #define QT_MOC_LITERAL(idx, ofs, len)

Variables

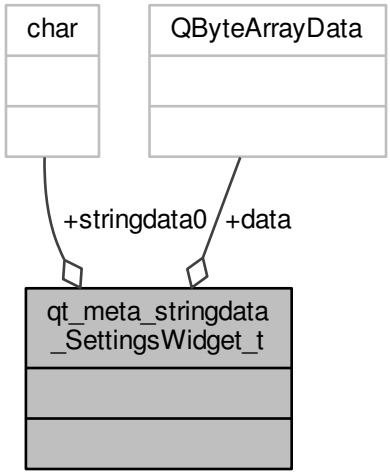
- static const qt_meta_stringdata_SettingsWidget_t qt_meta_stringdata_SettingsWidget
- static const uint qt_meta_data_SettingsWidget []

13.203.1 Data Structure Documentation

13.203.1.1 struct qt_meta_stringdata_SettingsWidget_t

Definition at line 21 of file moc_settingswidget.cpp.

Collaboration diagram for qt_meta_stringdata_SettingsWidget_t:



Data Fields

QByteArrayData	data[15]	
char	stringdata0[264]	

13.203.2 Macro Definition Documentation

13.203.2.1 #define QT_MOC_LITERAL(*idx, ofs, len*)

Value:

```
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET(len, \
    qptrdiff(offsetof(qt_meta_stringdata_SettingsWidget_t, stringdata0)
      + ofs \
        - idx * sizeof(QByteArrayData)) \
    )
```

Definition at line 25 of file moc_settingswidget.cpp.

13.203.3 Variable Documentation

13.203.3.1 const uint qt_meta_data_SettingsWidget[] [static]

Definition at line 59 of file moc_settingswidget.cpp.

13.203.3.2 const qt_meta_stringdata_SettingsWidget_t qt_meta_stringdata_SettingsWidget [static]

Initial value:

```
= {
    {
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 14 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  0    -  0  * sizeof
      (QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  15    -  1  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 0 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  37    -  2  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  38    -  3  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 5 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  59    -  4  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 21 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  65    -  5  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  87    -  6  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  108    -  7  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 19 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  128    -  8  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 20 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  148    -  9  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 7 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  169    -  10  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 23 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  177    -  11  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 22 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  201    -  12  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 16 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  224    -  13  *
      sizeof(QByteArrayData))   ),
Q_STATIC_BYTE_ARRAY_DATA_HEADER_INITIALIZER_WITH_OFFSET( 22 ,   qptrdiff(offsetof(
      qt_meta_stringdata_SettingsWidget_t, stringdata0) +  241    -  14  *
      sizeof(QByteArrayData))   )

    },
    "SettingsWidget\0on_pushButton_clicked\0"
    "\0on_capsBox_activated\0index\0"
    "on_shiftBox_activated\0on_ctrlBox_activated\0"
    "on_altBox_activated\0on_winBox_activated\0"
    "on_autoFWBox_clicked\0checked\0"
    "on_loginItemBox_clicked\0on_layoutBox_activated\0"
    "showLayoutDialog\0on_extraButton_clicked"
}
```

Definition at line 30 of file moc_settingswidget.cpp.

13.204 src/ckb/release/qrc_binary.cpp File Reference

Macros

- #define QT_RCC_PREPEND_NAMESPACE(name) name
- #define QT_RCC_MANGLE_NAMESPACE(name) name

Functions

- bool qRegisterResourceData (int, const unsigned char *, const unsigned char *, const unsigned char *)
- bool qUnregisterResourceData (int, const unsigned char *, const unsigned char *, const unsigned char *)
- int qInitResources_binary ()
- int qCleanupResources_binary ()

Variables

- static const unsigned char qt_resource_data []
- static const unsigned char qt_resource_name []
- static const unsigned char qt_resource_struct []

13.204.1 Macro Definition Documentation

13.204.1.1 #define QT_RCC_MANGLE_NAMESPACE(*name*) name

Definition at line 147 of file qrc_binary.cpp.

13.204.1.2 #define QT_RCC_PREPEND_NAMESPACE(*name*) name

Definition at line 146 of file qrc_binary.cpp.

Referenced by qCleanupResources_binary(), and qInitResources_binary().

13.204.2 Function Documentation

13.204.2.1 int qCleanupResources_binary ( )

Definition at line 171 of file qrc_binary.cpp.

References QT_RCC_PREPEND_NAMESPACE, qt_resource_data, qt_resource_name, qt_resource_struct, and qUnregisterResourceData().

```
172 {
173     QT_RCC_PREPEND_NAMESPACE(qUnregisterResourceData)
174         (0x01, qt_resource_struct, qt_resource_name,
    qt_resource_data);
175     return 1;
176 }
```

Here is the call graph for this function:

13.204.2.2 int qInitResources_binary ( )

Definition at line 163 of file qrc_binary.cpp.

References qRegisterResourceData(), QT_RCC_PREPEND_NAMESPACE, qt_resource_data, qt_resource_name, and qt_resource_struct.

```
164 {
165     QT_RCC_PREPEND_NAMESPACE(qRegisterResourceData)
166         (0x01, qt_resource_struct, qt_resource_name,
      qt_resource_data);
167     return 1;
168 }
```

Here is the call graph for this function:

13.204.2.3 bool qRegisterResourceData (int , const unsigned char * , const unsigned char * , const unsigned char *)

Referenced by qInitResources_binary().

Here is the caller graph for this function:

13.204.2.4 bool qUnregisterResourceData (int , const unsigned char * , const unsigned char * , const unsigned char *)

Referenced by qCleanupResources_binary().

Here is the caller graph for this function:

13.204.3 Variable Documentation

13.204.3.1 const unsigned char qt_resource_data[] [static]

Definition at line 9 of file qrc_binary.cpp.

Referenced by qCleanupResources_binary(), and qInitResources_binary().

13.204.3.2 const unsigned char qt_resource_name[] [static]

Initial value:

```
= {

  0x0,0x3,
  0x0,0x0,0x68,0xfe,
  0x0,0x62,
  0x0,0x69,0x0,0x6e,

  0x0,0x10,
  0x9,0x4b,0x25,0x27,
  0x0,0x63,
  0x0,0x6b,0x0,0x62,0x0,0x2d,0x0,0x6e,0x0,0x65,0x0,0x78,0x0,0x74,0x0,0x2d,0x0,0x6b,0x0,0x65,0x0,0x79,0x0,
       0x2e,0x0,0x67,0x0,0x70,0x0,0x67,

}
```

Definition at line 114 of file qrc_binary.cpp.

Referenced by qCleanupResources_binary(), and qInitResources_binary().

13.204.3.3 const unsigned char qt_resource_struct[] [static]

Initial value:

```
= {

  0x0,0x0,0x0,0x0,0x0,0x2,0x0,0x0,0x0,0x1,0x0,0x0,0x0,0x1,

  0x0,0x0,0x0,0x0,0x0,0x2,0x0,0x0,0x0,0x1,0x0,0x0,0x0,0x2,

  0x0,0x0,0x0,0xc,0x0,0x0,0x0,0x0,0x0,0x1,0x0,0x0,0x0,0x0,

}
```

Definition at line 128 of file qrc_binary.cpp.

Referenced by qCleanupResources_binary(), and qInitResources_binary().

13.205 src/ckb/release/qrc_image.cpp File Reference

Macros

- #define QT_RCC_PREPEND_NAMESPACE(name) name
- #define QT_RCC_MANGLE_NAMESPACE(name) name

Functions

- bool qRegisterResourceData (int, const unsigned char *, const unsigned char *, const unsigned char *)
- bool qUnregisterResourceData (int, const unsigned char *, const unsigned char *, const unsigned char *)
- int qInitResources_image ()
- int qCleanupResources_image ()

Variables

- static const unsigned char qt_resource_data []
- static const unsigned char qt_resource_name []
- static const unsigned char qt_resource_struct []

13.205.1 Macro Definition Documentation

13.205.1.1 #define QT_RCC_MANGLE_NAMESPACE(*name*) name

Definition at line 14160 of file qrc_image.cpp.

13.205.1.2 #define QT_RCC_PREPEND_NAMESPACE(*name*) name

Definition at line 14159 of file qrc_image.cpp.

Referenced by qCleanupResources_image(), and qInitResources_image().

13.205.2 Function Documentation

13.205.2.1 int qCleanupResources_image ( )

Definition at line 14184 of file qrc_image.cpp.

References QT_RCC_PREPEND_NAMESPACE, qt_resource_data, qt_resource_name, qt_resource_struct, and qUnregisterResourceData().

```
14185 {
14186     QT_RCC_PREPEND_NAMESPACE(qUnregisterResourceData)
14187         (0x01, qt_resource_struct, qt_resource_name,
      qt_resource_data);
14188     return 1;
14189 }
```

Here is the call graph for this function:

13.205.2.2 int qInitResources_image ( )

Definition at line 14176 of file qrc_image.cpp.

References qRegisterResourceData(), QT_RCC_PREPEND_NAMESPACE, qt_resource_data, qt_resource_name, and qt_resource_struct.

```
14177 {
14178     QT_RCC_PREPEND_NAMESPACE(qRegisterResourceData)
14179         (0x01, qt_resource_struct, qt_resource_name,
      qt_resource_data);
14180     return 1;
14181 }
```

Here is the call graph for this function:

13.205.2.3 bool qRegisterResourceData (int , const unsigned char * , const unsigned char * , const unsigned char *)

Referenced by qInitResources_image().

Here is the caller graph for this function:

13.205.2.4 bool qUnregisterResourceData (int , const unsigned char * , const unsigned char * , const unsigned char *)

Referenced by qCleanupResources_image().

Here is the caller graph for this function:

13.205.3 Variable Documentation

13.205.3.1 const unsigned char qt_resource_data[] [static]

Definition at line 9 of file qrc_image.cpp.

Referenced by qCleanupResources_image(), and qInitResources_image().

13.205.3.2 const unsigned char qt_resource_name[] [static]

Definition at line 14007 of file qrc_image.cpp.

Referenced by qCleanupResources_image(), and qInitResources_image().

13.205.3.3 const unsigned char qt_resource_struct[] [static]

Definition at line 14109 of file qrc_image.cpp.

Referenced by qCleanupResources_image(), and qInitResources_image().

13.206 src/ckb/release/qrc_text.cpp File Reference

Macros

- #define QT_RCC_PREPEND_NAMESPACE(name) name
- #define QT_RCC_MANGLE_NAMESPACE(name) name

Functions

- bool qRegisterResourceData (int, const unsigned char *, const unsigned char *, const unsigned char *)
- bool qUnregisterResourceData (int, const unsigned char *, const unsigned char *, const unsigned char *)
- int qInitResources_text ()
- int qCleanupResources_text ()

Variables

- static const unsigned char qt_resource_data []
- static const unsigned char qt_resource_name []
- static const unsigned char qt_resource_struct []

13.206.1 Macro Definition Documentation

13.206.1.1 #define QT_RCC_MANGLE_NAMESPACE(*name*) name

Definition at line 350 of file qrc_text.cpp.

13.206.1.2 #define QT_RCC_PREPEND_NAMESPACE(*name*) name

Definition at line 349 of file qrc_text.cpp.

Referenced by qCleanupResources_text(), and qInitResources_text().

13.206.2 Function Documentation

13.206.2.1 int qCleanupResources_text ( )

Definition at line 374 of file qrc_text.cpp.

References QT_RCC_PREPEND_NAMESPACE, qt_resource_data, qt_resource_name, qt_resource_struct, and qUnregisterResourceData().

```
375 {
376     QT_RCC_PREPEND_NAMESPACE(qUnregisterResourceData)
377         (0x01, qt_resource_struct, qt_resource_name,
      qt_resource_data);
378     return 1;
379 }
```

Here is the call graph for this function:

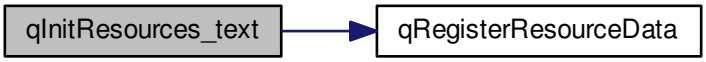
13.206.2.2 int qInitResources_text ( )

Definition at line 366 of file qrc_text.cpp.

References qRegisterResourceData(), QT_RCC_PREPEND_NAMESPACE, qt_resource_data, qt_resource_name, and qt_resource_struct.

```
367 {
368     QT_RCC_PREPEND_NAMESPACE(qRegisterResourceData)
369         (0x01, qt_resource_struct, qt_resource_name,
      qt_resource_data);
370     return 1;
371 }
```

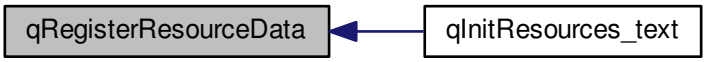
Here is the call graph for this function:



13.206.2.3 bool qRegisterResourceData (int , const unsigned char * , const unsigned char * , const unsigned char *)

Referenced by qInitResources_text().

Here is the caller graph for this function:



13.206.2.4 bool qUnregisterResourceData (int , const unsigned char * , const unsigned char * , const unsigned char *)

Referenced by qCleanupResources_text().

Here is the caller graph for this function:

13.206.3 Variable Documentation

13.206.3.1 const unsigned char qt_resource_data[] [static]

Definition at line 9 of file qrc_text.cpp.

Referenced by qCleanupResources_text(), and qInitResources_text().

13.206.3.2 const unsigned char qt_resource_name[] [static]

Initial value:

```
= {

  0x0,0x3,
  0x0,0x0,0x7b,0xf4,
  0x0,0x74,
  0x0,0x78,0x0,0x74,

  0x0,0xb,
  0xa,0x1e,0x1,0xe0,
  0x0,0x63,
  0x0,0x6b,0x0,0x62,0x0,0x2e,0x0,0x64,0x0,0x65,0x0,0x73,0x0,0x6b,0x0,0x74,0x0,0x6f,0x0,0x70,

  0x0,0x10,
  0xe,0xac,0xaf,0x46,
  0x0,0x64,
  0x0,0x65,0x0,0x6d,0x0,0x6f,0x0,0x70,0x0,0x72,0x0,0x6f,0x0,0x66,0x0,0x69,0x0,0x6c,0x0,0x65,0x0,0x2e,0x0,
       0x63,0x0,0x6f,0x0,0x6e,0x0,0x66,

  0x0,0x11,
  0x0,0xd4,0x25,0xa4,
  0x0,0x63,
  0x0,0x6f,0x0,0x6d,0x0,0x2e,0x0,0x63,0x0,0x6b,0x0,0x62,0x0,0x2e,0x0,0x63,0x0,0x6b,0x0,0x62,0x0,0x2e,0x0,
       0x70,0x0,0x6c,0x0,0x69,0x0,0x73,0x0,0x74,


}
```

Definition at line 302 of file qrc_text.cpp.

Referenced by qCleanupResources_text(), and qInitResources_text().

13.206.3.3 const unsigned char qt_resource_struct[] [static]

Initial value:

```
= {

  0x0,0x0,0x0,0x0,0x0,0x2,0x0,0x0,0x0,0x1,0x0,0x0,0x0,0x1,
```

```
  0x0,0x0,0x0,0x0,0x0,0x2,0x0,0x0,0x0,0x3,0x0,0x0,0x0,0x2,
  0x0,0x0,0x0,0x4e,0x0,0x0,0x0,0x0,0x0,0x1,0x0,0x0,0xf,0x74,
  0x0,0x0,0x0,0xc,0x0,0x0,0x0,0x0,0x0,0x1,0x0,0x0,0x0,0x0,
  0x0,0x0,0x0,0x28,0x0,0x1,0x0,0x0,0x0,0x1,0x0,0x0,0x0,0xc6,
}
```

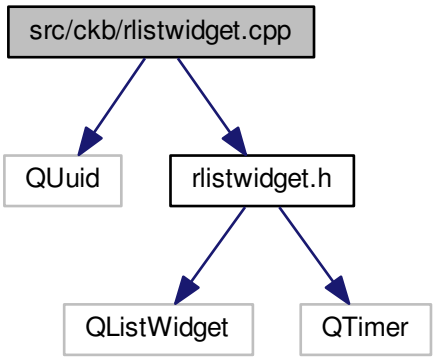
Definition at line 327 of file qrc_text.cpp.

Referenced by qCleanupResources_text(), and qInitResources_text().

13.207 src/ckb/rlistwidget.cpp File Reference

```
#include <QUuid>
#include "rlistwidget.h"
```

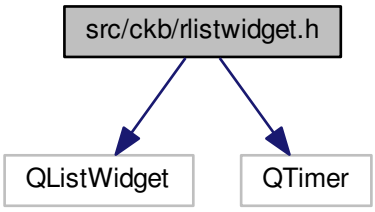
Include dependency graph for rlistwidget.cpp:



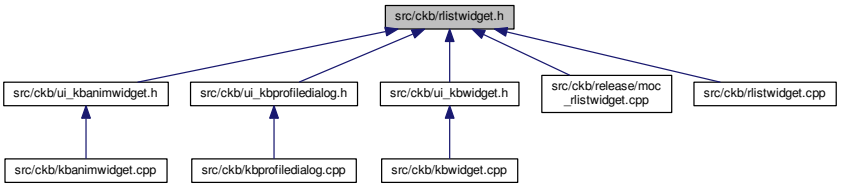
13.208 src/ckb/rlistwidget.h File Reference

```
#include <QListWidget>
#include <QTimer>
```

Include dependency graph for rlistwidget.h:



This graph shows which files directly or indirectly include this file:



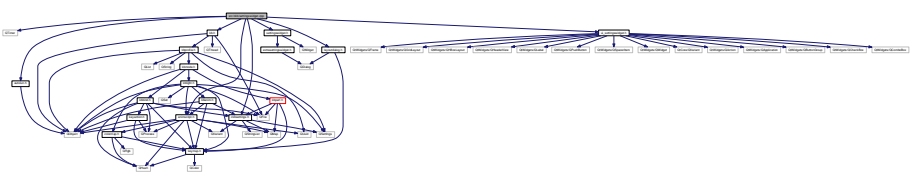
Data Structures

- class RListWidget

13.209 src/ckb/settingswidget.cpp File Reference

```
#include <QTimer>
#include "animscript.h"
#include "autorun.h"
#include "ckbsettings.h"
#include "kb.h"
#include "layoutdialog.h"
#include "settingswidget.h"
#include "ui_settingswidget.h"
```

Include dependency graph for settingswidget.cpp:



Functions

- static QString right (const QString &left)

Variables

- QString devpath

  Device path base ("/dev/input/ckb" or "/var/run/ckb")
- static QStringList modKeys
- static QStringList modNames

13.209.1 Function Documentation

13.209.1.1 static QString right (const QString & *left*) [inline], [static]

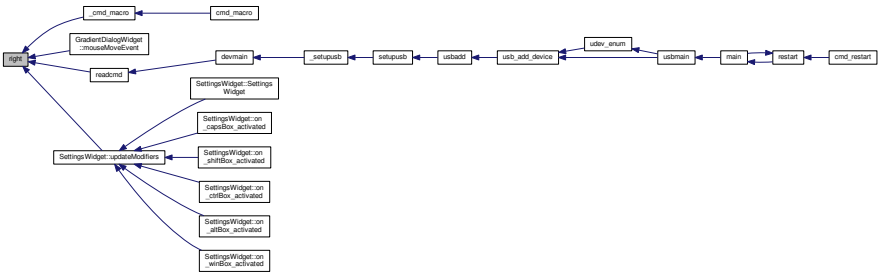
Definition at line 118 of file settingswidget.cpp.

References left.

Referenced by _cmd_macro(), GradientDialogWidget::mouseMoveEvent(), readcmd(), and SettingsWidget-::updateModifiers().

```
118                                                 {
119     if(left[0] != 'l')
120         return left;
121     return "r" + left.mid(1);
122 }
```

Here is the caller graph for this function:



13.209.2 Variable Documentation

13.209.2.1 QString devpath

Definition at line 4 of file kbmanager.cpp.

Referenced by _mkdevpath(), _mknotifynode(), _rmnotifynode(), _setupusb(), _updateconnected(), closeusb(), main(), mkfwnode(), os_inputmain(), os_setupusb(), rmdevpath(), and KbManager::scanKeyboards().

13.209.2.2 QStringList modKeys [static]

Definition at line 13 of file settingswidget.cpp.

Referenced by SettingsWidget::SettingsWidget(), and SettingsWidget::updateModifiers().

13.209.2.3 QStringList modNames [static]

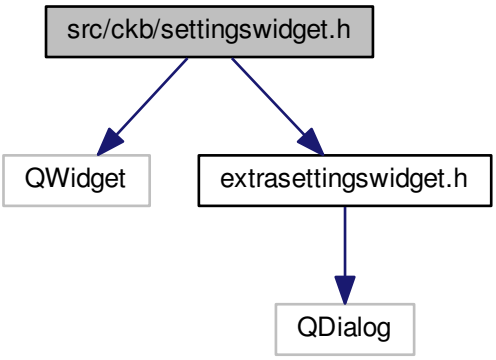
Definition at line 13 of file settingswidget.cpp.

Referenced by SettingsWidget::SettingsWidget().

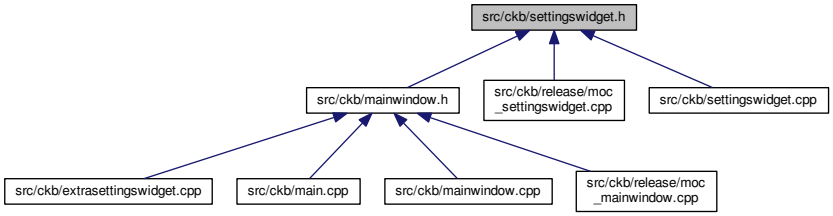
13.210 src/ckb/settingswidget.h File Reference

```
#include <QWidget>
#include "extrasettingswidget.h"
```

Include dependency graph for settingswidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class SettingsWidget

Namespaces

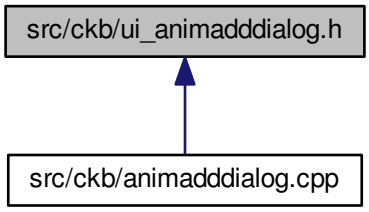
- Ui

13.211 src/ckb/ui_animadddialog.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QSpacerItem>
```

Include dependency graph for ui_animadddialog.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_AnimAddDialog
- class Ui::AnimAddDialog

Namespaces

- Ui

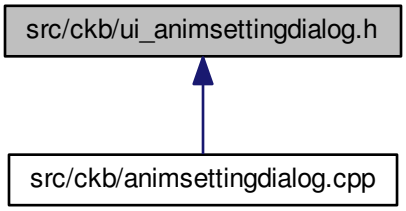
13.212 src/ckb/ui_animsettingdialog.h File Reference

```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QDoubleSpinBox>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QLineEdit>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QTabWidget>
#include <QtWidgets/QWidget>
```

Include dependency graph for ui_animsettingdialog.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_AnimSettingDialog
- class Ui::AnimSettingDialog

Namespaces

- Ui

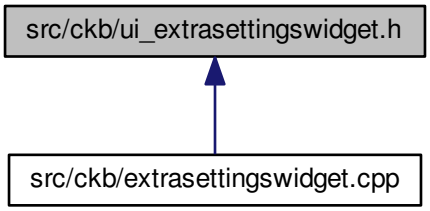
13.213 src/ckb/ui_extrasettingswidget.h File Reference

```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QSpinBox>
#include <QtWidgets/QWidget>
```

Include dependency graph for ui_extrasettingswidget.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_ExtraSettingsWidget
- class Ui::ExtraSettingsWidget

Namespaces

- Ui

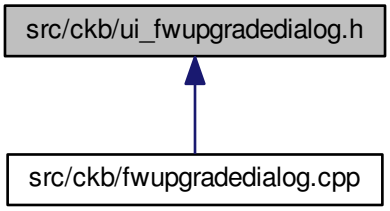
13.214 src/ckb/ui_fwupgradedialog.h File Reference

```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QDialog>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QProgressBar>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
```

Include dependency graph for ui_fwupgradedialog.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_FwUpgradeDialog
- class Ui::FwUpgradeDialog

Namespaces

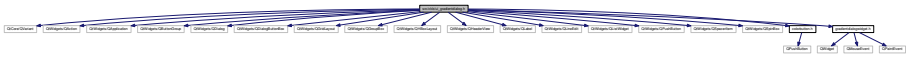
- Ui

13.215 src/ckb/ui_gradientdialog.h File Reference

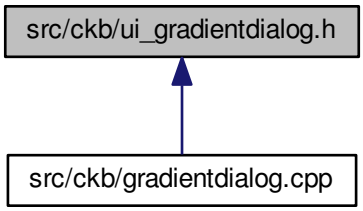
```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QGroupBox>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QLineEdit>
#include <QtWidgets/QListWidget>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QSpinBox>
#include "colorbutton.h"
#include "gradientdialogwidget.h"
```

Include dependency graph for ui_gradientdialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_GradientDialog
- class Ui::GradientDialog

Namespaces

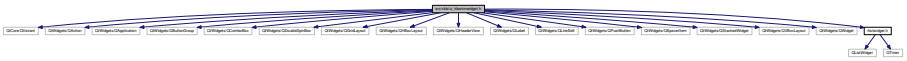
- Ui

13.216 src/ckb/ui_kbanimwidget.h File Reference

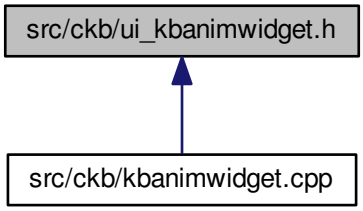
```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QDoubleSpinBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QLineEdit>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QStackedWidget>
#include <QtWidgets/QVBoxLayout>
#include <QtWidgets/QWidget>
#include "rlistwidget.h"
```

Include dependency graph for ui_kbanimwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_KbAnimWidget
- class Ui::KbAnimWidget

Namespaces

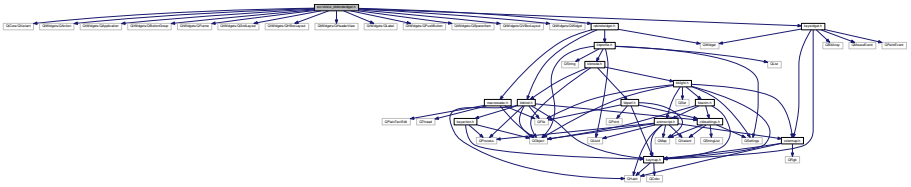
- Ui

13.217 src/ckb/ui_kbbindwidget.h File Reference

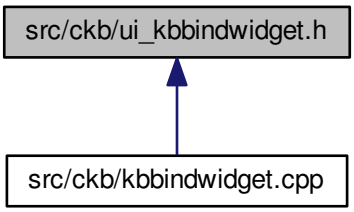
```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QVBoxLayout>
#include <QtWidgets/QWidget>
#include "keywidget.h"
#include "rebindwidget.h"
```

Include dependency graph for ui_kbbindwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_KbBindWidget
- class Ui::KbBindWidget

Namespaces

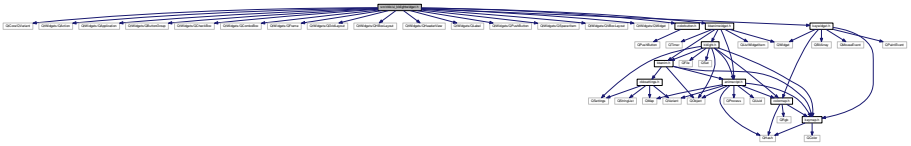
- Ui

13.218 src/ckb/ui_kblightwidget.h File Reference

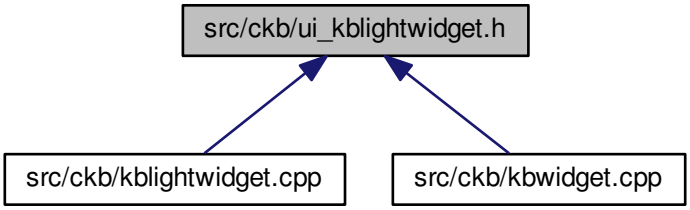
```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QVBoxLayout>
#include <QtWidgets/QWidget>
#include "colorbutton.h"
#include "kbanimwidget.h"
#include "keywidget.h"
```

Include dependency graph for ui_kblightwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_KbLightWidget
- class Ui::KbLightWidget

Namespaces

- Ui

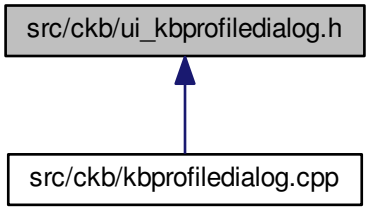
13.219 src/ckb/ui_kbprofiledialog.h File Reference

```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QSpacerItem>
#include "rlistwidget.h"
```

Include dependency graph for ui_kbprofiledialog.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_KbProfileDialog
- class Ui::KbProfileDialog

Namespaces

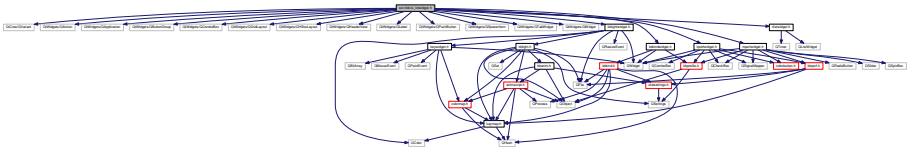
- Ui

13.220 src/ckb/ui_kbwidget.h File Reference

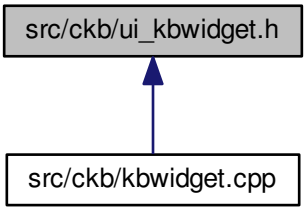
```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QTabWidget>
#include <QtWidgets/QWidget>
#include <kblightwidget.h>
#include "kbbindwidget.h"
#include "kperfwidget.h"
#include "mperfwidget.h"
#include "rlistwidget.h"
```

Include dependency graph for ui_kbwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_KbWidget
- class Ui::KbWidget

Namespaces

- Ui

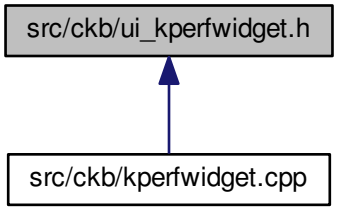
13.221 src/ckb/ui_kperfwidget.h File Reference

```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QSpinBox>
#include <QtWidgets/QWidget>
#include "colorbutton.h"
```

Include dependency graph for ui_kperfwidget.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_KPerfWidget
- class Ui::KPerfWidget

Namespaces

- Ui

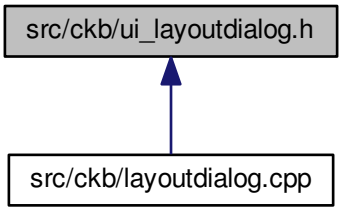
13.222 src/ckb/ui_layoutdialog.h File Reference

```
#include <QtCore/QVariant>
```

```
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QSpacerItem>
```

Include dependency graph for ui_layoutdialog.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_LayoutDialog
- class Ui::LayoutDialog

Namespaces

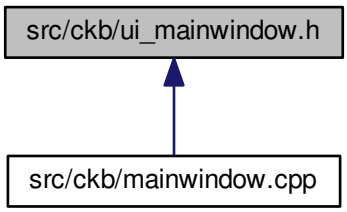
- Ui

13.223 src/ckb/ui_mainwindow.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QMainWindow>
#include <QtWidgets/QTabWidget>
#include <QtWidgets/QWidget>
```

Include dependency graph for ui_mainwindow.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_MainWindow
- class Ui::MainWindow

Namespaces

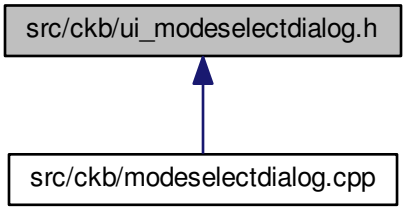
- Ui

13.224 src/ckb/ui_modeselectdialog.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QDialog>
#include <QtWidgets/QDialogButtonBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QListWidget>
#include <QtWidgets/QPushButton>
```

Include dependency graph for ui_modeselectdialog.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_ModeSelectDialog
- class Ui::ModeSelectDialog

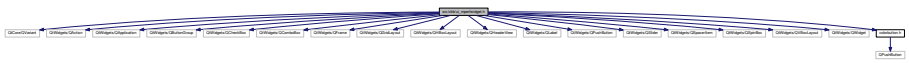
Namespaces

- Ui

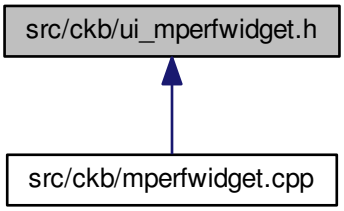
13.225 src/ckb/ui_mperfwidget.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSlider>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QSpinBox>
#include <QtWidgets/QVBoxLayout>
#include <QtWidgets/QWidget>
#include "colorbutton.h"
```

Include dependency graph for ui_mperfwidget.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_MPerfWidget
- class Ui::MPerfWidget

Namespaces

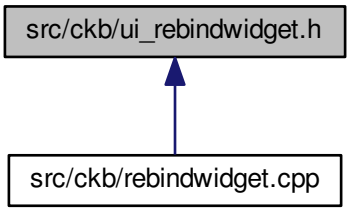
- Ui

13.226 src/ckb/ui_rebindwidget.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QGroupBox>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QLineEdit>
#include <QtWidgets/QPlainTextEdit>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QRadioButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QSpinBox>
#include <QtWidgets/QTabWidget>
#include <QtWidgets/QVBoxLayout>
#include <QtWidgets/QWidget>
```

Include dependency graph for ui_rebindwidget.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_RebindWidget
- class Ui::RebindWidget

Namespaces

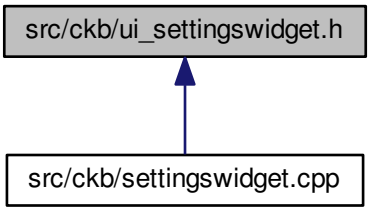
- Ui

13.227 src/ckb/ui_settingswidget.h File Reference

```
#include <QtCore/QVariant>
#include <QtWidgets/QAction>
#include <QtWidgets/QApplication>
#include <QtWidgets/QButtonGroup>
#include <QtWidgets/QCheckBox>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QFrame>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLabel>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QWidget>
```

Include dependency graph for ui_settingswidget.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- class Ui_SettingsWidget
- class Ui::SettingsWidget

Namespaces

- Ui

Index

~AnimAddDialog
AnimAddDialog, 50
~AnimScript
AnimScript, 59
~AnimSettingDialog
AnimSettingDialog, 107
~CkbSettings
CkbSettings, 127
~CkbSettingsWriter
CkbSettingsWriter, 140
~ColorMap
ColorMap, 152
~ExtraSettingsWidget
ExtraSettingsWidget, 178
~FwUpgradeDialog
FwUpgradeDialog, 187
~GradientDialog
GradientDialog, 218
~KPerfWidget
KPerfWidget, 711
~Kb
Kb, 266
~KbAnimWidget
KbAnimWidget, 351
~KbBindWidget
KbBindWidget, 405
~KbFirmware
KbFirmware, 416
~KbLight
KbLight, 430
~KbLightWidget
KbLightWidget, 466
~KbMode
KbMode, 494
~KbProfileDialog
KbProfileDialog, 558
~KbWidget
KbWidget, 574
~KeyAction
KeyAction, 610
~LayoutDialog
LayoutDialog, 724
~MPerfWidget
MPerfWidget, 787
~MacroReader
MacroReader, 728
~MainWindow
MainWindow, 742
~ModeSelectDialog
ModeSelectDialog, 759
~QuaGzipFile
QuaGzipFile, 822
~QuaZIODevice
QuaZIODevice, 828
~QuaZip
QuaZip, 837
~QuaZipDir
QuaZipDir, 868
~QuaZipFile
QuaZipFile, 888
~QuaZipFilePrivate
QuaZipFilePrivate, 919
~RebindWidget
RebindWidget, 976
~SGroup
SGroup, 1038
~SettingsWidget
SettingsWidget, 1016
_LAYOUT_MAX
KeyMap, 638
_MODEL_MAX
KeyMap, 639
_model, 1585
_readlines_ctx, 1259
_DEFAULT_SOURCE
os.h, 1466
_FILE_OFFSET_BIT
ioapi.h, 1783
_GNU_SOURCE
os.h, 1466
_LARGEFILE64_SOURCE
ioapi.h, 1784
_Z_OF
ioapi.h, 1786
unzip.c, 1834
unzip.h, 1888, 1889
zip.c, 1907
zip.h, 1960, 1961
__FILE_NOPATH__
includes.h, 1300
__USE_FILE_OFFSET64
ioapi.h, 1783
__USE_LARGEFILE64
ioapi.h, 1783
_alpha
ColorButton, 150
GradientButton, 206
_angleSnap

KbPerf, 535
_anim
AnimSettingDialog, 118
_animList
KbLight, 453
_animMap
KbLight, 453
_backing
CkbSettingsWriter, 141
_bigIcons
ColorButton, 150
_bind
KbBind, 395
KbMode, 503
_bindMap
KeyWidget, 682
_checkUpdates
KbFirmware, 416
_cmd_get
notify.c, 1454
_cmd_macro
input.c, 1304
_color
ColorButton, 150
_colorBuffer
AnimScript, 83
_colorMap
KbLight, 454
KeyWidget, 682
_colors
AnimScript, 83
ColorMap, 159
GradientDialogWidget, 242
_count
ColorMap, 159
_curDpi
KbPerf, 510
_current
GradientDialogWidget, 242
_currentMode
Kb, 307
KbProfile, 555
_currentProfile
Kb, 307
_daemonVersion
KbManager, 488
_delay
Kb, 307
_devParent
KbBind, 395
_devices
KbManager, 488
_dimming
KbLight, 454
_displayColorMap
KeyWidget, 683
_dither
Kb, 307
_dpiIndicator
KbPerf, 535
_eventTimer
KbManager, 488
_fileForBoard
KbFirmware, 417
_frameRate
Kb, 307
_freeprofile
profile.c, 1468
_friendlyName
Key, 604
_globalRemap
KbBind, 395
_globalSettings
ckbsettings.cpp, 1675
_guiVersion
KbManager, 488
_guid
KbAnim, 336
_hwProfile
Kb, 307
_iOpacity
KbPerf, 535
_id
KbMode, 503
KbProfile, 555
_indicatorList
KbLight, 454
_indicatorMap
KbLight, 454
_indicators
KeyWidget, 683
_info
AnimScript, 83
_isActive
KbAnim, 336
_isActiveKp
KbAnim, 336
_kbManager
KbManager, 489
_keyMap
KbProfile, 555
_keyNames
ColorMap, 159
_keys
AnimAddDialog, 53
AnimScript, 83
KbAnim, 336
KeyMap, 660
_kiss_fft_guts.h
C_ADD, 1612
C_ADDTO, 1612
C_FIXDIV, 1612
C_MUL, 1612
C_MULBYSCALAR, 1612
C_SUB, 1612
C_SUBFROM, 1613

HALF_OF, 1613
KISS_FFT_COS, 1613
KISS_FFT_SIN, 1613
kf_cexp, 1613
MAXFACTORS, 1614
pcpx, 1614
S_MUL, 1614
_lastFrameAnimMap
KbLight, 454
_lastFrameDimming
KbLight, 454
_lastFrameIndicatorMap
KbLight, 454
_latestForBoard
KbFirmware, 418
_layout
Kb, 308
_ledthread
input_linux.c, 1329
_liftHeight
KbPerf, 536
_light
AnimAddDialog, 53
KbMode, 503
_map
AnimScript, 83
KbAnim, 336
KbBind, 395
KbLight, 454
_mapCount
ColorMap, 159
_mkdevpath
devnode.c, 1259
_mknotifynode
devnode.c, 1262
_mode
KbAnim, 336
_modeList
ModeSelectDialog, 760
_model
Kb, 308
_modes
KbProfile, 555
_monochrome
KeyWidget, 683
_mouseAccel
Kb, 308
_name
KbAnim, 336
KbMode, 503
KbProfile, 555
_needsMapRefresh
KbLight, 454
_needsSave
Kb, 308
KbAnim, 336
KbBind, 395
KbLight, 454
KbMode, 503
KbPerf, 536
KbProfile, 556
_needsUpdate
KbBind, 396
KbPerf, 536
_nk95cmd
usb.h, 1556
usb_linux.c, 1586
_opacity
KbAnim, 337
_paramValues
AnimScript, 83
_parameters
KbAnim, 337
_patch
keymap.cpp, 1724
_path
AnimScript, 83
_perf
KbMode, 503
_presetValues
AnimScript, 84
_presets
AnimScript, 83
_previewAnim
KbLight, 455
_profiles
Kb, 308
_qColorMap
KbLight, 455
_removals
CkbSettingsWriter, 141
_resetusb
usb.c, 1524
usb.h, 1557
_rgbMode
KeyWidget, 683
_rmnotifynode
devnode.c, 1263
_scanTimer
KbManager, 489
_script
KbAnim, 337
_scriptGuid
KbAnim, 337
_scriptName
KbAnim, 337
_scrollSpeed
Kb, 308
_setLabel
ColorButton, 150
_settings
SGroup, 1038
_setupusb
usb.c, 1524
_shareDimming
kblight.cpp, 1705

_start
KbLight, 455
_start_dev
device.c, 1234
_stops
GradientButton, 206
GradientDialogWidget, 242
_storageName
Key, 604
_tempParameters
KbAnim, 337
_updateconnected
devnode.c, 1263
_updates
CkbSettingsWriter, 141
_usbrecv
usb.c, 1528
usb.h, 1558
_usbsend
usb.c, 1530
usb.h, 1559
_value
KeyAction, 632
_winLock
KbBind, 396
_xyLink
MPerfWidget, 797

ACCEL
command.h, 1227
ACTIVE
command.h, 1227
ADD
KeyWidget, 665
ALWAYS_OFF
KPerfWidget, 709
ALWAYS_ON
KPerfWidget, 709
ACT_LIGHT
device.h, 1238
ACT_LOCK
device.h, 1238
ACT_M1
device.h, 1238
ACT_M2
device.h, 1239
ACT_M3
device.h, 1239
ACT_MR_RING
device.h, 1239
ACT_NEXT
device.h, 1239
ACT_NEXT_NOWRAP
device.h, 1239
aCopyLabel
Ui::KbAnimWidget, 344
Ui_KbAnimWidget, 1087
ALLOC
unzip.c, 1832
zip.c, 1904
aMissingLabel
Ui::KbAnimWidget, 344
Ui_KbAnimWidget, 1087
aMissingPage
Ui::KbAnimWidget, 344
Ui_KbAnimWidget, 1087
ANGLE
ckb-pinwheel/main.c, 1423
ANIM_KRSTOP
keyaction.cpp, 1716
ANIM_MAX
ckb-ripple/main.c, 1440
ckb-wave/main.c, 1446
ANIM_ONCE
keyaction.cpp, 1716
ANSI_ENTER_H
keymap.cpp, 1719
ANSI_ENTER_W
keymap.cpp, 1720
ANSI_ENTER_X
keymap.cpp, 1720
ANSI_ENTER_Y
keymap.cpp, 1720
ANSI_LSHIFT_W
keymap.cpp, 1720
ANSI_LSHIFT_X
keymap.cpp, 1720
aNameLabel
Ui::KbAnimWidget, 344
Ui_KbAnimWidget, 1087
APPEND_STATUS_CREATE
zip.h, 1958
aSnapBox
Ui::MPerfWidget, 773
Ui_MPerfWidget, 1162
AUTO_CHECK_TIME
kbfirmware.cpp, 1701
aVerLabel
Ui::KbAnimWidget, 345
Ui_KbAnimWidget, 1088
aa
ckb-rain/main.c, 1432
ab
ckb-rain/main.c, 1432
action
KbBind, 370
actionAbout
Ui::MainWindow, 736
Ui_MainWindow, 1146
actionButton
Ui::FwUpgradeDialog, 198
Ui_FwUpgradeDialog, 1072
actionExit
Ui::MainWindow, 736
Ui_MainWindow, 1146
actionTab
Ui::RebindWidget, 956

Ui_RebindWidget, 1186
active
devcmd.__unnamed__, 161
activeDevices
kb.cpp, 1692
activeLights
kblight.cpp, 1705
Add
KbAnim, 314
add_data_in_datablock
zip.c, 1907
addAnim
KbAnimWidget, 351
KbLight, 430
addCurrentFileToDirectoryMap
QuaZipPrivate, 933
addDevice
MainWindow, 742
addNewModeItem
KbWidget, 574
addNewProfileItem
KbProfileDialog, 558
addPreset
GradientDialog, 218
adjustDisplay
KeyAction, 610
advance
AnimScript, 60
ag
ckb-rain/main.c, 1432
alloc
ColorMap, 153
allocate_new_datablock
zip.c, 1908
allocprofile
devcmd.__unnamed__, 161
profile.c, 1469
profile.h, 1485
allowAlpha
ColorButton, 144, 145
altBox
Ui::SettingsWidget, 1031
Ui_SettingsWidget, 1208
altLabel
Ui::SettingsWidget, 1031
Ui_SettingsWidget, 1208
angle
ckb-wave/main.c, 1452
angleDialChanged
AnimSettingDialog, 107
angleDialMapper
AnimSettingDialog, 118
angleFlip
animsettingdialog.cpp, 1646
angleSnap
KbPerf, 510, 511
angleSpinnerChanged
AnimSettingDialog, 108
angleSpinnerMapper
AnimSettingDialog, 118
angleSpinners
AnimSettingDialog, 118
anim
ckb-gradient/main.c, 1410
ckb-ripple/main.c, 1443
ckb-wave/main.c, 1452
AnimScript::Param, 58, 1644
anim_add
ckb-heat/main.c, 1412
ckb-mviz/main.c, 1417
ckb-ripple/main.c, 1440
ckb-wave/main.c, 1446
anim_remove
ckb-heat/main.c, 1413
ckb-mviz/main.c, 1417
ckb-ripple/main.c, 1440
ckb-wave/main.c, 1446
animAction
KeyAction, 611
AnimAddDialog, 48
∼AnimAddDialog, 50
_keys, 53
_light, 53
AnimAddDialog, 49
AnimAddDialog, 49
chosenPreset, 50
chosenScript, 51
on_animBox_activated, 51
on_presetBox_activated, 52
on_previewBox_clicked, 53
scripts, 53
showPreview, 53
ui, 54
animBox
Ui::AnimAddDialog, 46
Ui::RebindWidget, 956
Ui_AnimAddDialog, 1042
Ui_RebindWidget, 1186
animButton
Ui::KbLightWidget, 461
Ui::RebindWidget, 956
Ui_KbLightWidget, 1102
Ui_RebindWidget, 1187
animChanged
KbAnimWidget, 352
animCountLabel
Ui::ExtraSettingsWidget, 170
Ui_ExtraSettingsWidget, 1063
animInfo
KeyAction, 612
animKeypress
KbLight, 431
animKrBox
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
AnimList

KbLight, 429
animList
KbLight, 432
Ui::KbAnimWidget, 344
Ui_KbAnimWidget, 1087
animName
Ui::AnimSettingDialog, 93
Ui_AnimSettingDialog, 1050
animOnceBox
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
animPage
Ui::KbAnimWidget, 344
Ui_KbAnimWidget, 1088
animPathLabel
Ui::ExtraSettingsWidget, 170
Ui_ExtraSettingsWidget, 1064
animScanButton
Ui::ExtraSettingsWidget, 170
Ui_ExtraSettingsWidget, 1064
AnimScript, 54
~AnimScript, 59
_colorBuffer, 83
_colors, 83
_info, 83
_keys, 83
_map, 83
_paramValues, 83
_path, 83
_presetValues, 84
_presets, 83
advance, 60
AnimScript, 59
AnimScript, 59
author, 60
begin, 61
colors, 62
copy, 62
copyright, 63
count, 63
description, 64
durationMsec, 84
end, 64
firstFrame, 84
frame, 65
guid, 66
hasFrame, 66
hasKeypress, 67
hasParam, 67
inFrame, 84
init, 68
initialized, 84
KP_NAME, 84
KP_NONE, 84
KP_POSITION, 84
keypress, 69
lastFrame, 84
license, 70
list, 70
load, 71
minX, 85
minY, 85
name, 74
param, 74
paramIterator, 76
parameters, 75
path, 76
preset, 76
PresetValue, 59
presets, 77
printParams, 77
process, 85
readAnyFrame, 85
readFrame, 85
readProcess, 78
repeatMsec, 85
retrigger, 79
scan, 80
scripts, 85
setDuration, 80
stop, 81
stopped, 85
version, 82
year, 82
AnimSettingDialog, 99
~AnimSettingDialog, 107
_anim, 118
angleDialChanged, 107
angleDialMapper, 118
angleSpinnerChanged, 108
angleSpinnerMapper, 118
angleSpinners, 118
AnimSettingDialog, 102
AnimSettingDialog, 102
hLine, 109
hasRepeat, 119
kpStopCheck, 119
lastDuration, 119
name, 110
newDuration, 110
on_delayBox_valueChanged, 111
on_kpDelayBox_valueChanged, 112
on_kpReleaseBox_stateChanged, 112
on_kpRepeatBox_valueChanged, 113
on_repeatBox_valueChanged, 113
settingWidgets, 119
stopCheck, 119
ui, 119
updateMapper, 119
updateParam, 114
updateStops, 117
animTab
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
animWidget
Ui::KbLightWidget, 461

Ui_KbLightWidget, 1103
animation
KeyWidget, 683
animations
KbAnimWidget, 365
animcolor
ckb-gradient/main.c, 1410
ckb-heat/main.c, 1416
ckb-mviz/main.c, 1422
ckb-pinwheel/main.c, 1426
ckb-ripple/main.c, 1444
ckb-wave/main.c, 1452
animlength
ckb-pinwheel/main.c, 1426
ckb-ripple/main.c, 1444
ckb-wave/main.c, 1452
anims
ckb-heat/main.c, 1416
animscript.cpp
ONE_DAY, 1643
urlParam, 1643
animsettingdialog.cpp
angleFlip, 1646
appShare
main.cpp, 1759
mainwindow.cpp, 1764
append
KbProfile, 543
appendProfile
Kb, 269
applyButton
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
applyChanges
RebindWidget, 976
applyIndicators
KbPerf, 511
ar
ckb-rain/main.c, 1432
atEnd
QuaZipFile, 888
author
AnimScript, 60
autoClose
QuaZipPrivate, 937
autoFWBox
Ui::SettingsWidget, 1031
Ui_SettingsWidget, 1208
AutoRun, 120
available, 120
disable, 121
enable, 121
isEnabled, 122
once, 123
autoSave
Kb, 269
autorun.cpp
settingPath, 1648
available
AutoRun, 120

BIND
command.h, 1228
BOTH
KPerfWidget, 709
BR1
led_keyboard.c, 1381
BR2
led_keyboard.c, 1381
BR4
led_keyboard.c, 1381
BTN_WHEELDOWN
daemon/keymap.h, 1742
BTN_WHEELUP
daemon/keymap.h, 1742
BUFREADCOMMENT
unzip.c, 1833
zip.c, 1905
BUILD.md, 1219
BUTTON_HID_COUNT
keymap.c, 1336
backing
CkbSettings, 138
base
KbLight, 432
baseDpiIdx
KbPerf, 512, 513
begin
AnimScript, 61
beginGroup
CkbSettings, 127
bgButton
Ui::KbLightWidget, 461
Ui_KbLightWidget, 1103
bigIcons
ColorButton, 145, 146
bind
devcmd.__unnamed__, 161
KbBindWidget, 413
KbMode, 494
KbPerf, 513
RebindWidget, 1004
bindAction
KbBind, 371
BindMap
KeyWidget, 664
bindMap
KeyWidget, 665
bindTab
Ui::KbWidget, 598
Ui_KbWidget, 1115
bindWidget
Ui::KbWidget, 598
Ui_KbWidget, 1115
binding, 1509
bit_reverse_table
led_keyboard.c, 1390

blend
KbAnim, 317
blendAdd
kbanim.cpp, 1694
blendBox
Ui::KbAnimWidget, 345
Ui_KbAnimWidget, 1088
blendDivide
kbanim.cpp, 1694
blendFunc
kbanim.cpp, 1694
blendMultiply
kbanim.cpp, 1694
blendNormal
kbanim.cpp, 1694
blendSubtract
kbanim.cpp, 1695
blob
FwUpgradeDialog, 192
boxXChanged
MPerfWidget, 787
boxXMapper
MPerfWidget, 797
boxYChanged
MPerfWidget, 788
boxYMapper
MPerfWidget, 797
brightnessBox
Ui::ExtraSettingsWidget, 170
Ui::KbLightWidget, 461
Ui_ExtraSettingsWidget, 1064
Ui_KbLightWidget, 1103
btnClearMacro
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
btnStartMacro
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
btnStopMacro
Ui::RebindWidget, 957
Ui_RebindWidget, 1187
builtIn
GradientDialog::Preset, 800
buttonBox
Ui::AnimAddDialog, 47
Ui::AnimSettingDialog, 93
Ui::ExtraSettingsWidget, 170
Ui::GradientDialog, 211
Ui::KbProfileDialog, 569
Ui::LayoutDialog, 721
Ui::ModeSelectDialog, 756
Ui_AnimAddDialog, 1042
Ui_AnimSettingDialog, 1050
Ui_ExtraSettingsWidget, 1064
Ui_GradientDialog, 1079
Ui_KbProfileDialog, 1108
Ui_LayoutDialog, 1142
Ui_ModeSelectDialog, 1150
buttonMapper1
MPerfWidget, 797
buttonMapper2
MPerfWidget, 797
byPosition
KeyMap, 640
bytesAvailable
QuaZipFile, 889

CAPS
KbPerf, 508
CMD_FIRST
command.h, 1227
CMD_LAST
command.h, 1228
CMD_VT_FIRST
command.h, 1227
C_ADD
_kiss_fft_guts.h, 1612
C_ADDTO
_kiss_fft_guts.h, 1612
C_FIXDIV
_kiss_fft_guts.h, 1612
C_MUL
_kiss_fft_guts.h, 1612
C_MULBYSCALAR
_kiss_fft_guts.h, 1612
C_SUB
_kiss_fft_guts.h, 1612
C_SUBFROM
_kiss_fft_guts.h, 1613
CENTRALHEADERMAGIC
zip.c, 1905
CKB_CONTAINER
ckb-anim.h, 1654
CKB_COPYRIGHT
ckb-anim.h, 1654
CKB_DESCRIPTION
ckb-anim.h, 1654
CKB_DOWN
ckb-anim.h, 1655
CKB_GRAD_MAX
ckb-anim.h, 1655
CKB_GUID
ckb-anim.h, 1655
CKB_KEYCLEAR
ckb-anim.h, 1655
CKB_KEYNAME_MAX
ckb-anim.h, 1655
CKB_KP_NAME
ckb-anim.h, 1655
CKB_KP_NONE
ckb-anim.h, 1655
CKB_KP_POSITION
ckb-anim.h, 1655
CKB_KPMODE
ckb-anim.h, 1655
CKB_LICENSE
ckb-anim.h, 1656

CKB_LIVEPARAMS
ckb-anim.h, 1656
CKB_MAX_WORD
ckb-anim.h, 1656
CKB_NAME
ckb-anim.h, 1656
CKB_PARAM
ckb-anim.h, 1656
CKB_PARAM_AGRADIENT
ckb-anim.h, 1656
CKB_PARAM_ANGLE
ckb-anim.h, 1656
CKB_PARAM_ARGB
ckb-anim.h, 1656
CKB_PARAM_BOOL
ckb-anim.h, 1656
CKB_PARAM_DOUBLE
ckb-anim.h, 1657
CKB_PARAM_GRADIENT
ckb-anim.h, 1657
CKB_PARAM_LABEL
ckb-anim.h, 1657
CKB_PARAM_LONG
ckb-anim.h, 1657
CKB_PARAM_RGB
ckb-anim.h, 1657
CKB_PARAM_STRING
ckb-anim.h, 1657
CKB_PARSE_AGRADIENT
ckb-anim.h, 1657
CKB_PARSE_ANGLE
ckb-anim.h, 1657
CKB_PARSE_ARGB
ckb-anim.h, 1657
CKB_PARSE_BOOL
ckb-anim.h, 1658
CKB_PARSE_DOUBLE
ckb-anim.h, 1658
CKB_PARSE_GRADIENT
ckb-anim.h, 1658
CKB_PARSE_LONG
ckb-anim.h, 1658
CKB_PARSE_RGB
ckb-anim.h, 1658
CKB_PARSE_STRING
ckb-anim.h, 1658
CKB_PREEMPT
ckb-anim.h, 1658
CKB_PRESET_END
ckb-anim.h, 1658
CKB_PRESET_PARAM
ckb-anim.h, 1658
CKB_PRESET_START
ckb-anim.h, 1659
CKB_REAL_ANGLE
ckb-anim.h, 1659
CKB_REPEAT
ckb-anim.h, 1659
CKB_TIME_ABSOLUTE
ckb-anim.h, 1659
CKB_TIME_DURATION
ckb-anim.h, 1659
CKB_TIMEMODE
ckb-anim.h, 1659
CKB_UP
ckb-anim.h, 1659
CKB_VERSION
ckb-anim.h, 1659
CLEAR_KEYBIT
structures.h, 1518
CMD_COUNT
command.h, 1226
CMD_DEV_COUNT
command.h, 1226
CRC32
crypt.h, 1777
cacheWritesInProgress
ckbsettings.cpp, 1674
ckbsettingswriter.cpp, 1678
calculate
QuaAdler32, 811
QuaChecksum32, 814
QuaCrc32, 817
call_zopen64
qioapi.cpp, 1791
call_zseek64
qioapi.cpp, 1791
call_ztell64
qioapi.cpp, 1791
cancelButton
Ui::FwUpgradeDialog, 198
Ui::RebindWidget, 957
Ui_FwUpgradeDialog, 1073
Ui_RebindWidget, 1187
capsBox
Ui::KPerfWidget, 696
Ui::SettingsWidget, 1031
Ui_KPerfWidget, 1130
Ui_SettingsWidget, 1208
capsColorOff
Ui::KPerfWidget, 696
Ui_KPerfWidget, 1130
capsColorOn
Ui::KPerfWidget, 696
Ui_KPerfWidget, 1130
capsLabel
Ui::SettingsWidget, 1032
Ui_SettingsWidget, 1208
CaseSensitivity
QuaZip, 836
caseSensitivity
QuaZipDir, 869
QuaZipFilePrivate, 920
catchUp
KbAnim, 318
cd

QuaZipDir, 869
cdUp
QuaZipDir, 871
centralWidget
Ui::MainWindow, 736
Ui_MainWindow, 1146
change
RListWidget, 1009
changeAnim
KbLightWidget, 467
changeAnimKeys
KbLightWidget, 467
changeColor
KbLightWidget, 468
checkFwUpdates
MainWindow, 743
checkUpdates
KbFirmware, 419
checksum
QuaAdler32, 812
QuaCrc32, 818
childGroups
CkbSettings, 127
childKeys
CkbSettings, 128
chosenPreset
AnimAddDialog, 50
chosenScript
AnimAddDialog, 51
ckb-anim.h
CKB_CONTAINER, 1654
CKB_COPYRIGHT, 1654
CKB_DESCRIPTION, 1654
CKB_DOWN, 1655
CKB_GRAD_MAX, 1655
CKB_GUID, 1655
CKB_KEYCLEAR, 1655
CKB_KEYNAME_MAX, 1655
CKB_KP_NAME, 1655
CKB_KP_NONE, 1655
CKB_KP_POSITION, 1655
CKB_KPMODE, 1655
CKB_LICENSE, 1656
CKB_LIVEPARAMS, 1656
CKB_MAX_WORD, 1656
CKB_NAME, 1656
CKB_PARAM, 1656
CKB_PARAM_AGRADIENT, 1656
CKB_PARAM_ANGLE, 1656
CKB_PARAM_ARGB, 1656
CKB_PARAM_BOOL, 1656
CKB_PARAM_DOUBLE, 1657
CKB_PARAM_GRADIENT, 1657
CKB_PARAM_LABEL, 1657
CKB_PARAM_LONG, 1657
CKB_PARAM_RGB, 1657
CKB_PARAM_STRING, 1657
CKB_PARSE_AGRADIENT, 1657
CKB_PARSE_ANGLE, 1657
CKB_PARSE_ARGB, 1657
CKB_PARSE_BOOL, 1658
CKB_PARSE_DOUBLE, 1658
CKB_PARSE_GRADIENT, 1658
CKB_PARSE_LONG, 1658
CKB_PARSE_RGB, 1658
CKB_PARSE_STRING, 1658
CKB_PREEMPT, 1658
CKB_PRESET_END, 1658
CKB_PRESET_PARAM, 1658
CKB_PRESET_START, 1659
CKB_REAL_ANGLE, 1659
CKB_REPEAT, 1659
CKB_TIME_ABSOLUTE, 1659
CKB_TIME_DURATION, 1659
CKB_TIMEMODE, 1659
CKB_UP, 1659
CKB_VERSION, 1659
ckb_alpha_blend, 1660
ckb_frame, 1660
ckb_getline, 1661
ckb_grad_color, 1662
ckb_info, 1663
ckb_init, 1664
ckb_keypress, 1664
ckb_parameter, 1666
ckb_read_params, 1666
ckb_scan_grad, 1667
ckb_start, 1668
ckb_time, 1669
FALSE, 1659
M_PI, 1660
main, 1669
printurl, 1671
TRUE, 1660
urldecode, 1672
ckb-daemon/main.c
features_mask, 1404
hwload_mode, 1404
localecase, 1396
main, 1397
main_ac, 1404
main_av, 1404
quit, 1399
quitWithLock, 1400
reset_stop, 1404
restart, 1401
sighandler, 1402
sighandler2, 1403
timespec_add, 1403
ckb-gradient/main.c
anim, 1410
animcolor, 1410
ckb_frame, 1406
ckb_info, 1407
ckb_init, 1408
ckb_keypress, 1408

ckb_parameter, 1409
ckb_start, 1409
ckb_time, 1410
HOLD, 1406
kphold, 1410
kprelease, 1410
NONE, 1406
randomize, 1410
ckb-heat/main.c
anim_add, 1412
anim_remove, 1413
animcolor, 1416
anims, 1416
ckb_frame, 1413
ckb_info, 1414
ckb_init, 1414
ckb_keypress, 1415
ckb_parameter, 1415
ckb_start, 1415
ckb_time, 1415
ffade, 1416
pressestofull, 1416
randomBright, 1416
ckb-mviz/main.c
anim_add, 1417
anim_remove, 1417
animcolor, 1422
ckb_frame, 1418
ckb_info, 1418
ckb_init, 1419
ckb_keypress, 1419
ckb_parameter, 1419
ckb_start, 1419
ckb_time, 1420
gcounter, 1422
getFreqDec, 1420
inbuf, 1422
max, 1421
min, 1421
outbuf, 1422
pas, 1422
power, 1422
powers, 1422
ckb-pinwheel/main.c
ANGLE, 1423
animcolor, 1426
animlength, 1426
ckb_frame, 1424
ckb_info, 1424
ckb_init, 1425
ckb_keypress, 1425
ckb_parameter, 1425
ckb_start, 1425
ckb_time, 1426
counter_clock, 1426
frame, 1426
symmetric, 1426
x, 1426
x_offset, 1427
y, 1427
y_offset, 1427
ckb-rain/main.c
aa, 1432
ab, 1432
ag, 1432
ar, 1432
ckb_frame, 1428
ckb_info, 1429
ckb_init, 1430
ckb_keypress, 1430
ckb_parameter, 1430
ckb_start, 1431
ckb_time, 1431
DROP_MAX, 1428
drop, 1432
drop_add, 1431
maxsize, 1432
period, 1433
spawn, 1433
speed, 1433
tick, 1433
ckb-random/main.c
ckb_frame, 1435
ckb_info, 1435
ckb_init, 1435
ckb_keypress, 1436
ckb_parameter, 1436
ckb_start, 1436
ckb_time, 1437
current, 1438
fadein, 1438
newtarget, 1437
phase, 1438
target, 1438
useopacity, 1438
ckb-ripple/main.c
ANIM_MAX, 1440
anim, 1443
anim_add, 1440
anim_remove, 1440
animcolor, 1444
animlength, 1444
ckb_frame, 1440
ckb_info, 1441
ckb_init, 1442
ckb_keypress, 1442
ckb_parameter, 1442
ckb_start, 1443
ckb_time, 1443
kbsize, 1444
kprelease, 1444
randomize, 1444
symmetric, 1444
x_offset, 1444
y_offset, 1444
ckb-wave/main.c

ANIM_MAX, 1446
angle, 1452
anim, 1452
anim_add, 1446
anim_remove, 1446
animcolor, 1452
animlength, 1452
ckb_frame, 1446
ckb_info, 1447
ckb_init, 1448
ckb_keypress, 1448
ckb_parameter, 1449
ckb_start, 1450
ckb_time, 1451
kprelease, 1452
left, 1452
symmetric, 1452
top, 1452
width, 1452
ckb_gradient, 1654
ckb_key, 1652
ckb_runctx, 1653
ckb_alpha_blend
ckb-anim.h, 1660
ckb_err
includes.h, 1300
ckb_err_fn
includes.h, 1300
ckb_err_nofile
includes.h, 1301
ckb_fatal
includes.h, 1301
ckb_fatal_fn
includes.h, 1301
ckb_fatal_nofile
includes.h, 1301
ckb_frame
ckb-anim.h, 1660
ckb-gradient/main.c, 1406
ckb-heat/main.c, 1413
ckb-mviz/main.c, 1418
ckb-pinwheel/main.c, 1424
ckb-rain/main.c, 1428
ckb-random/main.c, 1435
ckb-ripple/main.c, 1440
ckb-wave/main.c, 1446
ckb_getline
ckb-anim.h, 1661
ckb_grad_color
ckb-anim.h, 1662
ckb_info
ckb-anim.h, 1663
ckb-gradient/main.c, 1407
ckb-heat/main.c, 1414
ckb-mviz/main.c, 1418
ckb-pinwheel/main.c, 1424
ckb-rain/main.c, 1429
ckb-random/main.c, 1435
ckb-ripple/main.c, 1441
ckb-wave/main.c, 1447
includes.h, 1301
ckb_info_fn
includes.h, 1301
ckb_info_nofile
includes.h, 1301
ckb_init
ckb-anim.h, 1664
ckb-gradient/main.c, 1408
ckb-heat/main.c, 1414
ckb-mviz/main.c, 1419
ckb-pinwheel/main.c, 1425
ckb-rain/main.c, 1430
ckb-random/main.c, 1435
ckb-ripple/main.c, 1442
ckb-wave/main.c, 1448
ckb_keypress
ckb-anim.h, 1664
ckb-gradient/main.c, 1408
ckb-heat/main.c, 1415
ckb-mviz/main.c, 1419
ckb-pinwheel/main.c, 1425
ckb-rain/main.c, 1430
ckb-random/main.c, 1436
ckb-ripple/main.c, 1442
ckb-wave/main.c, 1448
ckb_parameter
ckb-anim.h, 1666
ckb-gradient/main.c, 1409
ckb-heat/main.c, 1415
ckb-mviz/main.c, 1419
ckb-pinwheel/main.c, 1425
ckb-rain/main.c, 1430
ckb-random/main.c, 1436
ckb-ripple/main.c, 1442
ckb-wave/main.c, 1449
ckb_read_params
ckb-anim.h, 1666
ckb_s_err
includes.h, 1301
ckb_s_out
includes.h, 1301
ckb_scan_grad
ckb-anim.h, 1667
ckb_start
ckb-anim.h, 1668
ckb-gradient/main.c, 1409
ckb-heat/main.c, 1415
ckb-mviz/main.c, 1419
ckb-pinwheel/main.c, 1425
ckb-rain/main.c, 1431
ckb-random/main.c, 1436
ckb-ripple/main.c, 1443
ckb-wave/main.c, 1450
ckb_time
ckb-anim.h, 1669
ckb-gradient/main.c, 1410

ckb-heat/main.c, 1415
ckb-mviz/main.c, 1420
ckb-pinwheel/main.c, 1426
ckb-rain/main.c, 1431
ckb-random/main.c, 1437
ckb-ripple/main.c, 1443
ckb-wave/main.c, 1451
ckb_warn
includes.h, 1301
ckb_warn_fn
includes.h, 1302
ckb_warn_nofile
includes.h, 1302
ckbDaemonVersion
KbManager, 480
ckbDaemonVersionF
KbManager, 480
ckbGuiVersion
KbManager, 481
ckbGuiVersionF
KbManager, 481
CkbSettings, 124
~CkbSettings, 127
backing, 138
beginGroup, 127
childGroups, 127
childKeys, 128
CkbSettings, 126, 127
CkbSettings, 126, 127
cleanUp, 129
contains, 130
containsGroup, 130
endGroup, 131
get, 131
groups, 138
isBusy, 132
pwd, 133, 134
remove, 135
removeCache, 139
set, 135
setValue, 136
value, 137
writeCache, 139
CkbSettingsWriter, 139
~CkbSettingsWriter, 140
_backing, 141
_removals, 141
_updates, 141
CkbSettingsWriter, 140
CkbSettingsWriter, 140
run, 141
ckbVersion
KbFirmware::FW, 184
ckbsettings.cpp
_globalSettings, 1675
cacheWritesInProgress, 1674
globalCache, 1675
globalSettings, 1674
globalThread, 1675
lockMutex, 1674
lockMutexCache, 1674
lockMutexStatic, 1674
lockMutexStatic2, 1674
settingsCacheMutex, 1675
settingsMutex, 1675
ckbsettingswriter.cpp
cacheWritesInProgress, 1678
lockMutex, 1677
lockMutexCache, 1677
lockMutexStatic, 1677
lockMutexStatic2, 1677
settingsCacheMutex, 1678
settingsMutex, 1678
cleanBlob
FwUpgradeDialog, 187
cleanUp
CkbSettings, 129
cleanup
MainWindow, 744
clear
ColorMap, 153
clearAnimation
KeyWidget, 666
clearDirectoryMap
QuaZipPrivate, 934
clearSelection
KbAnimWidget, 353
KeyWidget, 666
close
KbLight, 433
QuaGzipFile, 822
QuaZIODevice, 828
QuaZip, 838
QuaZipFile, 889
closeAction
MainWindow, 752
closeEvent
FwUpgradeDialog, 187
MainWindow, 745
closeusb
usb.c, 1531
usb.h, 1561
cmd
command.h, 1227
Kb, 308
cmd_active_kb
device.h, 1240
device_keyboard.c, 1248
cmd_active_mouse
device.h, 1241
device_mouse.c, 1252
cmd_bind
input.c, 1306
input.h, 1320
cmd_dpi
dpi.c, 1280

dpi.h, 1286
cmd_dpisel
dpi.c, 1281
dpi.h, 1287
cmd_erase
profile.c, 1469
profile.h, 1486
cmd_eraseprofile
profile.c, 1470
profile.h, 1486
cmd_fwupdate
firmware.c, 1293
firmware.h, 1297
cmd_get
notify.c, 1456
notify.h, 1462
cmd_hwload_kb
profile.h, 1487
profile_keyboard.c, 1499
cmd_hwload_mouse
profile.h, 1488
profile_mouse.c, 1502
cmd_hwsave_kb
profile.h, 1489
profile_keyboard.c, 1500
cmd_hwsave_mouse
profile.h, 1490
profile_mouse.c, 1503
cmd_iauto
led.c, 1341
led.h, 1347
cmd_id
profile.c, 1470
profile.h, 1491
cmd_idle_kb
device.h, 1241
device_keyboard.c, 1248
cmd_idle_mouse
device.h, 1242
device_mouse.c, 1253
cmd_inotify
led.c, 1342
led.h, 1348
cmd_io_none
device_vtable.c, 1256
cmd_ioff
led.c, 1342
led.h, 1348
cmd_ion
led.c, 1343
led.h, 1349
cmd_lift
dpi.c, 1281
dpi.h, 1287
cmd_macro
input.c, 1307
input.h, 1321
cmd_macro_none
device_vtable.c, 1256
cmd_name
profile.c, 1471
profile.h, 1491
cmd_none
device_vtable.c, 1256
cmd_notify
notify.c, 1457
notify.h, 1462
cmd_pollrate
device.h, 1242
device_mouse.c, 1253
cmd_profileid
profile.c, 1471
profile.h, 1492
cmd_profilename
profile.c, 1472
profile.h, 1492
cmd_rebind
input.c, 1307
input.h, 1321
cmd_restart
notify.c, 1457
notify.h, 1463
cmd_rgb
led.c, 1343
led.h, 1349
cmd_snap
dpi.c, 1281
dpi.h, 1287
cmd_strings
command.c, 1224
cmd_unbind
input.c, 1307
input.h, 1321
cmdhandler
command.h, 1227
cmdhandler_io
command.h, 1227
cmdhandler_mac
command.h, 1227
cmdpath
Kb, 308
color
ColorButton, 146, 147
KbLight, 434, 435
colorAt
GradientDialogWidget, 230
ColorButton, 141
_alpha, 150
_bigIcons, 150
_color, 150
_setLabel, 150
allowAlpha, 144, 145
bigIcons, 145, 146
color, 146, 147
ColorButton, 144
colorChanged, 148

ColorButton, 144
pickColor, 148
setLabel, 148, 149
updateImage, 149
colorChanged
ColorButton, 148
GradientDialog, 219
MPerfWidget, 789
colorClicked
MPerfWidget, 790
colorForName
ColorMap, 154, 155
colorLink
MPerfWidget, 798
ColorMap, 151
~ColorMap, 152
_colors, 159
_count, 159
_keyNames, 159
_mapCount, 159
alloc, 153
clear, 153
colorForName, 154, 155
ColorMap, 152
ColorMap, 152
colors, 155, 156
count, 156
deAlloc, 156
init, 157
keyNames, 158
operator=, 158
operator==, 159
colorMap
KbLight, 435
KeyWidget, 666, 667
colormap.cpp
qs_strcmp, 1680
colormap.h
QColorMap, 1682
QColorMapIterator, 1682
QMutableColorMapIterator, 1682
colors
AnimScript, 62
ColorMap, 155, 156
comboBox
Ui::LayoutDialog, 721
Ui_LayoutDialog, 1142
command.h
ACCEL, 1227
ACTIVE, 1227
BIND, 1228
CMD_FIRST, 1227
CMD_LAST, 1228
CMD_VT_FIRST, 1227
DELAY, 1227
DITHER, 1227
DPI, 1228
DPISEL, 1228
ERASE, 1227
ERASEPROFILE, 1227
FPS, 1227
FWUPDATE, 1227
GET, 1228
HWLOAD, 1227
HWSAVE, 1227
IAUTO, 1228
ID, 1228
IDLE, 1227
INOTIFY, 1228
IOFF, 1228
ION, 1228
LAYOUT, 1227
LIFT, 1228
MACRO, 1228
MODE, 1227
NAME, 1228
NONE, 1227
NOTIFY, 1228
NOTIFYOFF, 1227
NOTIFYON, 1227
POLLRATE, 1227
PROFILEID, 1228
PROFILENAME, 1228
REBIND, 1228
RESTART, 1228
RGB, 1228
SCROLLSPEED, 1227
SNAP, 1228
SWITCH, 1227
UNBIND, 1228
command.c
cmd_strings, 1224
readcmd, 1220
TRY_WITH_RESET, 1220
command.h
CMD_COUNT, 1226
CMD_DEV_COUNT, 1226
cmd, 1227
cmdhandler, 1227
cmdhandler_io, 1227
cmdhandler_mac, 1227
devcmd, 1227
readcmd, 1229
CommandLineBackground
main.cpp, 1758
CommandLineClose
main.cpp, 1758
CommandLineError
main.cpp, 1758
CommandLineHelpRequested
main.cpp, 1758
CommandLineOK
main.cpp, 1758
CommandLineVersionRequested
main.cpp, 1758
CommandLineParseResults

main.cpp, 1758
comment
QuaZipFileInfo, 908
QuaZipFileInfo64, 915
QuaZipNewInfo, 929
QuaZipPrivate, 937
commentCodec
QuaZipPrivate, 937
commitParams
KbAnim, 320
compressDir
JlCompress, 244
compressFile
JlCompress, 245, 246
compressFiles
JlCompress, 247
compressSubDir
JlCompress, 249
compressedSize
QuaZipFileInfo, 908
QuaZipFileInfo64, 915
configLabel
mainwindow.cpp, 1764
Constants
QuaZip, 836
contains
CkbSettings, 130
KeyMap, 640
containsGroup
CkbSettings, 130
convertCaseSensitivity
QuaZip, 839
convertMacroBox
RebindWidget, 978
copy
AnimScript, 62
copyButton
Ui::KbBindWidget, 401
Ui::KPerfWidget, 696
Ui::MPerfWidget, 773
Ui_KbBindWidget, 1095
Ui_KPerfWidget, 1130
Ui_MPerfWidget, 1162
copyData
JlCompress.cpp, 1788
copyright
AnimScript, 63
corsair_kbcopy
daemon/keymap.h, 1745
keymap.c, 1336
corsair_mousecopy
daemon/keymap.h, 1745
keymap.c, 1337
count
AnimScript, 63
ColorMap, 156
KeyMap, 641
QuaZipDir, 871
counter_clock
ckb-pinwheel/main.c, 1426
crc
QuaZipFileInfo, 908
QuaZipFileInfo64, 915
QuaZipFilePrivate, 920
crypt.h
CRC32, 1777
decrypt_byte, 1778
init_keys, 1778
update_keys, 1779
zdecode, 1778
zencode, 1778
csDefault
QuaZip, 836
csInsensitive
QuaZip, 836
csSensitive
QuaZip, 836
csize
QuaZipFile, 890
ctrlBox
Ui::SettingsWidget, 1032
Ui_SettingsWidget, 1209
ctrlLabel
Ui::SettingsWidget, 1032
Ui_SettingsWidget, 1209
curLabel
Ui::FwUpgradeDialog, 198
Ui_FwUpgradeDialog, 1073
curfile64_info, 1903
current
ckb-random/main.c, 1438
GradientDialogWidget, 231
KbAnimWidget, 365
currentBind
Kb, 269
currentChanged
GradientDialog, 219
GradientDialogWidget, 231
currentData
RListWidget, 1012
currentLight
Kb, 270
currentMode
Kb, 271
KbProfile, 543, 544
KbWidget, 591
currentPerf
Kb, 271
currentPreset
GradientDialog, 226
currentProfile
Kb, 272
currentSelection
KbBindWidget, 413
KbLightWidget, 476

d

QuaGzipFile, 825
QuaZIODevice, 832
QuaZipDir, 882
DE
KeyMap, 638
DELAY
command.h, 1227
DITHER
command.h, 1227
DK
KeyMap, 638
DPI
command.h, 1228
DPISEL
command.h, 1228
DAEMON.md, 1219
DATA_ROLE
RListWidget, 1012
DEBUG
usb_linux.c, 1585
DEF_MEM_LEVEL
zip.h, 1959
DELAY_LONG
usb.h, 1546
DELAY_MEDIUM
usb.h, 1546
DELAY_SHORT
usb.h, 1546
DESCRIPTORHEADERMAGIC
zip.c, 1905
DEV_MAX
device.h, 1239
DIALOG_HEIGHT_MAX
fwupgradedialog.cpp, 1685
DIALOG_HEIGHT_MIN
fwupgradedialog.cpp, 1686
DIALOG_WIDTH
fwupgradedialog.cpp, 1686
DPI_BASE
mperfwidget.cpp, 1774
DPI_COUNT
KbPerf, 536
MPerfWidget, 798
structures.h, 1518
DPI_CUST_IDX
rebindwidget.cpp, 1965
DPI_CUSTOM
KeyAction, 632
DPI_CYCLE_DOWN
KeyAction, 633
DPI_CYCLE_UP
KeyAction, 633
DPI_DOWN
KeyAction, 633
DPI_MAX
KbPerf, 536
MPerfWidget, 798
DPI_MIN
KbPerf, 536
MPerfWidget, 798
DPI_OFFSET
rebindwidget.cpp, 1965
DPI_SNIPER
KeyAction, 633
DPI_UP
KeyAction, 633
DROP_MAX
ckb-rain/main.c, 1428
daemon/keymap.h
BTN_WHEELDOWN, 1742
BTN_WHEELUP, 1742
corsair_kbcopy, 1745
corsair_mousecopy, 1745
hid_kb_translate, 1745
hid_mouse_translate, 1747
KEY_BACKSLASH_ISO, 1742
KEY_CORSAIR, 1742
KEY_NONE, 1742
KEY_UNBOUND, 1743
keymap, 1748
LED_DPI, 1743
LED_MOUSE, 1743
MOUSE_BUTTON_FIRST, 1743
MOUSE_EXTRA_FIRST, 1743
N_BUTTONS_EXTENDED, 1743
N_BUTTONS_HW, 1743
N_KEY_ZONES, 1743
N_KEYBYTES_EXTENDED, 1743
N_KEYBYTES_HW, 1743
N_KEYBYTES_INPUT, 1744
N_KEYS_EXTENDED, 1744
N_KEYS_EXTRA, 1744
N_KEYS_HW, 1744
N_KEYS_INPUT, 1744
N_MOUSE_ZONES, 1744
SCAN_KBD, 1744
SCAN_MOUSE, 1744
SCAN_SILENT, 1744
daemonDialogText
mainwindow.cpp, 1764
dataDescriptorWritingEnabled
QuaZipPrivate, 937
dataForBoard
KbFirmware, 419
dateTime
QuaZipFileInfo, 908
QuaZipFileInfo64, 915
QuaZipNewInfo, 929
deAlloc
ColorMap, 156
decrypt_byte
crypt.h, 1778
defaultAction
KbBind, 371, 372
KeyAction, 612
defaultFileNameCodec

QuaZipPrivate, 938
delayBox
Ui::AnimSettingDialog, 93
Ui::ExtraSettingsWidget, 170
Ui_AnimSettingDialog, 1051
Ui_ExtraSettingsWidget, 1064
delayButtons
Ui::RebindWidget, 957
Ui_RebindWidget, 1188
deleteButton
Ui::KbAnimWidget, 345
Ui_KbAnimWidget, 1088
deleteHw
Kb, 272
deletePrevious
Kb, 273
description
AnimScript, 64
devLabel
Ui::FwUpgradeDialog, 198
Ui_FwUpgradeDialog, 1073
devParent
KbBind, 372
devTab
Ui::KbWidget, 598
Ui_KbWidget, 1115
devUpdate
KbWidget, 575
devcmd, 1226
command.h, 1227
devcmd.__unnamed__, 160
active, 161
allocprofile, 161
bind, 161
dpi, 161
dpisel, 161
erase, 161
eraseprofile, 161
freeprofile, 161
fwupdate, 161
get, 161
hwload, 161
hwsave, 161
iauto, 161
id, 161
idle, 161
inotify, 161
ioff, 161
ion, 162
lift, 162
loadprofile, 162
macro, 162
name, 162
notify, 162
pollrate, 162
profileid, 162
profilename, 162
rebind, 162
restart, 162
rgb, 162
setmodeindex, 162
snap, 162
start, 162
unbind, 162
updatedpi, 162
updateindicators, 162
updatergb, 162
device
KbProfileDialog, 564
KbWidget, 591
device.c
_start_dev, 1234
devlistmutex, 1236
devmutex, 1236
hwload_mode, 1236
inputmutex, 1236
keyboard, 1236
macromutex, 1236
macromutex2, 1237
macrovar, 1237
start_dev, 1236
device.h
ACT_LIGHT, 1238
ACT_LOCK, 1238
ACT_M1, 1238
ACT_M2, 1239
ACT_M3, 1239
ACT_MR_RING, 1239
ACT_NEXT, 1239
ACT_NEXT_NOWRAP, 1239
cmd_active_kb, 1240
cmd_active_mouse, 1241
cmd_idle_kb, 1241
cmd_idle_mouse, 1242
cmd_pollrate, 1242
DEV_MAX, 1239
devmutex, 1247
dmutex, 1239
IN_CORSAIR, 1239
IN_HID, 1239
IS_CONNECTED, 1240
imutex, 1239
inputmutex, 1247
keyboard, 1247
macromutex, 1247
macromutex2, 1247
macrovar, 1247
mmutex, 1240
mmutex2, 1240
mvar, 1240
setactive, 1240
setactive_kb, 1242
setactive_mouse, 1244
setmodeindex_nrgb, 1245
start_dev, 1246
start_kb_nrgb, 1246

device_keyboard.c
cmd_active_kb, 1248
cmd_idle_kb, 1248
setactive_kb, 1249
setmodeindex_nrgb, 1251
start_kb_nrgb, 1251
device_mouse.c
cmd_active_mouse, 1252
cmd_idle_mouse, 1253
cmd_pollrate, 1253
setactive_mouse, 1253
device_vtable.c
cmd_io_none, 1256
cmd_macro_none, 1256
cmd_none, 1256
int1_int_none, 1256
int1_void_none, 1256
loadprofile_none, 1257
vtable_keyboard, 1257
vtable_keyboard_nonrgb, 1257
vtable_mouse, 1257
devices
KbManager, 482
devicesLabel
Ui::SettingsWidget, 1032
Ui_SettingsWidget, 1209
devlistmutex
device.c, 1236
devmain
usb.c, 1533
devmutex
device.c, 1236
device.h, 1247
devnode.c
_mkdevpath, 1259
_mknotifynode, 1262
_rmnotifynode, 1263
_updateconnected, 1263
devpath, 1271
gid, 1271
MAX_BUFFER, 1259
mkdevpath, 1264
mkfwnode, 1265
mknotifynode, 1266
readlines, 1266
readlines_ctx_free, 1267
readlines_ctx_init, 1268
rm_recursive, 1268
rmdevpath, 1269
rmnotifynode, 1270
S_GID_READ, 1259
updateconnected, 1270
devnode.h
devpath, 1279
gid, 1279
mkdevpath, 1273
mkfwnode, 1274
mknotifynode, 1275
readlines, 1275
readlines_ctx, 1273
readlines_ctx_free, 1276
readlines_ctx_init, 1277
rmdevpath, 1277
rmnotifynode, 1278
S_CUSTOM, 1273
S_CUSTOM_R, 1273
S_READ, 1273
S_READDIR, 1273
S_READWRITE, 1273
updateconnected, 1279
devpath
devnode.c, 1271
devnode.h, 1279
Kb, 308
kbmanager.cpp, 1708
mainwindow.cpp, 1764
settingswidget.cpp, 2074
didLoad
KbBind, 373
KbLight, 436
KbPerf, 514
didUpdateSelection
KbAnimWidget, 353
dimming
KbLight, 436, 437
dirName
QuaZipDir, 872
directoryCaseInsensitive
QuaZipPrivate, 938
directoryCaseSensitive
QuaZipPrivate, 938
disable
AutoRun, 121
disableAppNap
media_mac.m, 1769
diskNumberStart
QuaZipFileInfo, 908
QuaZipFileInfo64, 915
displayColorMap
KeyWidget, 667
dither
Kb, 273, 274
ditherBox
Ui::ExtraSettingsWidget, 170
Ui_ExtraSettingsWidget, 1064
Divide
KbAnim, 315
dmutex
device.h, 1239
doUpdate
KbMode, 495
downloadFinished
KbFirmware, 420
downloaded
KbFirmware, 420
dpi

devcmd.__unnamed__, 161
KbPerf, 514, 515
dpi.c
cmd_dpi, 1280
cmd_dpisel, 1281
cmd_lift, 1281
cmd_snap, 1281
loaddpi, 1281
printdpi, 1282
savedpi, 1283
updatedpi, 1284
dpi.h
cmd_dpi, 1286
cmd_dpisel, 1287
cmd_lift, 1287
cmd_snap, 1287
loaddpi, 1287
printdpi, 1288
savedpi, 1289
updatedpi, 1290
dpiAction
KeyAction, 613
dpiBaseIdx
KbPerf, 536
dpiBox
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
dpiButton
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
dpiClr
KbPerf, 536
dpiColor
KbPerf, 516, 517
dpiCurX
KbPerf, 536
dpiCurY
KbPerf, 537
dpiCustLabel
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
dpiCustXBox
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
dpiCustYBox
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
dpiCycleDown
KbPerf, 517
dpiCycleUp
KbPerf, 518
dpiDown
KbPerf, 518
dpiEnabled
KbPerf, 519
dpiExp
mperfwidget.cpp, 1772
dpiIndicator
KbPerf, 520
dpiInfo
KeyAction, 614
dpiLog
mperfwidget.cpp, 1773
dpiOn
KbPerf, 537
dpiRound
mperfwidget.cpp, 1773
dpiUp
KbPerf, 520
dpiWarning
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
dpiX
KbPerf, 537
dpiY
KbPerf, 537
dpisel
devcmd.__unnamed__, 161
dpiset, 1510
dragged
RListWidget, 1012
drawInfo
KeyWidget, 667
driverName
KeyAction, 614
drop
ckb-rain/main.c, 1432
drop_add
ckb-rain/main.c, 1431
duplicateAnim
KbAnimWidget, 354
KbLight, 437
durationMsec
AnimScript, 84

ERASE
command.h, 1227
ERASEPROFILE
command.h, 1227
ES
KeyMap, 638
EU
KeyMap, 638
EU_DVORAK
KeyMap, 638
eBox1
Ui::MPerfWidget, 773
Ui_MPerfWidget, 1163
eBox2
Ui::MPerfWidget, 773
Ui_MPerfWidget, 1163
eBox3
Ui::MPerfWidget, 773
Ui_MPerfWidget, 1163
eBox4
Ui::MPerfWidget, 773
Ui_MPerfWidget, 1163

eBox5
Ui::MPerfWidget, 773
Ui_MPerfWidget, 1163
ENDHEADERMAGIC
zip.c, 1905
ENUM_C
media.h, 1766
ENUM_END_C
media.h, 1766
EXTERN_C
media.h, 1766
effectiveParams
KbAnim, 320
enable
AutoRun, 121
enableChanged
MPerfWidget, 790
enableMapper
MPerfWidget, 798
end
AnimScript, 64
endGroup
CkbSettings, 131
enter
RListWidget, 1009
entryInfoList
QuaZipDir, 872, 873
entryInfoList64
QuaZipDir, 873, 874
entryList
QuaZipDir, 874, 875
erase
devcmd.__unnamed__, 161
eraseprofile
devcmd.__unnamed__, 161
euid_guard_start
os.h, 1467
euid_guard_stop
os.h, 1467
evLoop
FwUpgradeDialog, 193
eventTimer
KbManager, 482
exec
FwUpgradeDialog, 188
exists
QuaZipDir, 875, 876
exitSuccess
FwUpgradeDialog, 193
externalAttr
QuaZipFileInfo, 908
QuaZipFileInfo64, 915
QuaZipNewInfo, 929
extra
QuaZipFileInfo, 908
QuaZipFileInfo64, 916
SettingsWidget, 1023
extraButton
Ui::SettingsWidget, 1032
Ui_SettingsWidget, 1209
extraGlobal
QuaZipNewInfo, 930
extraLocal
QuaZipNewInfo, 930
ExtraSettingsWidget, 175
~ExtraSettingsWidget, 178
ExtraSettingsWidget, 177
ExtraSettingsWidget, 177
on_animScanButton_clicked, 178
on_brightnessBox_clicked, 179
on_delayBox_clicked, 179
on_ditherBox_clicked, 179
on_fpsBox_valueChanged, 180
on_mAccelBox_clicked, 180
on_sAccelBox_clicked, 181
on_sSpeedBox_valueChanged, 181
on_trayBox_clicked, 182
pollUpdates, 182
ui, 183
extractDir
JlCompress, 252
extractFile
JlCompress, 254, 256
extractFiles
JlCompress, 257
extrasettingswidget.cpp
lastSharedDimming, 1682

FPS
command.h, 1227
FR
KeyMap, 638
FWUPDATE
command.h, 1227
FALSE
ckb-anim.h, 1659
FEAT_ADJRATE
structures.h, 1518
FEAT_ANSI
structures.h, 1518
FEAT_BIND
structures.h, 1518
FEAT_COMMON
structures.h, 1518
FEAT_FWUPDATE
structures.h, 1518
FEAT_FWVERSION
structures.h, 1518
FEAT_HWLOAD
structures.h, 1519
FEAT_ISO
structures.h, 1519
FEAT_LMASK
structures.h, 1519
FEAT_MONOCHROME
structures.h, 1519
FEAT_MOUSEACCEL

structures.h, 1519
FEAT_NOTIFY
structures.h, 1519
FEAT_POLLRATE
structures.h, 1519
FEAT_RGB
structures.h, 1519
FEAT_STD_NRGB
structures.h, 1519
FEAT_STD_RGB
structures.h, 1520
FW
KbFirmware::FW, 184
FW_MAXSIZE
firmware.c, 1292
FW_NOFILE
firmware.c, 1292
FW_OK
firmware.c, 1292
FW_USBFAIL
firmware.c, 1292
FW_WRONGDEV
firmware.c, 1292
fadein
ckb-random/main.c, 1438
features
Kb, 309
features_mask
ckb-daemon/main.c, 1404
usb.c, 1541
ffade
ckb-heat/main.c, 1416
file_in_zip64_read_info_s, 1830
fileName
KbFirmware::FW, 184
QuaZipFilePrivate, 920
fileNameCodec
QuaZipPrivate, 938
filePath
QuaZipDir, 877
fill_qiodevice64_filefunc
qioapi.cpp, 1792
fill_qiodevice_filefunc
qioapi.cpp, 1793
fill_zlib_filefunc64_32_def_from_filefunc32
ioapi.h, 1786
qioapi.cpp, 1794
fillRect
GradientDialogWidget, 231
filter
QuaZipDir, 877
find
Kb, 274
KbProfile, 544
findAnim
KbLight, 438
findAnimIdx
KbLight, 439
firmware
Kb, 309
firmware.c
cmd_fwupdate, 1293
FW_MAXSIZE, 1292
FW_NOFILE, 1292
FW_OK, 1292
FW_USBFAIL, 1292
FW_WRONGDEV, 1292
fwupdate, 1293
getfwversion, 1295
firmware.h
cmd_fwupdate, 1297
getfwversion, 1298
firstFrame
AnimScript, 84
flags
QuaZipFileInfo, 908
QuaZipFileInfo64, 916
flush
QuaGzipFile, 823
QuaZIODevice, 829
fnBox
Ui::RebindWidget, 958
Ui_RebindWidget, 1188
fnButton
Ui::RebindWidget, 958
Ui_RebindWidget, 1189
fnKeys
RebindWidget, 1005
fps
KbManager, 482
fpsBox
Ui::ExtraSettingsWidget, 171
Ui_ExtraSettingsWidget, 1064
fpsWarnLabel
Ui::ExtraSettingsWidget, 171
Ui_ExtraSettingsWidget, 1064
frame
AnimScript, 65
ckb-pinwheel/main.c, 1426
frameDisplayed
KbLight, 439
frameRate
Kb, 275
frameUpdate
Kb, 276
KbLight, 440
free_datablock
zip.c, 1909
free_linkedlist
zip.c, 1909
freebind
input.c, 1308
input.h, 1322
freemode
profile.c, 1472
freeprofile

devcmd.__unnamed__, 161
profile.c, 1473
profile.h, 1493
friendlyActionName
KbBind, 373
friendlyName
KbBind, 374
Key, 603
KeyAction, 615
KeyMap, 641
fromName
KeyMap, 643
fromStorage
KeyMap, 644
fromString
GradientButton, 202
functions
kbanim.cpp, 1695
fwLabel
Ui::KbWidget, 598
Ui_KbWidget, 1115
fwTable
KbFirmware, 424
fwTableHash
KbFirmware, 424
fwUpdButton
Ui::KbWidget, 598
Ui_KbWidget, 1115
fwUpdLabel
Ui::KbWidget, 599
Ui_KbWidget, 1115
fwUpdLayout
Ui::KbWidget, 599
Ui_KbWidget, 1116
fwUpdPath
Kb, 309
fwUpdate
Kb, 277
fwUpdateFinished
FwUpgradeDialog, 189
Kb, 278
fwUpdateProgress
FwUpgradeDialog, 190
Kb, 278
FwUpgradeDialog, 184
~FwUpgradeDialog, 187
blob, 192
cleanBlob, 187
closeEvent, 187
evLoop, 193
exec, 188
exitSuccess, 193
fwUpdateFinished, 189
fwUpdateProgress, 190
FwUpgradeDialog, 186
FwUpgradeDialog, 186
kb, 193
on_actionButton_clicked, 190
on_cancelButton_clicked, 191
removeDev, 191
saveBlob, 192
savePath, 193
ui, 193
fwVersion
KbFirmware::FW, 184
fwupdate
devcmd.__unnamed__, 161
firmware.c, 1293
fwupgradedialog.cpp
DIALOG_HEIGHT_MAX, 1685
DIALOG_HEIGHT_MIN, 1686
DIALOG_WIDTH, 1686
ids, 1686
verifyFw, 1685

GB
KeyMap, 638
GB_DVORAK
KeyMap, 638
GET
command.h, 1228
GLAIVE
KeyMap, 639
GLAIVE_HEIGHT
keymap.cpp, 1720
GLAIVE_WIDTH
keymap.cpp, 1720
GUID
KbProfileDialog, 565
KbWidget, 592
gcounter
ckb-mviz/main.c, 1422
get
CkbSettings, 131
devcmd.__unnamed__, 161
get_vtable
usb.c, 1535
getActualFileName
QuaZipFile, 891
getCaseSensitivity
QuaZipFile, 892
getComment
QuaZip, 840
getCommentCodec
QuaZip, 840
getCurrentFileInfo
QuaZip, 841, 842
getCurrentFileName
QuaZip, 843
getDefaultFileNameCodec
QuaZipPrivate, 935
getEntriesCount
QuaZip, 844
getFileInfo
QuaZipFile, 892, 893
getFileInfoList
QuaZip, 845

QuaZipPrivate, 935
getFileInfoList64
QuaZip, 845
getFileList
JlCompress, 258
getFileName
QuaGzipFile, 823
QuaZipFile, 894
getFileNameCodec
QuaZip, 846
getFileNameList
QuaZip, 846
getFreqDec
ckb-mviz/main.c, 1420
getGradient
GradientDialog, 220
getIndicator
KbPerf, 521
getIoDevice
QuaZIODevice, 830
QuaZip, 847
getKeyMap
Kb, 279
getLayout
KeyMap, 644, 645
getMacroNumber
Kb, 279
KbBind, 374
getMacroPath
Kb, 279
KbBind, 375
getMap
keymap.cpp, 1725
getMode
QuaZip, 847
getModel
KeyMap, 646
getMuteState
media.h, 1767
media_linux.cpp, 1768
media_mac.m, 1770
getNTFSTime
quazipfileinfo.cpp, 1819
getNTFSaTime
QuaZipFileInfo64, 911
getNTFScTime
QuaZipFileInfo64, 912
getNTFSmTime
QuaZipFileInfo64, 912
getPermissions
QuaZipFileInfo, 907
QuaZipFileInfo64, 913
getUnzFile
QuaZip, 848
getZip
QuaZipFile, 894
getZipError
QuaZip, 849
QuaZipFile, 894
getZipFile
QuaZip, 850
getZipName
QuaZip, 850
QuaZipFile, 895
getfwversion
firmware.c, 1295
firmware.h, 1298
gethwmodename
profile.c, 1474
profile.h, 1493
gethwprofilename
profile.c, 1474
profile.h, 1494
getid
profile.c, 1475
profile.h, 1495
getmodename
profile.c, 1476
profile.h, 1495
getprofilename
profile.c, 1476
profile.h, 1496
gid
devnode.c, 1271
devnode.h, 1279
GlaiveKeys
keymap.cpp, 1733
glaiveOverlay
keywidget.cpp, 1750
globalCache
ckbsettings.cpp, 1675
globalRemap
KbBind, 376
globalRemapTime
KbBind, 396
globalSettings
ckbsettings.cpp, 1674
globalThread
ckbsettings.cpp, 1675
goToFirstFile
QuaZip, 851
goToFirstUnmappedFile
QuaZipPrivate, 936
goToNextFile
QuaZip, 852
gradient
GradientDialog::Preset, 800
GradientButton, 200
_alpha, 206
_stops, 206
fromString, 202
GradientButton, 202
gradientChanged, 203
GradientButton, 202
pickGradient, 203
toString, 204

updateImage, 205
gradientChanged
GradientButton, 203
GradientDialog, 214
~GradientDialog, 218
addPreset, 218
colorChanged, 219
currentChanged, 219
currentPreset, 226
getGradient, 220
GradientDialog, 217
GradientDialog, 217
makeIcon, 221
on_presetDelete_clicked, 222
on_presetList_currentItemChanged, 222
on_presetName_textEdited, 223
on_presetSave_clicked, 223
on_stopOpacity_valueChanged, 224
on_stopPos_valueChanged, 224
presets, 226
setPreset, 224
ui, 227
updatePresets, 225
GradientDialog::Preset, 799
builtIn, 800
gradient, 800
name, 800
Preset, 800
GradientDialogWidget, 227
_colors, 242
_current, 242
_stops, 242
colorAt, 230
current, 231
currentChanged, 231
fillRect, 231
GradientDialogWidget, 230
GradientDialogWidget, 230
makeStops, 232
mouseMoveEvent, 233
mousePressEvent, 234
mouseReleaseEvent, 234
moveCurrent, 235
paintEvent, 236
selected, 242
selectedOffset, 242
selectedPos, 242
selectedStop, 237
selectionColors, 238
setCurrentColor, 238
setStops, 239
stopCount, 240
stopRect, 240
stops, 241
gradientdialog.cpp
prefsPath, 1689
gridLayout
Ui::AnimAddDialog, 47
Ui::AnimSettingDialog, 94
Ui::ExtraSettingsWidget, 171
Ui::FwUpgradeDialog, 199
Ui::GradientDialog, 211
Ui::KbAnimWidget, 345
Ui::KbBindWidget, 401
Ui::KbLightWidget, 461
Ui::KbProfileDialog, 569
Ui::KbWidget, 599
Ui::KPerfWidget, 696
Ui::LayoutDialog, 721
Ui::MainWindow, 737
Ui::ModeSelectDialog, 756
Ui::MPerfWidget, 773
Ui::RebindWidget, 959
Ui::SettingsWidget, 1032
Ui_AnimAddDialog, 1042
Ui_AnimSettingDialog, 1051
Ui_ExtraSettingsWidget, 1064
Ui_FwUpgradeDialog, 1073
Ui_GradientDialog, 1079
Ui_KbAnimWidget, 1088
Ui_KbBindWidget, 1095
Ui_KbLightWidget, 1103
Ui_KbProfileDialog, 1108
Ui_KbWidget, 1116
Ui_KPerfWidget, 1130
Ui_LayoutDialog, 1142
Ui_MainWindow, 1146
Ui_ModeSelectDialog, 1150
Ui_MPerfWidget, 1163
Ui_RebindWidget, 1189
Ui_SettingsWidget, 1209
gridLayout_10
Ui::RebindWidget, 959
Ui_RebindWidget, 1189
gridLayout_2
Ui::AnimSettingDialog, 94
Ui::FwUpgradeDialog, 199
Ui::GradientDialog, 211
Ui::KbAnimWidget, 345
Ui::KbBindWidget, 401
Ui::KbLightWidget, 461
Ui::KbWidget, 599
Ui::RebindWidget, 959
Ui_AnimSettingDialog, 1051
Ui_FwUpgradeDialog, 1073
Ui_GradientDialog, 1079
Ui_KbAnimWidget, 1088
Ui_KbBindWidget, 1095
Ui_KbLightWidget, 1103
Ui_KbWidget, 1116
Ui_RebindWidget, 1189
gridLayout_3
Ui::GradientDialog, 212
Ui::KbAnimWidget, 345
Ui::KbWidget, 599
Ui::RebindWidget, 959

Ui::SettingsWidget, 1032
Ui_GradientDialog, 1079
Ui_KbAnimWidget, 1088
Ui_KbWidget, 1116
Ui_RebindWidget, 1189
Ui_SettingsWidget, 1209
gridLayout_4
Ui::KbAnimWidget, 345
Ui::KbLightWidget, 461
Ui::KbWidget, 599
Ui::RebindWidget, 959
Ui_KbAnimWidget, 1088
Ui_KbLightWidget, 1103
Ui_KbWidget, 1116
Ui_RebindWidget, 1189
gridLayout_5
Ui::KbWidget, 599
Ui::RebindWidget, 959
Ui::SettingsWidget, 1032
Ui_KbWidget, 1116
Ui_RebindWidget, 1189
Ui_SettingsWidget, 1209
gridLayout_7
Ui::KbWidget, 599
Ui_KbWidget, 1116
gridLayout_8
Ui::RebindWidget, 959
Ui_RebindWidget, 1189
groupBox_2
Ui::GradientDialog, 212
Ui::RebindWidget, 959
Ui_GradientDialog, 1080
Ui_RebindWidget, 1189
groupBox_3
Ui::RebindWidget, 959
Ui_RebindWidget, 1190
groupBox_4
Ui::RebindWidget, 960
Ui_RebindWidget, 1190
groups
CkbSettings, 138
guid
AnimScript, 66
KbAnim, 321
UsbId, 1218
guidString
UsbId, 1215

HARPOON
KeyMap, 639
HIGH
KbPerf, 507
HW_IMAX
KbPerf, 508
HWLOAD
command.h, 1227
HWSAVE
command.h, 1227
HALF_OF
_kiss_fft_guts.h, 1613
HARPOON_HEIGHT
keymap.cpp, 1720
HARPOON_WIDTH
keymap.cpp, 1720
HAS_ANY_FEATURE
structures.h, 1520
HAS_FEATURES
structures.h, 1520
hLine
AnimSettingDialog, 109
HOLD
ckb-gradient/main.c, 1406
HW_I_COUNT
KbPerf, 537
KPerfWidget, 716
HW_STANDARD
notify.c, 1453
HWMODE_K70
structures.h, 1520
HWMODE_K95
structures.h, 1520
HWMODE_MAX
Kb, 309
structures.h, 1520
HWMODE_OR_RETURN
notify.c, 1453
handleNotificationChannel
KbBind, 376
harpOverlay
keywidget.cpp, 1750
HarpoonKeys
keymap.cpp, 1733
has_key
led.c, 1344
hasCurrentFile
QuaZip, 853
hasCurrentFile_f
QuaZipPrivate, 938
hasDownloaded
KbFirmware, 421
hasFrame
AnimScript, 66
hasGPG
KbFirmware, 424
hasKeypress
AnimScript, 67
hasLed
Key, 604
hasParam
AnimScript, 67
hasParameter
KbAnim, 321
hasRepeat
AnimSettingDialog, 119
hasScan
Key, 604
hasShownNewFW

KbWidget, 592
hash
KbFirmware::FW, 184
height
KbPerf, 507
Key, 604
KeyMap, 647
helpStatus
RebindWidget, 979
hid_kb_translate
daemon/keymap.h, 1745
keymap.c, 1337
hid_mouse_translate
daemon/keymap.h, 1747
keymap.c, 1339
horizontalLayout
Ui::AnimAddDialog, 47
Ui::ExtraSettingsWidget, 171
Ui::GradientDialog, 212
Ui::KbAnimWidget, 345
Ui::KbLightWidget, 462
Ui::KPerfWidget, 697
Ui::MPerfWidget, 774
Ui::RebindWidget, 960
Ui::SettingsWidget, 1032
Ui_AnimAddDialog, 1042
Ui_ExtraSettingsWidget, 1064
Ui_GradientDialog, 1080
Ui_KbAnimWidget, 1088
Ui_KbLightWidget, 1103
Ui_KPerfWidget, 1130
Ui_MPerfWidget, 1163
Ui_RebindWidget, 1190
Ui_SettingsWidget, 1209
horizontalLayout_2
Ui::ExtraSettingsWidget, 171
Ui::GradientDialog, 212
Ui::KbBindWidget, 401
Ui::KbLightWidget, 462
Ui::KPerfWidget, 697
Ui::MPerfWidget, 774
Ui::RebindWidget, 960
Ui::SettingsWidget, 1033
Ui_ExtraSettingsWidget, 1065
Ui_GradientDialog, 1080
Ui_KbBindWidget, 1095
Ui_KbLightWidget, 1103
Ui_KPerfWidget, 1130
Ui_MPerfWidget, 1163
Ui_RebindWidget, 1190
Ui_SettingsWidget, 1209
horizontalLayout_3
Ui::KbLightWidget, 462
Ui::RebindWidget, 960
Ui_KbLightWidget, 1103
Ui_RebindWidget, 1190
horizontalLayout_4
Ui::MPerfWidget, 774
Ui_MPerfWidget, 1163
horizontalSpacer
Ui::AnimAddDialog, 47
Ui::AnimSettingDialog, 94
Ui::ExtraSettingsWidget, 171
Ui::FwUpgradeDialog, 199
Ui::GradientDialog, 212
Ui::KbAnimWidget, 345
Ui::KbBindWidget, 401
Ui::KbProfileDialog, 569
Ui::KPerfWidget, 697
Ui::LayoutDialog, 721
Ui::MPerfWidget, 774
Ui::RebindWidget, 960
Ui::SettingsWidget, 1033
Ui_AnimAddDialog, 1042
Ui_AnimSettingDialog, 1051
Ui_ExtraSettingsWidget, 1065
Ui_FwUpgradeDialog, 1073
Ui_GradientDialog, 1080
Ui_KbAnimWidget, 1089
Ui_KbBindWidget, 1096
Ui_KbProfileDialog, 1109
Ui_KPerfWidget, 1130
Ui_LayoutDialog, 1142
Ui_MPerfWidget, 1163
Ui_RebindWidget, 1190
Ui_SettingsWidget, 1209
horizontalSpacer_10
Ui::RebindWidget, 960
Ui::SettingsWidget, 1033
Ui_RebindWidget, 1190
Ui_SettingsWidget, 1210
horizontalSpacer_11
Ui::RebindWidget, 960
Ui_RebindWidget, 1190
horizontalSpacer_12
Ui::RebindWidget, 960
Ui_RebindWidget, 1190
horizontalSpacer_13
Ui::RebindWidget, 960
Ui_RebindWidget, 1191
horizontalSpacer_14
Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_15
Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_16
Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_17
Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_18
Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_19

Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_2
Ui::AnimAddDialog, 47
Ui::AnimSettingDialog, 94
Ui::ExtraSettingsWidget, 171
Ui::FwUpgradeDialog, 199
Ui::GradientDialog, 212
Ui::KbAnimWidget, 346
Ui::KbBindWidget, 401
Ui::KbProfileDialog, 569
Ui::KbWidget, 599
Ui::KPerfWidget, 697
Ui::LayoutDialog, 721
Ui::MPerfWidget, 774
Ui::RebindWidget, 961
Ui::SettingsWidget, 1033
Ui_AnimAddDialog, 1042
Ui_AnimSettingDialog, 1051
Ui_ExtraSettingsWidget, 1065
Ui_FwUpgradeDialog, 1073
Ui_GradientDialog, 1080
Ui_KbAnimWidget, 1089
Ui_KbBindWidget, 1096
Ui_KbProfileDialog, 1109
Ui_KbWidget, 1116
Ui_KPerfWidget, 1131
Ui_LayoutDialog, 1142
Ui_MPerfWidget, 1164
Ui_RebindWidget, 1191
Ui_SettingsWidget, 1210
horizontalSpacer_20
Ui::RebindWidget, 961
Ui_RebindWidget, 1191
horizontalSpacer_21
Ui::RebindWidget, 961
Ui_RebindWidget, 1192
horizontalSpacer_22
Ui::RebindWidget, 962
Ui_RebindWidget, 1192
horizontalSpacer_23
Ui::RebindWidget, 962
Ui_RebindWidget, 1192
horizontalSpacer_24
Ui::RebindWidget, 962
Ui_RebindWidget, 1192
horizontalSpacer_25
Ui::RebindWidget, 962
Ui_RebindWidget, 1192
horizontalSpacer_26
Ui::RebindWidget, 962
Ui_RebindWidget, 1192
horizontalSpacer_3
Ui::AnimSettingDialog, 94
Ui::ExtraSettingsWidget, 171
Ui::FwUpgradeDialog, 199
Ui::GradientDialog, 212
Ui::KbAnimWidget, 346
Ui::KbLightWidget, 462
Ui::KbWidget, 600
Ui::KPerfWidget, 697
Ui::MPerfWidget, 774
Ui::RebindWidget, 962
Ui_AnimSettingDialog, 1051
Ui_ExtraSettingsWidget, 1065
Ui_FwUpgradeDialog, 1073
Ui_GradientDialog, 1080
Ui_KbAnimWidget, 1089
Ui_KbLightWidget, 1104
Ui_KbWidget, 1116
Ui_KPerfWidget, 1131
Ui_MPerfWidget, 1164
Ui_RebindWidget, 1192
horizontalSpacer_4
Ui::AnimSettingDialog, 94
Ui::FwUpgradeDialog, 199
Ui::KbAnimWidget, 346
Ui::KbLightWidget, 462
Ui::KPerfWidget, 697
Ui::MPerfWidget, 774
Ui::RebindWidget, 962
Ui_AnimSettingDialog, 1051
Ui_FwUpgradeDialog, 1073
Ui_KbAnimWidget, 1089
Ui_KbLightWidget, 1104
Ui_KPerfWidget, 1131
Ui_MPerfWidget, 1164
Ui_RebindWidget, 1192
horizontalSpacer_5
Ui::AnimSettingDialog, 94
Ui::KbAnimWidget, 346
Ui::KbLightWidget, 462
Ui::KPerfWidget, 697
Ui::MPerfWidget, 774
Ui::RebindWidget, 962
Ui_AnimSettingDialog, 1051
Ui_KbAnimWidget, 1089
Ui_KbLightWidget, 1104
Ui_KPerfWidget, 1131
Ui_MPerfWidget, 1164
Ui_RebindWidget, 1192
horizontalSpacer_6
Ui::AnimSettingDialog, 94
Ui::KbLightWidget, 462
Ui::KPerfWidget, 697
Ui::MPerfWidget, 774
Ui::RebindWidget, 962
Ui_AnimSettingDialog, 1051
Ui_KbLightWidget, 1104
Ui_KPerfWidget, 1131
Ui_MPerfWidget, 1164
Ui_RebindWidget, 1193
horizontalSpacer_7
Ui::KPerfWidget, 697
Ui::MPerfWidget, 775
Ui::RebindWidget, 963

Ui::SettingsWidget, 1033
Ui_KPerfWidget, 1131
Ui_MPerfWidget, 1164
Ui_RebindWidget, 1193
Ui_SettingsWidget, 1210
horizontalSpacer_8
Ui::KPerfWidget, 698
Ui::RebindWidget, 963
Ui_KPerfWidget, 1131
Ui_RebindWidget, 1193
horizontalSpacer_9
Ui::RebindWidget, 963
Ui::SettingsWidget, 1033
Ui_RebindWidget, 1193
Ui_SettingsWidget, 1210
hwIType
KbPerf, 537
hwLoading
Kb, 309
HwMode
KPerfWidget, 709
hwModeCount
Kb, 309
hwModified
UsbId, 1218
hwModifiedString
UsbId, 1216
hwProfile
Kb, 280
hwSave
Kb, 281
hwSaveButton
Ui::KbWidget, 600
Ui_KbWidget, 1117
hwload
devcmd.__unnamed__, 161
hwload_mode
ckb-daemon/main.c, 1404
device.c, 1236
usb.c, 1541
hwloadmode
profile_keyboard.c, 1501
hwloadprofile
profile.h, 1485
hwprofile, 1514
hwsave
devcmd.__unnamed__, 161
hwtonative
profile.c, 1477
profile.h, 1497

IAUTO
command.h, 1228
ID
command.h, 1228
IDLE
command.h, 1227
INOTIFY
command.h, 1228
IOFF
command.h, 1228
ION
command.h, 1228
IT
KeyMap, 638
I_CAPS
structures.h, 1520
I_COUNT
KbPerf, 537
KPerfWidget, 717
I_NUM
structures.h, 1520
I_SCROLL
structures.h, 1520
i_hw
KbPerf, 507
KPerfWidget, 708
iButton0
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1164
iButton1
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1164
iButton2
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1164
iButton3
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1165
iButton4
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1165
iButton5
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1165
iButtonO
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1165
iColor
KbPerf, 537
iEnable
KbPerf, 537
IFACE_MAX
structures.h, 1520
IN_CORSAIR
device.h, 1239
IN_HID
device.h, 1239
INDEX_OF
includes.h, 1302
iOpacity
KbPerf, 521, 522
IS_CONNECTED
device.h, 1240
IS_FULLRANGE
usb.h, 1546
IS_GLAIVE

usb.h, 1546
IS_HARPOON
usb.h, 1546
IS_K63
usb.h, 1546
IS_K65
usb.h, 1547
IS_K68
usb.h, 1547
IS_K70
usb.h, 1547
IS_K95
usb.h, 1547
IS_M65
usb.h, 1547
IS_MOD
input.h, 1320
IS_MONOCHROME
usb.h, 1547
IS_MONOCHROME_DEV
usb.h, 1547
IS_MOUSE
usb.h, 1547
IS_MOUSE_DEV
usb.h, 1548
IS_PLATINUM
usb.h, 1548
IS_RGB
usb.h, 1548
IS_RGB_DEV
usb.h, 1548
IS_SABRE
usb.h, 1548
IS_SCIMITAR
usb.h, 1548
IS_STRAFE
usb.h, 1548
IS_V2_OVERRIDE
usb.h, 1548
IS_WHEEL
input.c, 1304
iState
Kb, 309
iauto
devcmd.__unnamed__, 161
iconClicked
MainWindow, 745
id
devcmd.__unnamed__, 161
KbMode, 495, 496
KbProfile, 545
idle
devcmd.__unnamed__, 161
ids
fwupgradedialog.cpp, 1686
imutex
device.h, 1239
inFrame
AnimScript, 84
inbuf
ckb-mviz/main.c, 1422
includes.h
__FILE_NOPATH__, 1300
ckb_err, 1300
ckb_err_fn, 1300
ckb_err_nofile, 1301
ckb_fatal, 1301
ckb_fatal_fn, 1301
ckb_fatal_nofile, 1301
ckb_info, 1301
ckb_info_fn, 1301
ckb_info_nofile, 1301
ckb_s_err, 1301
ckb_s_out, 1301
ckb_warn, 1301
ckb_warn_fn, 1302
ckb_warn_nofile, 1302
INDEX_OF, 1302
timespec_add, 1303
timespec_eq, 1302
timespec_ge, 1302
timespec_gt, 1302
timespec_le, 1302
timespec_lt, 1302
uchar, 1303
ushort, 1303
indexOf
Kb, 282
KbProfile, 546
indicBox
Ui::MPerfWidget, 775
Ui_MPerfWidget, 1165
indicator
KbPerf, 508
KPerfWidget, 709
indicators
KPerfWidget, 717
infoTab
Ui::AnimSettingDialog, 94
Ui_AnimSettingDialog, 1052
infoUpdated
Kb, 283
init
AnimScript, 68
ColorMap, 157
KbManager, 483
init_keys
crypt.h, 1778
init_linkedlist
zip.c, 1910
initbind
input.c, 1308
input.h, 1322
initialized
AnimScript, 84
initmode

profile.c, 1478
inotify
devcmd.__unnamed__, 161
input.c
_cmd_macro, 1304
cmd_bind, 1306
cmd_macro, 1307
cmd_rebind, 1307
cmd_unbind, 1307
freebind, 1308
IS_WHEEL, 1304
initbind, 1308
inputupdate, 1309
inputupdate_keys, 1310
macro_pt_dequeue, 1312
macro_pt_enqueue, 1313
macro_pt_first, 1313
macromask, 1314
play_macro, 1314
pt_head, 1317
pt_tail, 1317
updateindicators_kb, 1316
input.h
cmd_bind, 1320
cmd_macro, 1321
cmd_rebind, 1321
cmd_unbind, 1321
freebind, 1322
IS_MOD, 1320
initbind, 1322
inputupdate, 1323
os_inputclose, 1324
os_inputopen, 1324
os_keypress, 1325
os_mousemove, 1326
os_setupindicators, 1327
parameter_t, 1320
ptlist_t, 1320
updateindicators_kb, 1328
input_linux.c
_ledthread, 1329
isync, 1330
os_inputclose, 1330
os_inputopen, 1331
os_keypress, 1332
os_mousemove, 1332
os_setupindicators, 1333
uinputopen, 1334
inputmutex
device.c, 1236
device.h, 1247
inputupdate
input.c, 1309
input.h, 1323
inputupdate_keys
input.c, 1310
insert
KbProfile, 546
instance
KbFirmware, 424
int1_int_none
device_vtable.c, 1256
int1_void_none
device_vtable.c, 1256
intensityBox
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1131
internal
QuaZipFilePrivate, 920
internalAttr
QuaZipFileInfo, 909
QuaZipFileInfo64, 916
QuaZipNewInfo, 930
ioDevice
QuaZipPrivate, 938
ioapi.h
_FILE_OFFSET_BIT, 1783
_LARGEFILE64_SOURCE, 1784
_Z_OF, 1786
__USE_LARGEFILE64, 1783
fill_zlib_filefunc64_32_def_from_filefunc32, 1786
OF, 1784
ZCALLBACK, 1784
ZCLOSE64, 1784
ZERROR64, 1784
ZFAKECLOSE64, 1784
ZOPEN64, 1785
ZPOS64_T, 1786
ZREAD64, 1785
ZSEEK64, 1785
ZTELL64, 1786
ZWRITE64, 1786
zlib_filefunc64_32_def, 1786
zlib_filefunc64_def, 1786
zlib_filefunc_def, 1786
ioff
devcmd.__unnamed__, 161
ion
devcmd.__unnamed__, 162
isActive
KbAnim, 322
isAnim
KeyAction, 616
isAutoClose
QuaZip, 853
isBusy
CkbSettings, 132
isDPI
KeyAction, 617
isDataDescriptorWritingEnabled
QuaZip, 853
isEnabled
AutoRun, 122
isEncrypted
QuaZipFileInfo64, 914
isISO

KbBind, 377
KeyMap, 647
isJP
KeyMap, 648
isKeyboard
Kb, 283
KbBind, 378
KeyMap, 649, 650
isMacro
KeyAction, 617
isMedia
KeyAction, 617
isMouse
Kb, 284
KbBind, 378
KeyAction, 618
KeyMap, 650, 651
isNormal
KeyAction, 618
isOpen
Kb, 285
QuaZip, 854
isPTBR
KeyMap, 651, 652
isProgram
KeyAction, 619
isRaw
QuaZipFile, 895
isRoot
QuaZipDir, 878
isRunning
KbAnim, 322
main.cpp, 1759
isSequential
QuaGzipFile, 823
QuaZIODevice, 830
QuaZipFile, 896
isSetting
MPerfWidget, 798
isSpecial
KeyAction, 619
isStarted
KbLight, 441
isUnbound
KeyAction, 620
isValidMacro
KeyAction, 620
isZip64Enabled
QuaZip, 854
isblack
led_mouse.c, 1392
iselect
led.c, 1344
isync
input_linux.c, 1330

JP
KeyMap, 638
JlCompress, 243
compressDir, 244
compressFile, 245, 246
compressFiles, 247
compressSubDir, 249
extractDir, 252
extractFile, 254, 256
extractFiles, 257
getFileList, 258
removeFile, 259
JlCompress.cpp
copyData, 1788

K63
KeyMap, 639
K63_HEIGHT
keymap.cpp, 1720
K63_TOP_COUNT
keymap.cpp, 1720
K63_WIDTH
keymap.cpp, 1721
K63TopRow
keymap.cpp, 1733
K65
KeyMap, 639
K65_HEIGHT
keymap.cpp, 1721
K65_TOP_COUNT
keymap.cpp, 1721
K65_WIDTH
keymap.cpp, 1721
K65TopRow
keymap.cpp, 1734
K68
KeyMap, 639
K68_HEIGHT
keymap.cpp, 1721
K68_TOP_COUNT
keymap.cpp, 1721
K68_WIDTH
keymap.cpp, 1721
K68TopRow
keymap.cpp, 1734
K70
KeyMap, 639
K70_HEIGHT
keymap.cpp, 1721
K70_WIDTH
keymap.cpp, 1721
K70_X_START
keymap.cpp, 1721
K95
KeyMap, 639
K95P
KeyMap, 639
K95_HEIGHT
keymap.cpp, 1722
K95_WIDTH
keymap.cpp, 1722
K95BaseMap

keymap.cpp, 1734
K95Keys
keymap.cpp, 1734
k95Label1
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1131
k95Label2
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1132
k95Label3
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1132
k95Label4
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1132
k95Label5
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1132
k95Label6
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1132
k95Line
Ui::KPerfWidget, 698
Ui_KPerfWidget, 1132
K95P_HEIGHT
keymap.cpp, 1722
K95P_WIDTH
keymap.cpp, 1722
K95P_X_START
keymap.cpp, 1722
K95PLbar
keymap.cpp, 1735
k95Spacer
Ui::KPerfWidget, 699
Ui_KPerfWidget, 1132
k95Widgets
KPerfWidget, 717
KPerfWidget
ALWAYS_OFF, 709
ALWAYS_ON, 709
BOTH, 709
NORMAL, 709
RGB, 709
KPerfWidget::IndicatorUi, 707, 1753
KB_NAME_LEN
structures.h, 1521
KEY_BACKSLASH_ISO
daemon/keymap.h, 1742
KEY_CORSAIR
daemon/keymap.h, 1742
KEY_NONE
daemon/keymap.h, 1742
KEY_SIZE
keywidget.cpp, 1750
KEY_UNBOUND
daemon/keymap.h, 1743
KEYCOUNT_GLAIVE
keymap.cpp, 1722
KEYCOUNT_HARPOON
keymap.cpp, 1722
KEYCOUNT_K95
keymap.cpp, 1722
KEYCOUNT_M65
keymap.cpp, 1722
KEYCOUNT_SABRE
keymap.cpp, 1723
KEYCOUNT_SCIM
keymap.cpp, 1723
KISS_FFT_COS
_kiss_fft_guts.h, 1613
KISS_FFT_FREE
kiss_fft.h, 1627
KISS_FFT_MALLOC
kiss_fft.h, 1627
KISS_FFT_SIN
_kiss_fft_guts.h, 1613
KP_NAME
AnimScript, 84
KP_NONE
AnimScript, 84
KP_POSITION
AnimScript, 84
kPerfTab
Ui::KbWidget, 600
Ui_KbWidget, 1117
KPerfWidget, 705
~KPerfWidget, 711
HW_I_COUNT, 716
HwMode, 709
I_COUNT, 717
i_hw, 708
indicator, 709
indicators, 717
k95Widgets, 717
KPerfWidget, 709
KPerfWidget, 709
mode2Raw, 711
on_copyButton_clicked, 712
on_intensityBox_valueChanged, 713
perf, 717
profile, 717
raw2Mode, 713
setPerf, 714
ui, 717
uiUpdated, 715
updateMapper, 717
kPerfWidget
Ui::KbWidget, 600
Ui_KbWidget, 1117
KSTRAFE_HEIGHT
keymap.cpp, 1723
KSTRAFE_WIDTH
keymap.cpp, 1723
KSTRAFE_X_START
keymap.cpp, 1723
KStrafeKeys

keymap.cpp, 1735
Kb, 261
~Kb, 266
_currentMode, 307
_currentProfile, 307
_delay, 307
_dither, 307
_frameRate, 307
_hwProfile, 307
_layout, 308
_model, 308
_mouseAccel, 308
_needsSave, 308
_profiles, 308
_scrollSpeed, 308
appendProfile, 269
autoSave, 269
cmd, 308
cmdpath, 308
currentBind, 269
currentLight, 270
currentMode, 271
currentPerf, 271
currentProfile, 272
deleteHw, 272
deletePrevious, 273
devpath, 308
dither, 273, 274
features, 309
find, 274
firmware, 309
frameRate, 275
frameUpdate, 276
fwUpdPath, 309
fwUpdate, 277
fwUpdateFinished, 278
fwUpdateProgress, 278
getKeyMap, 279
getMacroNumber, 279
getMacroPath, 279
HWMODE_MAX, 309
hwLoading, 309
hwModeCount, 309
hwProfile, 280
hwSave, 281
iState, 309
indexOf, 282
infoUpdated, 283
isKeyboard, 283
isMouse, 284
isOpen, 285
Kb, 266
KbManager, 307
lastAutoSave, 309
layout, 285
load, 286
macroDelay, 287
macroNumber, 309
macroPath, 310
matches, 288
modeChanged, 288
modeRenamed, 289
model, 288
monochrome, 310
mouseAccel, 289
needsSave, 290
newMode, 290, 291
newProfile, 291, 292
notifyNumber, 310
notifyPath, 310
pollrate, 310
prefsPath, 310
prevMode, 310
prevProfile, 310
profileAdded, 293
profileChanged, 293
profileRenamed, 293
profiles, 294
readNotify, 295
run, 299
save, 300
scrollSpeed, 301
setCurrentMode, 302, 303
setCurrentProfile, 304
updateLayout, 305
usbModel, 310
usbSerial, 311
writeProfileHeader, 306
kb
FwUpgradeDialog, 193
kb.cpp
activeDevices, 1692
notifyPathMutex, 1692
notifyPaths, 1692
KbAnim
Add, 314
Divide, 315
Multiply, 314
Normal, 314
Subtract, 314
KbFirmware
NO, 416
UNKNOWN, 416
YES, 416
KbId, 1684
KbPerf
CAPS, 508
HIGH, 507
HW_IMAX, 508
LIGHT, 508
LOCK, 508
LOW, 507
LOWMED, 507
MACRO, 508
MEDHIGH, 507
MEDIUM, 507

MODE, 508
MUTE, 508
NONE, 508
NORMAL, 508
NUM, 508
OFF, 508
ON, 508
SCROLL, 508
KbAnim, 311
_guid, 336
_isActive, 336
_isActiveKp, 336
_keys, 336
_map, 336
_mode, 336
_name, 336
_needsSave, 336
_opacity, 337
_parameters, 337
_script, 337
_scriptGuid, 337
_scriptName, 337
_tempParameters, 337
blend, 317
catchUp, 318
commitParams, 320
effectiveParams, 320
guid, 321
hasParameter, 321
isActive, 322
isRunning, 322
KbAnim, 315–317
KbAnim, 315–317
keypress, 323
keys, 324, 325
kpRepeatMsec, 337
kpRepeatTime, 337
kpStopTime, 337
map, 325, 326
Mode, 314
mode, 326
name, 326, 327
needsSave, 327
newId, 328
opacity, 328
parameter, 328, 329
reInit, 330
repeatKey, 338
repeatMsec, 338
repeatTime, 338
resetParams, 330
save, 331
script, 332
scriptName, 332
stop, 333
stopTime, 338
trigger, 334
updateParams, 335
KbAnimWidget, 348
~KbAnimWidget, 351
addAnim, 351
animChanged, 352
animations, 365
clearSelection, 353
current, 365
didUpdateSelection, 353
duplicateAnim, 354
KbAnimWidget, 350
KbAnimWidget, 350
light, 365
noReorder, 365
on_animList_currentItemChanged, 354
on_animList_customContextMenuRequested, 355
on_animList_itemChanged, 356
on_blendBox_activated, 356
on_deleteButton_clicked, 357
on_keyButton_clicked, 357
on_nameBox_textEdited, 358
on_opacityBox_valueChanged, 359
on_propertyButton_clicked, 359
refreshList, 360
reorderAnims, 361
selectedKeys, 365
setCurrent, 362
setLight, 363
setSelectedKeys, 364
ui, 365
KbBind, 366
_bind, 395
_devParent, 395
_globalRemap, 395
_map, 395
_needsSave, 395
_needsUpdate, 396
_winLock, 396
action, 370
bindAction, 371
defaultAction, 371, 372
devParent, 372
didLoad, 373
friendlyActionName, 373
friendlyName, 374
getMacroNumber, 374
getMacroPath, 375
globalRemap, 376
globalRemapTime, 396
handleNotificationChannel, 376
isISO, 377
isKeyboard, 378
isMouse, 378
KbBind, 369
KbBind, 369
KeyAction, 395
keyEvent, 379
lastCmd, 396
lastGlobalRemapTime, 396

layoutChanged, 380
light, 381
load, 381
loadGlobalRemap, 383
map, 383, 384
modeParent, 384
needsSave, 384
noAction, 385
perf, 386
resetAction, 387
save, 388
saveGlobalRemap, 389
setAction, 390
setGlobalRemap, 391
setNeedsUpdate, 392
update, 392
updated, 394
winLock, 394, 395
KbBindWidget, 403
~KbBindWidget, 405
bind, 413
currentSelection, 413
KbBindWidget, 405
KbBindWidget, 405
newLayout, 406
newSelection, 406
on_copyButton_clicked, 408
on_resetButton_clicked, 408
profile, 413
setBind, 409
ui, 413
updateBind, 410
updateSelDisplay, 412
kbConnected
KbManager, 484
kbDisconnected
KbManager, 484
KbFirmware, 413
~KbFirmware, 416
_checkUpdates, 416
_fileForBoard, 417
_latestForBoard, 418
checkUpdates, 419
dataForBoard, 419
downloadFinished, 420
downloaded, 420
fwTable, 424
fwTableHash, 424
hasDownloaded, 421
hasGPG, 424
instance, 424
KbFirmware, 416
KbFirmware, 416
lastCheck, 424
lastFinished, 424
networkManager, 424
processDownload, 421
tableDownload, 424
versionForBoard, 423
KbFirmware::FW, 183
ckbVersion, 184
FW, 184
fileName, 184
fwVersion, 184
hash, 184
url, 184
KbLight, 425
~KbLight, 430
_animList, 453
_animMap, 453
_colorMap, 454
_dimming, 454
_indicatorList, 454
_indicatorMap, 454
_lastFrameAnimMap, 454
_lastFrameDimming, 454
_lastFrameIndicatorMap, 454
_map, 454
_needsMapRefresh, 454
_needsSave, 454
_previewAnim, 455
_qColorMap, 455
_start, 455
addAnim, 430
animKeypress, 431
AnimList, 429
animList, 432
base, 432
close, 433
color, 434, 435
colorMap, 435
didLoad, 436
dimming, 436, 437
duplicateAnim, 437
findAnim, 438
findAnimIdx, 439
frameDisplayed, 439
frameUpdate, 440
isStarted, 441
KbLight, 429
KbLight, 429
lastFrameSignal, 455
load, 442
MAX_DIM, 455
map, 443, 444
needsSave, 444
open, 445
previewAnim, 446
printRGB, 447
rebuildBaseMap, 448
resetIndicators, 448
restartAnimation, 449
save, 450
setIndicator, 451
shareDimming, 451, 452
stopPreview, 452

updated, 453
KbLightWidget, 464
~KbLightWidget, 466
changeAnim, 467
changeAnimKeys, 467
changeColor, 468
currentSelection, 476
KbLightWidget, 466
KbLightWidget, 466
light, 476
newSelection, 469
on_animButton_clicked, 470
on_bgButton_clicked, 471
on_brightnessBox_activated, 471
on_showAnimBox_clicked, 472
setLight, 472
setMonochrome, 473
toggleSidelight, 474
ui, 476
updateLight, 475
KbManager, 477
_daemonVersion, 488
_devices, 488
_eventTimer, 488
_guiVersion, 488
_kbManager, 489
_scanTimer, 489
ckbDaemonVersion, 480
ckbDaemonVersionF, 480
ckbGuiVersion, 481
ckbGuiVersionF, 481
devices, 482
eventTimer, 482
fps, 482
init, 483
Kb, 307
kbConnected, 484
kbDisconnected, 484
KbManager, 479
kbManager, 484
KbManager, 479
parseVersionString, 485
scanKeyboards, 485
scanTimer, 487
stop, 487
versionUpdated, 488
kbManager
KbManager, 484
KbMode, 489
~KbMode, 494
_bind, 503
_id, 503
_light, 503
_name, 503
_needsSave, 503
_perf, 503
bind, 494
doUpdate, 495
id, 495, 496
KbMode, 492, 493
KbMode, 492, 493
keyMap, 496
light, 497
name, 497, 498
needsSave, 498
newId, 499
perf, 500
save, 500
setNeedsSave, 501
setNeedsUpdate, 501
updated, 502
KbPerf, 503
_angleSnap, 535
_curDpi, 510
_dpiIndicator, 535
_iOpacity, 535
_liftHeight, 536
_needsSave, 536
_needsUpdate, 536
angleSnap, 510, 511
applyIndicators, 511
baseDpiIdx, 512, 513
bind, 513
DPI_COUNT, 536
DPI_MAX, 536
DPI_MIN, 536
didLoad, 514
dpi, 514, 515
dpiBaseIdx, 536
dpiClr, 536
dpiColor, 516, 517
dpiCurX, 536
dpiCurY, 537
dpiCycleDown, 517
dpiCycleUp, 518
dpiDown, 518
dpiEnabled, 519
dpiIndicator, 520
dpiOn, 537
dpiUp, 520
dpiX, 537
dpiY, 537
getIndicator, 521
HW_I_COUNT, 537
height, 507
hwIType, 537
I_COUNT, 537
i_hw, 507
iColor, 537
iEnable, 537
iOpacity, 521, 522
indicator, 508
KbPerf, 508, 509
KbPerf, 508, 509
liftHeight, 522
light, 522

light100Color, 538
lightIndicator, 523
load, 524
modeParent, 526
muteNAColor, 538
needsSave, 526
OTHER, 538
operator=, 527
popDpi, 527
pushDpi, 528, 529
pushSniper, 530
pushedDpis, 538
runningPushIdx, 538
SNIPER, 538
save, 530
setIndicator, 531
setNeedsUpdate, 532
settingsUpdated, 532
sniperDpi, 532, 533
update, 534
KbProfile, 538
_currentMode, 555
_id, 555
_keyMap, 555
_modes, 555
_name, 555
_needsSave, 556
append, 543
currentMode, 543, 544
find, 544
id, 545
indexOf, 546
insert, 546
KbProfile, 541, 542
KbProfile, 541, 542
keyMap, 547
modeCount, 548
ModeList, 541
modes, 548, 549
move, 549
name, 550
needsSave, 550
newId, 551
removeAll, 552
save, 553
setNeedsSave, 554
setNeedsUpdate, 554
KbProfileDialog, 556
~KbProfileDialog, 558
addNewProfileItem, 558
device, 564
GUID, 565
KbProfileDialog, 558
KbProfileDialog, 558
NEW_FLAG, 565
on_profileList_currentItemChanged, 559
on_profileList_customContextMenuRequested, 559
on_profileList_itemChanged, 561
on_profileList_itemClicked, 562
profileList_reordered, 562
repopulate, 563
ui, 565
KbWidget, 570
~KbWidget, 574
addNewModeItem, 574
currentMode, 591
devUpdate, 575
device, 591
GUID, 592
hasShownNewFW, 592
KbWidget, 572
KbWidget, 572
lastAutoSave, 592
modeChanged, 575
modeIcon, 577
modeUpdate, 579
modesList_reordered, 578
NEW_FLAG, 592
name, 579
on_fwUpdButton_clicked, 579
on_hwSaveButton_clicked, 581
on_modesList_currentItemChanged, 582
on_modesList_customContextMenuRequested, 583
on_modesList_itemChanged, 584
on_modesList_itemClicked, 585
on_profileBox_activated, 586
on_tabWidget_currentChanged, 586
profileChanged, 587
showFirstTab, 588
showFwUpdate, 588
showLastTab, 589
ui, 592
updateFwButton, 589
updateProfileList, 590
kbWidgets
MainWindow, 752
kbanim.cpp
blendAdd, 1694
blendDivide, 1694
blendFunc, 1694
blendMultiply, 1694
blendNormal, 1694
blendSubtract, 1695
functions, 1695
kbfirmware.cpp
AUTO_CHECK_TIME, 1701
tableName, 1701
kblight.cpp
_shareDimming, 1705
activeLights, 1705
lToS, 1703
monoRgb, 1703
sToL, 1704
kbmanager.cpp

devpath, 1708
kbperf.cpp
map_last, 1710
kbsize
ckb-ripple/main.c, 1444
kbsyspath
usb_linux.c, 1608
Key, 602
_friendlyName, 604
_storageName, 604
friendlyName, 603
hasLed, 604
hasScan, 604
height, 604
name, 605
operator bool, 603
storageName, 604
width, 605
x, 605
y, 605
key, 1741
KeyMap, 652
KeyAction
NORMAL, 609
SPECIAL, 609
UNBOUND, 609
keyAnim, 1405
KeyMap
_LAYOUT_MAX, 638
_MODEL_MAX, 639
DE, 638
DK, 638
ES, 638
EU, 638
EU_DVORAK, 638
FR, 638
GB, 638
GB_DVORAK, 638
GLAIVE, 639
HARPOON, 639
IT, 638
JP, 638
K63, 639
K65, 639
K68, 639
K70, 639
K95, 639
K95P, 639
M65, 639
MX, 638
NO, 638
NO_LAYOUT, 638
NO_MODEL, 639
PL, 638
PT_BR, 638
SABRE, 639
SCIMITAR, 639
SE, 638
STRAFE, 639
US, 638
US_DVORAK, 638
KeyPatch, 1719
KeyWidget
ADD, 665
NONE, 665
SET, 665
SUBTRACT, 665
TOGGLE, 665
KeyAction, 605
~KeyAction, 610
_value, 632
adjustDisplay, 610
animAction, 611
animInfo, 612
DPI_CUSTOM, 632
DPI_CYCLE_DOWN, 633
DPI_CYCLE_UP, 633
DPI_DOWN, 633
DPI_SNIPER, 633
DPI_UP, 633
defaultAction, 612
dpiAction, 613
dpiInfo, 614
driverName, 614
friendlyName, 615
isAnim, 616
isDPI, 617
isMacro, 617
isMedia, 617
isMouse, 618
isNormal, 618
isProgram, 619
isSpecial, 619
isUnbound, 620
isValidMacro, 620
KbBind, 395
KeyAction, 610
keyEvent, 621
keyRelease, 624
KeyAction, 610
LIGHT_DOWN, 633
LIGHT_DOWN_WRAP, 633
LIGHT_UP, 633
LIGHT_UP_WRAP, 633
LOCK_OFF, 634
LOCK_ON, 634
LOCK_TOGGLE, 634
lightAction, 625
lockAction, 625
MODE_NEXT, 634
MODE_NEXT_WRAP, 634
MODE_PREV, 634
MODE_PREV_WRAP, 634
macroAction, 625
macroContent, 626
macroDisplay, 626

macroFullLine, 627
macroLine, 628
macroTiming, 628
modeAction, 629
noAction, 629
operator QString, 630
operator=, 630
PROGRAM_PR_INDEF, 634
PROGRAM_PR_KPSTOP, 635
PROGRAM_PR_KRSTOP, 635
PROGRAM_PR_MULTI, 635
PROGRAM_RE_INDEF, 635
PROGRAM_RE_KPSTOP, 635
PROGRAM_RE_MULTI, 635
preProgram, 634
programAction, 630
programInfo, 630
relProgram, 635
sniperValue, 635
specialInfo, 631
Type, 609
type, 631
value, 632
keyButton
Ui::KbAnimWidget, 346
Ui_KbAnimWidget, 1089
keyEvent
KbBind, 379
KeyAction, 621
keyHeight
KeyMap, 660
keyLayout
KeyMap, 660
KeyMap, 636
_keys, 660
byPosition, 640
contains, 640
count, 641
friendlyName, 641
fromName, 643
fromStorage, 644
getLayout, 644, 645
getModel, 646
height, 647
isISO, 647
isJP, 648
isKeyboard, 649, 650
isMouse, 650, 651
isPTBR, 651, 652
key, 652
keyHeight, 660
keyLayout, 660
KeyMap, 639, 640
keyModel, 661
keyWidth, 661
KeyMap, 639, 640
keys, 653
Layout, 638
layout, 653
layoutNames, 654
locale, 654
map, 655
Model, 638
model, 655
modelHeight, 656
modelWidth, 656
name, 657
operator const QHash< QString, Key > &, 657
positions, 658
strLayout, 658
strModel, 659
toStorage, 659
width, 660
keyMap
KbMode, 496
KbProfile, 547
KeyWidget, 683
keyModel
KeyMap, 661
keyNames
ColorMap, 158
keyRelease
KeyAction, 624
keyTab
Ui::RebindWidget, 963
Ui_RebindWidget, 1193
KeyWidget, 661
_bindMap, 682
_colorMap, 682
_displayColorMap, 683
_indicators, 683
_monochrome, 683
_rgbMode, 683
animation, 683
BindMap, 664
bindMap, 665
clearAnimation, 666
clearSelection, 666
colorMap, 666, 667
displayColorMap, 667
drawInfo, 667
keyMap, 683
KeyWidget, 665
KeyWidget, 665
map, 668, 669
monochrome, 669, 670
mouseCurrentX, 683
mouseCurrentY, 683
mouseDownMode, 683
mouseDownX, 684
mouseDownY, 684
mouseMoveEvent, 670
mousePressEvent, 671
mouseReleaseEvent, 672
newSelection, 684
paintEvent, 673

rgbMode, 678, 679
selectAll, 679
selection, 684
selectionChanged, 680
setAnimation, 680
setAnimationToSelection, 681
setSelection, 681
sidelightToggled, 682
keyWidget
Ui::KbBindWidget, 401
Ui::KbLightWidget, 462
Ui_KbBindWidget, 1096
Ui_KbLightWidget, 1104
keyWidth
KeyMap, 661
keyaction.cpp
ANIM_KRSTOP, 1716
ANIM_ONCE, 1716
keyanim, 1411
keyboard
device.c, 1236
device.h, 1247
keymacro, 1508
keymap
daemon/keymap.h, 1748
keymap.c, 1340
keymap.c
BUTTON_HID_COUNT, 1336
corsair_kbcopy, 1336
corsair_mousecopy, 1337
hid_kb_translate, 1337
hid_mouse_translate, 1339
keymap, 1340
keymap.cpp
_patch, 1724
ANSI_ENTER_H, 1719
ANSI_ENTER_W, 1720
ANSI_ENTER_X, 1720
ANSI_ENTER_Y, 1720
ANSI_LSHIFT_W, 1720
ANSI_LSHIFT_X, 1720
GLAIVE_HEIGHT, 1720
GLAIVE_WIDTH, 1720
getMap, 1725
GlaiveKeys, 1733
HARPOON_HEIGHT, 1720
HARPOON_WIDTH, 1720
HarpoonKeys, 1733
K63_HEIGHT, 1720
K63_TOP_COUNT, 1720
K63_WIDTH, 1721
K63TopRow, 1733
K65_HEIGHT, 1721
K65_TOP_COUNT, 1721
K65_WIDTH, 1721
K65TopRow, 1734
K68_HEIGHT, 1721
K68_TOP_COUNT, 1721
K68_WIDTH, 1721
K68TopRow, 1734
K70_HEIGHT, 1721
K70_WIDTH, 1721
K70_X_START, 1721
K95_HEIGHT, 1722
K95_WIDTH, 1722
K95BaseMap, 1734
K95Keys, 1734
K95P_HEIGHT, 1722
K95P_WIDTH, 1722
K95P_X_START, 1722
K95PLbar, 1735
KEYCOUNT_GLAIVE, 1722
KEYCOUNT_HARPOON, 1722
KEYCOUNT_K95, 1722
KEYCOUNT_M65, 1722
KEYCOUNT_SABRE, 1723
KEYCOUNT_SCIM, 1723
KSTRAFE_HEIGHT, 1723
KSTRAFE_WIDTH, 1723
KSTRAFE_X_START, 1723
KStrafeKeys, 1735
LBARCOUNT_K95P, 1723
LBS, 1723
M65_HEIGHT, 1723
M65_WIDTH, 1723
M65Keys, 1736
N_LAYOUTS, 1723
N_MODELS, 1724
NS, 1724
PATCH_COUNT, 1724
patch, 1724
patchABNT2, 1729
patchANSI, 1730
patchDE, 1736
patchDK, 1736
patchDvorak, 1736
patchES, 1736
patchEU, 1737
patchFR, 1737
patchISO, 1730
patchIT, 1737
patchJP, 1737
patchJP106, 1731
patchJP106fn, 1732
patchMX, 1737
patchNO, 1738
patchPTBR, 1738
patchSE, 1738
patchnonJP106, 1732
SABRE_HEIGHT, 1724
SABRE_WIDTH, 1724
SCIM_HEIGHT, 1724
SCIM_WIDTH, 1724
SabreKeys, 1738
ScimKeys, 1739
standardMaps, 1739

keypress
AnimScript, 69
KbAnim, 323
keys
KbAnim, 324, 325
KeyMap, 653
keywidget.cpp
glaiveOverlay, 1750
harpOverlay, 1750
KEY_SIZE, 1750
m65Overlay, 1750
monoRgb, 1749
sabOverlay, 1750
scimOverlay, 1751
kf_bfly2
kiss_fft.c, 1615
kf_bfly3
kiss_fft.c, 1615
kf_bfly4
kiss_fft.c, 1616
kf_bfly5
kiss_fft.c, 1617
kf_bfly_generic
kiss_fft.c, 1618
kf_cexp
_kiss_fft_guts.h, 1613
kf_factor
kiss_fft.c, 1619
kf_work
kiss_fft.c, 1620
kiss_fft_cpx, 1627
kiss_fft_state, 1611
kiss_fftr_state, 1633
kiss_fft
kiss_fft.c, 1621
kiss_fft.h, 1628
kiss_fft.c
kf_bfly2, 1615
kf_bfly3, 1615
kf_bfly4, 1616
kf_bfly5, 1617
kf_bfly_generic, 1618
kf_factor, 1619
kf_work, 1620
kiss_fft, 1621
kiss_fft_alloc, 1622
kiss_fft_cleanup, 1623
kiss_fft_next_fast_size, 1624
kiss_fft_stride, 1624
kiss_fft.h
KISS_FFT_FREE, 1627
KISS_FFT_MALLOC, 1627
kiss_fft, 1628
kiss_fft_alloc, 1629
kiss_fft_cfg, 1628
kiss_fft_cleanup, 1630
kiss_fft_free, 1627
kiss_fft_next_fast_size, 1630
kiss_fft_scalar, 1628
kiss_fft_stride, 1631
kiss_fftr_next_fast_size_real, 1628
kiss_fft_alloc
kiss_fft.c, 1622
kiss_fft.h, 1629
kiss_fft_cfg
kiss_fft.h, 1628
kiss_fft_cleanup
kiss_fft.c, 1623
kiss_fft.h, 1630
kiss_fft_free
kiss_fft.h, 1627
kiss_fft_next_fast_size
kiss_fft.c, 1624
kiss_fft.h, 1630
kiss_fft_scalar
kiss_fft.h, 1628
kiss_fft_stride
kiss_fft.c, 1624
kiss_fft.h, 1631
kiss_fftr
kiss_fftr.c, 1634
kiss_fftr.h, 1638
kiss_fftr.c
kiss_fftr, 1634
kiss_fftr_alloc, 1635
kiss_fftri, 1636
kiss_fftr.h
kiss_fftr, 1638
kiss_fftr_alloc, 1639
kiss_fftr_cfg, 1638
kiss_fftr_free, 1638
kiss_fftri, 1640
kiss_fftr_alloc
kiss_fftr.c, 1635
kiss_fftr.h, 1639
kiss_fftr_cfg
kiss_fftr.h, 1638
kiss_fftr_free
kiss_fftr.h, 1638
kiss_fftr_next_fast_size_real
kiss_fft.h, 1628
kiss_fftri
kiss_fftr.c, 1636
kiss_fftr.h, 1640
kpDelayBox
Ui::AnimSettingDialog, 94
Ui_AnimSettingDialog, 1052
kpModeStopBox
Ui::AnimSettingDialog, 95
Ui_AnimSettingDialog, 1052
kpReleaseBox
Ui::AnimSettingDialog, 95
Ui_AnimSettingDialog, 1052
kpRepeatBox
Ui::AnimSettingDialog, 95
Ui_AnimSettingDialog, 1052

kpRepeatLabel
Ui::AnimSettingDialog, 95
Ui_AnimSettingDialog, 1052
kpRepeatLabel_2
Ui::AnimSettingDialog, 95
Ui_AnimSettingDialog, 1052
kpRepeatMsec
KbAnim, 337
kpRepeatTime
KbAnim, 337
kpStopCheck
AnimSettingDialog, 119
kpStopTime
KbAnim, 337
kphold
ckb-gradient/main.c, 1410
kprelease
ckb-gradient/main.c, 1410
ckb-ripple/main.c, 1444
ckb-wave/main.c, 1452

LAYOUT
command.h, 1227
LIFT
command.h, 1228
LIGHT
KbPerf, 508
LOCK
KbPerf, 508
LOW
KbPerf, 507
LOWMED
KbPerf, 507
LBARCOUNT_K95P
keymap.cpp, 1723
LBS
keymap.cpp, 1723
LED_DPI
daemon/keymap.h, 1743
LED_MOUSE
daemon/keymap.h, 1743
lHeightBox
Ui::MPerfWidget, 777
Ui_MPerfWidget, 1167
LIFT_MAX
structures.h, 1521
LIFT_MIN
structures.h, 1521
LIGHT_DOWN
KeyAction, 633
LIGHT_DOWN_WRAP
KeyAction, 633
LIGHT_UP
KeyAction, 633
LIGHT_UP_WRAP
KeyAction, 633
LOCALHEADERMAGIC
zip.c, 1905
LOCK_OFF
KeyAction, 634
LOCK_ON
KeyAction, 634
LOCK_TOGGLE
KeyAction, 634
lToS
kblight.cpp, 1703
label
Ui::AnimAddDialog, 47
Ui::AnimSettingDialog, 95
Ui::ExtraSettingsWidget, 171
Ui::GradientDialog, 212
Ui::KbBindWidget, 401
Ui::KbLightWidget, 462
Ui::KPerfWidget, 699
Ui::LayoutDialog, 721
Ui::ModeSelectDialog, 756
Ui::MPerfWidget, 776
Ui::RebindWidget, 963
Ui::SettingsWidget, 1033
Ui_AnimAddDialog, 1043
Ui_AnimSettingDialog, 1052
Ui_ExtraSettingsWidget, 1065
Ui_GradientDialog, 1080
Ui_KbBindWidget, 1096
Ui_KbLightWidget, 1104
Ui_KPerfWidget, 1132
Ui_LayoutDialog, 1142
Ui_ModeSelectDialog, 1150
Ui_MPerfWidget, 1165
Ui_RebindWidget, 1193
Ui_SettingsWidget, 1210
label_10
Ui::AnimSettingDialog, 95
Ui::ExtraSettingsWidget, 172
Ui::KPerfWidget, 699
Ui_AnimSettingDialog, 1052
Ui_ExtraSettingsWidget, 1065
Ui_KPerfWidget, 1132
label_11
Ui::AnimSettingDialog, 95
Ui::ExtraSettingsWidget, 172
Ui::KPerfWidget, 699
Ui::MPerfWidget, 776
Ui_AnimSettingDialog, 1053
Ui_ExtraSettingsWidget, 1065
Ui_KPerfWidget, 1133
Ui_MPerfWidget, 1165
label_12
Ui::AnimSettingDialog, 95
Ui::KPerfWidget, 699
Ui::MPerfWidget, 776
Ui_AnimSettingDialog, 1053
Ui_KPerfWidget, 1133
Ui_MPerfWidget, 1165
label_13
Ui::AnimSettingDialog, 96
Ui::MPerfWidget, 776

Ui_AnimSettingDialog, 1053
Ui_MPerfWidget, 1165
label_14
Ui::KPerfWidget, 699
Ui::MPerfWidget, 776
Ui_KPerfWidget, 1133
Ui_MPerfWidget, 1166
label_15
Ui::KPerfWidget, 699
Ui::MPerfWidget, 776
Ui_KPerfWidget, 1133
Ui_MPerfWidget, 1166
label_16
Ui::KPerfWidget, 699
Ui_KPerfWidget, 1133
label_17
Ui::KPerfWidget, 699
Ui_KPerfWidget, 1133
label_18
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1133
label_19
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1133
label_2
Ui::AnimAddDialog, 47
Ui::AnimSettingDialog, 96
Ui::ExtraSettingsWidget, 172
Ui::FwUpgradeDialog, 199
Ui::GradientDialog, 212
Ui::KbBindWidget, 402
Ui::KbLightWidget, 463
Ui::KPerfWidget, 700
Ui::LayoutDialog, 722
Ui::MPerfWidget, 776
Ui::SettingsWidget, 1033
Ui_AnimAddDialog, 1043
Ui_AnimSettingDialog, 1053
Ui_ExtraSettingsWidget, 1065
Ui_FwUpgradeDialog, 1074
Ui_GradientDialog, 1080
Ui_KbBindWidget, 1096
Ui_KbLightWidget, 1104
Ui_KPerfWidget, 1133
Ui_LayoutDialog, 1143
Ui_MPerfWidget, 1166
Ui_SettingsWidget, 1210
label_20
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1134
label_21
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1134
label_22
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1134
label_23
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1134
label_24
Ui::KPerfWidget, 700
Ui_KPerfWidget, 1134
label_25
Ui::MPerfWidget, 776
Ui_MPerfWidget, 1166
label_3
Ui::AnimSettingDialog, 96
Ui::ExtraSettingsWidget, 172
Ui::KbLightWidget, 463
Ui::KbWidget, 600
Ui::KPerfWidget, 700
Ui::MPerfWidget, 776
Ui::SettingsWidget, 1033
Ui_AnimSettingDialog, 1053
Ui_ExtraSettingsWidget, 1065
Ui_KbLightWidget, 1104
Ui_KbWidget, 1117
Ui_KPerfWidget, 1134
Ui_MPerfWidget, 1166
Ui_SettingsWidget, 1210
label_4
Ui::AnimSettingDialog, 96
Ui::FwUpgradeDialog, 199
Ui::KbWidget, 600
Ui::KPerfWidget, 701
Ui::MPerfWidget, 777
Ui_AnimSettingDialog, 1053
Ui_FwUpgradeDialog, 1074
Ui_KbWidget, 1117
Ui_KPerfWidget, 1134
Ui_MPerfWidget, 1166
label_5
Ui::AnimSettingDialog, 96
Ui::ExtraSettingsWidget, 172
Ui::KbAnimWidget, 346
Ui::KPerfWidget, 701
Ui::MPerfWidget, 777
Ui::SettingsWidget, 1033
Ui_AnimSettingDialog, 1053
Ui_ExtraSettingsWidget, 1066
Ui_KbAnimWidget, 1089
Ui_KPerfWidget, 1134
Ui_MPerfWidget, 1166
Ui_SettingsWidget, 1210
label_6
Ui::AnimSettingDialog, 96
Ui::ExtraSettingsWidget, 172
Ui::KbAnimWidget, 346
Ui::KPerfWidget, 701
Ui::MPerfWidget, 777
Ui::SettingsWidget, 1034
Ui_AnimSettingDialog, 1053
Ui_ExtraSettingsWidget, 1066
Ui_KbAnimWidget, 1089
Ui_KPerfWidget, 1134
Ui_MPerfWidget, 1166

Ui_SettingsWidget, 1210
label_7
Ui::AnimSettingDialog, 96
Ui::KbAnimWidget, 346
Ui::KPerfWidget, 701
Ui::MPerfWidget, 777
Ui_AnimSettingDialog, 1053
Ui_KbAnimWidget, 1089
Ui_KPerfWidget, 1135
Ui_MPerfWidget, 1166
label_8
Ui::AnimSettingDialog, 96
Ui::KPerfWidget, 701
Ui::MPerfWidget, 777
Ui::SettingsWidget, 1034
Ui_AnimSettingDialog, 1054
Ui_KPerfWidget, 1135
Ui_MPerfWidget, 1167
Ui_SettingsWidget, 1211
label_9
Ui::AnimSettingDialog, 96
Ui::KbWidget, 600
Ui_AnimSettingDialog, 1054
Ui_KbWidget, 1117
lastAutoSave
Kb, 309
KbWidget, 592
lastCheck
KbFirmware, 424
lastCmd
KbBind, 396
lastDuration
AnimSettingDialog, 119
lastFinished
KbFirmware, 424
lastFrame
AnimScript, 84
lastFrameSignal
KbLight, 455
lastGlobalRemapTime
KbBind, 396
lastKnown
media_linux.cpp, 1769
lastMappedDirectoryEntry
QuaZipPrivate, 938
lastSharedDimming
extrasettingswidget.cpp, 1682
Layout
KeyMap, 638
layout
Kb, 285
KeyMap, 653
layoutBox
Ui::SettingsWidget, 1034
Ui_SettingsWidget, 1211
layoutChanged
KbBind, 380
LayoutDialog, 722
~LayoutDialog, 724
LayoutDialog, 724
LayoutDialog, 724
selected, 725
ui, 725
layoutNames
KeyMap, 654
layoutWidget
Ui::RebindWidget, 963
Ui_RebindWidget, 1193
layoutWidget1
Ui::RebindWidget, 963
Ui_RebindWidget, 1193
layoutWidget2
Ui::RebindWidget, 963
Ui_RebindWidget, 1193
layoutWidget_2
Ui::RebindWidget, 963
Ui_RebindWidget, 1194
lbl_macro
Ui::RebindWidget, 964
Ui_RebindWidget, 1194
led.c
cmd_iauto, 1341
cmd_inotify, 1342
cmd_ioff, 1342
cmd_ion, 1343
cmd_rgb, 1343
has_key, 1344
iselect, 1344
printrgb, 1345
led.h
cmd_iauto, 1347
cmd_inotify, 1348
cmd_ioff, 1348
cmd_ion, 1349
cmd_rgb, 1349
loadrgb_kb, 1350
loadrgb_mouse, 1352
printrgb, 1352
savergb_kb, 1354
savergb_mouse, 1355
updatergb_kb, 1356
updatergb_mouse, 1357
led_keyboard.c
BR1, 1381
BR2, 1381
BR4, 1381
bit_reverse_table, 1390
loadrgb_kb, 1382
makergb_512, 1384
makergb_full, 1385
O0, 1381
O1, 1381
O2, 1381
O3, 1382
O4, 1382
O5, 1382

O6, 1382
O7, 1382
O8, 1382
ordered8to3, 1385
quantize8to3, 1386
rgbcmp, 1386
savergb_kb, 1387
updatergb_kb, 1388
led_mouse.c
isblack, 1392
loadrgb_mouse, 1393
rgbcmp, 1393
savergb_mouse, 1394
updatergb_mouse, 1394
left
ckb-wave/main.c, 1452
license
AnimScript, 70
lift
devcmd.__unnamed__, 162
liftHeight
KbPerf, 522
light
KbAnimWidget, 365
KbBind, 381
KbLightWidget, 476
KbMode, 497
KbPerf, 522
light100Color
KbPerf, 538
lightAction
KeyAction, 625
lightBox
Ui::KPerfWidget, 701
Ui::RebindWidget, 964
Ui_KPerfWidget, 1135
Ui_RebindWidget, 1194
lightButton
Ui::RebindWidget, 964
Ui_RebindWidget, 1194
lightColor1
Ui::KPerfWidget, 701
Ui_KPerfWidget, 1135
lightColor2
Ui::KPerfWidget, 701
Ui_KPerfWidget, 1135
lightColor3
Ui::KPerfWidget, 701
Ui_KPerfWidget, 1135
lightIndicator
KbPerf, 523
lightTab
Ui::KbWidget, 600
Ui_KbWidget, 1117
lightWidget
Ui::KbWidget, 600
Ui_KbWidget, 1117
lightWrapBox
Ui::RebindWidget, 964
Ui_RebindWidget, 1194
lighting, 1511
line
Ui::AnimSettingDialog, 97
Ui::ExtraSettingsWidget, 172
Ui::KbBindWidget, 402
Ui::KbLightWidget, 463
Ui::KPerfWidget, 702
Ui::MPerfWidget, 777
Ui::SettingsWidget, 1034
Ui_AnimSettingDialog, 1054
Ui_ExtraSettingsWidget, 1066
Ui_KbBindWidget, 1096
Ui_KbLightWidget, 1104
Ui_KPerfWidget, 1135
Ui_MPerfWidget, 1167
Ui_SettingsWidget, 1211
line_2
Ui::AnimSettingDialog, 97
Ui::ExtraSettingsWidget, 172
Ui::KbBindWidget, 402
Ui::KbLightWidget, 463
Ui::KPerfWidget, 702
Ui::MPerfWidget, 777
Ui::SettingsWidget, 1034
Ui_AnimSettingDialog, 1054
Ui_ExtraSettingsWidget, 1066
Ui_KbBindWidget, 1096
Ui_KbLightWidget, 1105
Ui_KPerfWidget, 1135
Ui_MPerfWidget, 1167
Ui_SettingsWidget, 1211
line_3
Ui::ExtraSettingsWidget, 172
Ui::KPerfWidget, 702
Ui::MPerfWidget, 777
Ui::SettingsWidget, 1034
Ui_ExtraSettingsWidget, 1066
Ui_KPerfWidget, 1135
Ui_MPerfWidget, 1167
Ui_SettingsWidget, 1211
linkedlist_data_s, 1902
linkedlist_datablock_internal_s, 1901
linkedlist_data
zip.c, 1907
linkedlist_datablock_internal
zip.c, 1907
list
AnimScript, 70
load
AnimScript, 71
Kb, 286
KbBind, 381
KbLight, 442
KbPerf, 524
LoadCentralDirectoryRecord
zip.c, 1911

loadGlobalRemap
KbBind, 383
loaddpi
dpi.c, 1281
dpi.h, 1287
loadprofile
devcmd.__unnamed__, 162
profile.c, 1478
profile.h, 1497
loadprofile_none
device_vtable.c, 1257
loadrgb_kb
led.h, 1350
led_keyboard.c, 1382
loadrgb_mouse
led.h, 1352
led_mouse.c, 1393
local
unzip.c, 1833
zip.c, 1905
locale
KeyMap, 654
localecase
ckb-daemon/main.c, 1396
lockAction
KeyAction, 625
lockBox
Ui::KPerfWidget, 702
Ui::RebindWidget, 964
Ui_KPerfWidget, 1136
Ui_RebindWidget, 1194
lockButton
Ui::RebindWidget, 964
Ui_RebindWidget, 1194
lockColorOff
Ui::KPerfWidget, 702
Ui_KPerfWidget, 1136
lockColorOn
Ui::KPerfWidget, 702
Ui_KPerfWidget, 1136
lockMutex
ckbsettings.cpp, 1674
ckbsettingswriter.cpp, 1677
lockMutexCache
ckbsettings.cpp, 1674
ckbsettingswriter.cpp, 1677
lockMutexStatic
ckbsettings.cpp, 1674
ckbsettingswriter.cpp, 1677
lockMutexStatic2
ckbsettings.cpp, 1674
ckbsettingswriter.cpp, 1677
loginItemBox
Ui::SettingsWidget, 1034
Ui_SettingsWidget, 1211

M65
KeyMap, 639
M65_HEIGHT
keymap.cpp, 1723
M65_WIDTH
keymap.cpp, 1723
M65Keys
keymap.cpp, 1736
m65Overlay
keywidget.cpp, 1750
MACRO
command.h, 1228
KbPerf, 508
MAX_FILE_NAME_LENGTH
QuaZip, 836
MEDHIGH
KbPerf, 507
MEDIUM
KbPerf, 507
MODE
command.h, 1227
KbPerf, 508
MPerfWidget::DpiUi, 784, 1775
MUTE
KbPerf, 508
MUTED
media.h, 1766
MX
KeyMap, 638
M_PI
ckb-anim.h, 1660
MACRO_MAX
structures.h, 1521
MAKEULONG64
zip.c, 1905
MAX_BUFFER
devnode.c, 1259
MAX_DIM
KbLight, 455
MAXFACTORS
_kiss_fft_guts.h, 1614
mAccelBox
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1066
MD_NAME_LEN
structures.h, 1521
MODE_COUNT
structures.h, 1521
MODE_NEXT
KeyAction, 634
MODE_NEXT_WRAP
KeyAction, 634
MODE_PREV
KeyAction, 634
MODE_PREV_WRAP
KeyAction, 634
MOUSE_BUTTON_FIRST
daemon/keymap.h, 1743
MOUSE_EXTRA_FIRST
daemon/keymap.h, 1743
mPerfTab

Ui::KbWidget, 601
Ui_KbWidget, 1118
MPerfWidget, 781
~MPerfWidget, 787
_xyLink, 797
boxXChanged, 787
boxXMapper, 797
boxYChanged, 788
boxYMapper, 797
buttonMapper1, 797
buttonMapper2, 797
colorChanged, 789
colorClicked, 790
colorLink, 798
DPI_COUNT, 798
DPI_MAX, 798
DPI_MIN, 798
enableChanged, 790
enableMapper, 798
isSetting, 798
MPerfWidget, 785
MPerfWidget, 785
on_aSnapBox_clicked, 791
on_copyButton_clicked, 791
on_indicBox_clicked, 792
on_lHeightBox_activated, 792
on_spinBox_valueChanged, 793
on_xyBox_clicked, 793
perf, 798
profile, 798
setPerf, 794
sliderXMapper, 798
sliderXMoved, 795
sliderYMapper, 798
sliderYMoved, 796
stages, 799
ui, 799
mPerfWidget
Ui::KbWidget, 601
Ui_KbWidget, 1118
MSG_SIZE
structures.h, 1521
macReader
RebindWidget, 1005
macro
devcmd.__unnamed__, 162
macro_pt_dequeue
input.c, 1312
macro_pt_enqueue
input.c, 1313
macro_pt_first
input.c, 1313
macroAction
KeyAction, 625
macroBox
MacroReaderThread, 733
Ui::KPerfWidget, 702
Ui_KPerfWidget, 1136
macroColorOff
Ui::KPerfWidget, 702
Ui_KPerfWidget, 1136
macroColorOn
Ui::KPerfWidget, 702
Ui_KPerfWidget, 1136
macroContent
KeyAction, 626
macroDelay
Kb, 287
macroDisplay
KeyAction, 626
macroFullLine
KeyAction, 627
macroLine
KeyAction, 628
macroNumber
Kb, 309
MacroReaderThread, 733
macroPath
Kb, 310
MacroReaderThread, 733
MacroReader, 725
~MacroReader, 728
MacroReader, 728
macroReaderThread, 729
MacroReader, 728
startWorkInAThread, 728
MacroReaderThread, 729
macroBox, 733
macroNumber, 733
macroPath, 733
MacroReaderThread, 731
macroText, 733
MacroReaderThread, 731
readMacro, 732
run, 732
macroReaderThread
MacroReader, 729
macroTab
Ui::RebindWidget, 964
Ui_RebindWidget, 1194
macroText
MacroReaderThread, 733
macroTiming
KeyAction, 628
macroaction, 1507
macromask
input.c, 1314
macromutex
device.c, 1236
device.h, 1247
macromutex2
device.c, 1237
device.h, 1247
macrovar
device.c, 1237
device.h, 1247

main
ckb-anim.h, 1669
ckb-daemon/main.c, 1397
main.cpp, 1760
main.cpp
CommandLineBackground, 1758
CommandLineClose, 1758
CommandLineError, 1758
CommandLineHelpRequested, 1758
CommandLineOK, 1758
CommandLineVersionRequested, 1758
main.cpp
appShare, 1759
CommandLineParseResults, 1758
isRunning, 1759
main, 1760
parseCommandLine, 1761
pidActive, 1763
SHMEM_SIZE, 1758
SHMEM_SIZE_V015, 1758
main_ac
ckb-daemon/main.c, 1404
main_av
ckb-daemon/main.c, 1404
MainWindow, 737
∼MainWindow, 742
addDevice, 742
checkFwUpdates, 743
cleanup, 744
closeAction, 752
closeEvent, 745
iconClicked, 745
kbWidgets, 752
MainWindow, 740
mainWindow, 752
MainWindow, 740
quitApp, 746
removeDevice, 746
restoreAction, 752
settingsWidget, 752
SettingsWidget, 1023
showFwUpdateNotification, 747
showWindow, 748
stateChange, 748
timerTick, 749
toggleTrayIcon, 750
trayIcon, 752
trayIconMenu, 752
ui, 752
updateVersion, 751
mainWindow
MainWindow, 752
mainwindow.cpp
appShare, 1764
configLabel, 1764
daemonDialogText, 1764
devpath, 1764
makeIcon
GradientDialog, 221
makeStops
GradientDialogWidget, 232
makergb_512
led_keyboard.c, 1384
makergb_full
led_keyboard.c, 1385
map
KbAnim, 325, 326
KbBind, 383, 384
KbLight, 443, 444
KeyMap, 655
KeyWidget, 668, 669
map_last
kbperf.cpp, 1710
matches
Kb, 288
max
ckb-mviz/main.c, 1421
maxsize
ckb-rain/main.c, 1432
mb2Box
Ui::RebindWidget, 964
Ui_RebindWidget, 1195
mb2Button
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
mbBox
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
mbButton
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
mdAdd
QuaZip, 836
mdAppend
QuaZip, 836
mdCreate
QuaZip, 836
mdNotOpen
QuaZip, 836
mdUnzip
QuaZip, 836
media.h
MUTED, 1766
UNKNOWN, 1766
UNMUTED, 1766
media.h
ENUM_C, 1766
ENUM_END_C, 1766
EXTERN_C, 1766
getMuteState, 1767
muteState, 1766
media_linux.cpp
getMuteState, 1768
lastKnown, 1769
media_mac.m
disableAppNap, 1769

getMuteState, 1770
mediaBox
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
mediaButton
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
mediaKeys
RebindWidget, 1005
method
QuaZipFileInfo, 909
QuaZipFileInfo64, 916
min
ckb-mviz/main.c, 1421
minX
AnimScript, 85
minY
AnimScript, 85
mkdevpath
devnode.c, 1264
devnode.h, 1273
mkfwnode
devnode.c, 1265
devnode.h, 1274
mknotifynode
devnode.c, 1266
devnode.h, 1275
mmutex
device.h, 1240
mmutex2
device.h, 1240
moc_animadddialog.cpp
QT_MOC_LITERAL, 1968
qt_meta_data_AnimAddDialog, 1968
qt_meta_stringdata_AnimAddDialog, 1969
moc_animscript.cpp
QT_MOC_LITERAL, 1971
qt_meta_data_AnimScript, 1971
qt_meta_stringdata_AnimScript, 1972
moc_animsettingdialog.cpp
QT_MOC_LITERAL, 1973
qt_meta_data_AnimSettingDialog, 1974
qt_meta_stringdata_AnimSettingDialog, 1974
moc_ckbsettingswriter.cpp
QT_MOC_LITERAL, 1976
qt_meta_data_CkbSettingsWriter, 1976
qt_meta_stringdata_CkbSettingsWriter, 1977
moc_colorbutton.cpp
QT_MOC_LITERAL, 1978
qt_meta_data_ColorButton, 1979
qt_meta_stringdata_ColorButton, 1979
moc_extrasettingswidget.cpp
QT_MOC_LITERAL, 1981
qt_meta_data_ExtraSettingsWidget, 1981
qt_meta_stringdata_ExtraSettingsWidget, 1981
moc_fwupgradedialog.cpp
QT_MOC_LITERAL, 1984
qt_meta_data_FwUpgradeDialog, 1984
qt_meta_stringdata_FwUpgradeDialog, 1985
moc_gradientbutton.cpp
QT_MOC_LITERAL, 1987
qt_meta_data_GradientButton, 1987
qt_meta_stringdata_GradientButton, 1988
moc_gradientdialog.cpp
QT_MOC_LITERAL, 1989
qt_meta_data_GradientDialog, 1990
qt_meta_stringdata_GradientDialog, 1990
moc_gradientdialogwidget.cpp
QT_MOC_LITERAL, 1992
qt_meta_data_GradientDialogWidget, 1992
qt_meta_stringdata_GradientDialogWidget, 1993
moc_kb.cpp
QT_MOC_LITERAL, 1995
qt_meta_data_Kb, 1995
qt_meta_stringdata_Kb, 1995
moc_kbanim.cpp
QT_MOC_LITERAL, 1997
qt_meta_data_KbAnim, 1997
qt_meta_stringdata_KbAnim, 1998
moc_kbanimwidget.cpp
QT_MOC_LITERAL, 2000
qt_meta_data_KbAnimWidget, 2000
qt_meta_stringdata_KbAnimWidget, 2000
moc_kbbind.cpp
QT_MOC_LITERAL, 2002
qt_meta_data_KbBind, 2002
qt_meta_stringdata_KbBind, 2002
moc_kbbindwidget.cpp
QT_MOC_LITERAL, 2004
qt_meta_data_KbBindWidget, 2005
qt_meta_stringdata_KbBindWidget, 2005
moc_kbfirmware.cpp
QT_MOC_LITERAL, 2007
qt_meta_data_KbFirmware, 2007
qt_meta_stringdata_KbFirmware, 2008
moc_kblight.cpp
QT_MOC_LITERAL, 2010
qt_meta_data_KbLight, 2010
qt_meta_stringdata_KbLight, 2011
moc_kblightwidget.cpp
QT_MOC_LITERAL, 2013
qt_meta_data_KbLightWidget, 2013
qt_meta_stringdata_KbLightWidget, 2013
moc_kbmanager.cpp
QT_MOC_LITERAL, 2015
qt_meta_data_KbManager, 2015
qt_meta_stringdata_KbManager, 2015
moc_kbmode.cpp
QT_MOC_LITERAL, 2017
qt_meta_data_KbMode, 2018
qt_meta_stringdata_KbMode, 2018
moc_kbperf.cpp
QT_MOC_LITERAL, 2020
qt_meta_data_KbPerf, 2020
qt_meta_stringdata_KbPerf, 2021
moc_kbprofile.cpp

QT_MOC_LITERAL, 2022
qt_meta_data_KbProfile, 2023
qt_meta_stringdata_KbProfile, 2023
moc_kbprofiledialog.cpp
QT_MOC_LITERAL, 2025
qt_meta_data_KbProfileDialog, 2025
qt_meta_stringdata_KbProfileDialog, 2026
moc_kbwidget.cpp
QT_MOC_LITERAL, 2028
qt_meta_data_KbWidget, 2028
qt_meta_stringdata_KbWidget, 2028
moc_keyaction.cpp
QT_MOC_LITERAL, 2030
qt_meta_data_KeyAction, 2030
qt_meta_stringdata_KeyAction, 2031
moc_keywidget.cpp
QT_MOC_LITERAL, 2032
qt_meta_data_KeyWidget, 2033
qt_meta_stringdata_KeyWidget, 2033
moc_kperfwidget.cpp
QT_MOC_LITERAL, 2035
qt_meta_data_KPerfWidget, 2035
qt_meta_stringdata_KPerfWidget, 2036
moc_layoutdialog.cpp
QT_MOC_LITERAL, 2038
qt_meta_data_LayoutDialog, 2038
qt_meta_stringdata_LayoutDialog, 2039
moc_macroreader.cpp
QT_MOC_LITERAL, 2041
qt_meta_data_MacroReader, 2041
qt_meta_data_MacroReaderThread, 2042
qt_meta_stringdata_MacroReader, 2042
qt_meta_stringdata_MacroReaderThread, 2042
moc_mainwindow.cpp
QT_MOC_LITERAL, 2044
qt_meta_data_MainWindow, 2044
qt_meta_stringdata_MainWindow, 2044
moc_modeselectdialog.cpp
QT_MOC_LITERAL, 2046
qt_meta_data_ModeSelectDialog, 2046
qt_meta_stringdata_ModeSelectDialog, 2047
moc_mperfwidget.cpp
QT_MOC_LITERAL, 2048
qt_meta_data_MPerfWidget, 2049
qt_meta_stringdata_MPerfWidget, 2049
moc_quagzipfile.cpp
QT_MOC_LITERAL, 2051
qt_meta_data_QuaGzipFile, 2051
qt_meta_stringdata_QuaGzipFile, 2052
moc_quaziodevice.cpp
QT_MOC_LITERAL, 2053
qt_meta_data_QuaZIODevice, 2054
qt_meta_stringdata_QuaZIODevice, 2054
moc_quazipfile.cpp
QT_MOC_LITERAL, 2055
qt_meta_data_QuaZipFile, 2056
qt_meta_stringdata_QuaZipFile, 2056
moc_rebindwidget.cpp
QT_MOC_LITERAL, 2058
qt_meta_data_RebindWidget, 2058
qt_meta_stringdata_RebindWidget, 2058
moc_rlistwidget.cpp
QT_MOC_LITERAL, 2060
qt_meta_data_RListWidget, 2060
qt_meta_stringdata_RListWidget, 2060
moc_settingswidget.cpp
QT_MOC_LITERAL, 2062
qt_meta_data_SettingsWidget, 2063
qt_meta_stringdata_SettingsWidget, 2063
modBox
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
modButton
Ui::RebindWidget, 965
Ui_RebindWidget, 1195
modKeys
RebindWidget, 1005
settingswidget.cpp, 2074
modNames
settingswidget.cpp, 2074
Mode
KbAnim, 314
QuaZip, 836
mode
KbAnim, 326
QuaZipPrivate, 938
mode2Raw
KPerfWidget, 711
modeAction
KeyAction, 629
modeBox
Ui::KPerfWidget, 703
Ui::RebindWidget, 966
Ui_KPerfWidget, 1136
Ui_RebindWidget, 1195
modeButton
Ui::RebindWidget, 966
Ui_RebindWidget, 1196
modeChanged
Kb, 288
KbWidget, 575
modeColorOff
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1136
modeColorOn
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1136
modeCount
KbProfile, 548
modeIcon
KbWidget, 577
ModeList
KbProfile, 541
modeList
Ui::ModeSelectDialog, 756
Ui_ModeSelectDialog, 1150

modeParent
KbBind, 384
KbPerf, 526
modeRenamed
Kb, 289
ModeSelectDialog, 757
~ModeSelectDialog, 759
_modeList, 760
ModeSelectDialog, 759
ModeSelectDialog, 759
on_selAllButton_clicked, 759
on_selNoneButton_clicked, 759
selection, 760
ui, 760
modeUpdate
KbWidget, 579
modeWrapBox
Ui::RebindWidget, 966
Ui_RebindWidget, 1196
Model
KeyMap, 638
model
Kb, 288
KeyMap, 655
modelHeight
KeyMap, 656
modelWidth
KeyMap, 656
models
usb_linux.c, 1608
modes
KbProfile, 548, 549
modesList
Ui::KbWidget, 601
Ui_KbWidget, 1117
modesList_reordered
KbWidget, 578
modified
UsbId, 1218
modifiedString
UsbId, 1216, 1217
monoRgb
kblight.cpp, 1703
keywidget.cpp, 1749
monochrome
Kb, 310
KeyWidget, 669, 670
mouseAccel
Kb, 289
mouseCurrentX
KeyWidget, 683
mouseCurrentY
KeyWidget, 683
mouseDownMode
KeyWidget, 683
mouseDownX
KeyWidget, 684
mouseDownY
KeyWidget, 684
mouseExtKeys
RebindWidget, 1005
mouseKeys
RebindWidget, 1005
mouseMoveEvent
GradientDialogWidget, 233
KeyWidget, 670
mousePressEvent
GradientDialogWidget, 234
KeyWidget, 671
mouseReleaseEvent
GradientDialogWidget, 234
KeyWidget, 672
mouseTab
Ui::RebindWidget, 966
Ui_RebindWidget, 1196
move
KbProfile, 549
moveCurrent
GradientDialogWidget, 235
mperfwidget.cpp
DPI_BASE, 1774
dpiExp, 1772
dpiLog, 1773
dpiRound, 1773
SET_END, 1772
SET_START, 1772
xyLinkPath, 1774
Multiply
KbAnim, 314
muteBox
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1137
muteColorNA
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1137
muteColorOff
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1137
muteColorOn
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1137
muteNAColor
KbPerf, 538
muteState
media.h, 1766
mvar
device.h, 1240

NAME
command.h, 1228
NO
KbFirmware, 416
KeyMap, 638
NO_LAYOUT
KeyMap, 638
NO_MODEL
KeyMap, 639

NONE
command.h, 1227
KbPerf, 508
KeyWidget, 665
NORMAL
KPerfWidget, 709
KbPerf, 508
KeyAction, 609
NOTIFY
command.h, 1228
NOTIFYOFF
command.h, 1227
NOTIFYON
command.h, 1227
NUM
KbPerf, 508
N_BUTTONS_EXTENDED
daemon/keymap.h, 1743
N_BUTTONS_HW
daemon/keymap.h, 1743
N_KEY_ZONES
daemon/keymap.h, 1743
N_KEYBYTES_EXTENDED
daemon/keymap.h, 1743
N_KEYBYTES_HW
daemon/keymap.h, 1743
N_KEYBYTES_INPUT
daemon/keymap.h, 1744
N_KEYS_EXTENDED
daemon/keymap.h, 1744
N_KEYS_EXTRA
daemon/keymap.h, 1744
N_KEYS_HW
daemon/keymap.h, 1744
N_KEYS_INPUT
daemon/keymap.h, 1744
N_LAYOUTS
keymap.cpp, 1723
N_MODELS
keymap.cpp, 1724
usb_linux.c, 1585
N_MOUSE_ZONES
daemon/keymap.h, 1744
NEEDS_FW_UPDATE
structures.h, 1521
NEW_FLAG
KbProfileDialog, 565
KbWidget, 592
NK95_HWOFF
usb.h, 1548
NK95_HWON
usb.h, 1549
NK95_M1
usb.h, 1549
NK95_M2
usb.h, 1549
NK95_M3
usb.h, 1549
NONE
ckb-gradient/main.c, 1406
NS
keymap.cpp, 1724
name
AnimScript, 74
AnimSettingDialog, 110
devcmd.__unnamed__, 162
GradientDialog::Preset, 800
KbAnim, 326, 327
KbMode, 497, 498
KbProfile, 550
KbWidget, 579
Key, 605
KeyMap, 657
QuaZipFileInfo, 909
QuaZipFileInfo64, 916
QuaZipNewInfo, 930
nameBox
Ui::KbAnimWidget, 346
Ui_KbAnimWidget, 1090
nameFilters
QuaZipDir, 878
nativetohw
profile.c, 1479
profile.h, 1498
needsSave
Kb, 290
KbAnim, 327
KbBind, 384
KbLight, 444
KbMode, 498
KbPerf, 526
KbProfile, 550
networkManager
KbFirmware, 424
newDuration
AnimSettingDialog, 110
newGuid
UsbId, 1217
newId
KbAnim, 328
KbMode, 499
KbProfile, 551
newLabel
Ui::FwUpgradeDialog, 199
Ui_FwUpgradeDialog, 1074
newLayout
KbBindWidget, 406
newMode
Kb, 290, 291
newModified
UsbId, 1217
newProfile
Kb, 291, 292
newSelection
KbBindWidget, 406
KbLightWidget, 469

KeyWidget, 684
newtarget
ckb-random/main.c, 1437
nk95cmd
usb.h, 1549
noAction
KbBind, 385
KeyAction, 629
noAnimLabel
Ui::KbAnimWidget, 347
Ui_KbAnimWidget, 1090
noReorder
KbAnimWidget, 365
noSelPage
Ui::KbAnimWidget, 347
Ui_KbAnimWidget, 1090
Normal
KbAnim, 314
notify
devcmd.__unnamed__, 162
notify.c
_cmd_get, 1454
cmd_get, 1456
cmd_notify, 1457
cmd_restart, 1457
HW_STANDARD, 1453
HWMODE_OR_RETURN, 1453
nprintf, 1458
nprintind, 1458
nprintkey, 1459
restart, 1460
notify.h
cmd_get, 1462
cmd_notify, 1462
cmd_restart, 1463
nprintf, 1463
nprintind, 1464
nprintkey, 1465
notifyNumber
Kb, 310
notifyPath
Kb, 310
notifyPathMutex
kb.cpp, 1692
notifyPaths
kb.cpp, 1692
nprintf
notify.c, 1458
notify.h, 1463
nprintind
notify.c, 1458
notify.h, 1464
nprintkey
notify.c, 1459
notify.h, 1465
numBox
Ui::KPerfWidget, 703
Ui::RebindWidget, 966
Ui_KPerfWidget, 1137
Ui_RebindWidget, 1196
numButton
Ui::RebindWidget, 966
Ui_RebindWidget, 1196
numColorOff
Ui::KPerfWidget, 703
Ui_KPerfWidget, 1137
numColorOn
Ui::KPerfWidget, 704
Ui_KPerfWidget, 1137
numKeys
RebindWidget, 1005

O0
led_keyboard.c, 1381
O1
led_keyboard.c, 1381
O2
led_keyboard.c, 1381
O3
led_keyboard.c, 1382
O4
led_keyboard.c, 1382
O5
led_keyboard.c, 1382
O6
led_keyboard.c, 1382
O7
led_keyboard.c, 1382
O8
led_keyboard.c, 1382
OFF
KbPerf, 508
ON
KbPerf, 508
OF
ioapi.h, 1784
ONE_DAY
animscript.cpp, 1643
OTHER
KbPerf, 538
OUTFIFO_MAX
structures.h, 1521
on_aSnapBox_clicked
MPerfWidget, 791
on_actionButton_clicked
FwUpgradeDialog, 190
on_altBox_activated
SettingsWidget, 1016
on_animBox_activated
AnimAddDialog, 51
on_animBox_currentIndexChanged
RebindWidget, 980
on_animButton_clicked
KbLightWidget, 470
RebindWidget, 980
on_animList_currentItemChanged
KbAnimWidget, 354

on_animList_customContextMenuRequested
KbAnimWidget, 355
on_animList_itemChanged
KbAnimWidget, 356
on_animScanButton_clicked
ExtraSettingsWidget, 178
on_applyButton_clicked
RebindWidget, 981
on_autoFWBox_clicked
SettingsWidget, 1017
on_bgButton_clicked
KbLightWidget, 471
on_blendBox_activated
KbAnimWidget, 356
on_brightnessBox_activated
KbLightWidget, 471
on_brightnessBox_clicked
ExtraSettingsWidget, 179
on_btnClearMacro_clicked
RebindWidget, 981
on_btnStartMacro_clicked
RebindWidget, 982
on_btnStopMacro_clicked
RebindWidget, 982
on_cancelButton_clicked
FwUpgradeDialog, 191
RebindWidget, 983
on_capsBox_activated
SettingsWidget, 1017
on_copyButton_clicked
KbBindWidget, 408
KPerfWidget, 712
MPerfWidget, 791
on_ctrlBox_activated
SettingsWidget, 1017
on_delayBox_clicked
ExtraSettingsWidget, 179
on_delayBox_valueChanged
AnimSettingDialog, 111
on_deleteButton_clicked
KbAnimWidget, 357
on_ditherBox_clicked
ExtraSettingsWidget, 179
on_dpiBox_currentIndexChanged
RebindWidget, 984
on_dpiButton_clicked
RebindWidget, 985
on_extraButton_clicked
SettingsWidget, 1018
on_fnBox_currentIndexChanged
RebindWidget, 985
on_fnButton_clicked
RebindWidget, 985
on_fpsBox_valueChanged
ExtraSettingsWidget, 180
on_fwUpdButton_clicked
KbWidget, 579
on_hwSaveButton_clicked
KbWidget, 581
on_indicBox_clicked
MPerfWidget, 792
on_intensityBox_valueChanged
KPerfWidget, 713
on_keyButton_clicked
KbAnimWidget, 357
on_kpDelayBox_valueChanged
AnimSettingDialog, 112
on_kpReleaseBox_stateChanged
AnimSettingDialog, 112
on_kpRepeatBox_valueChanged
AnimSettingDialog, 113
on_lHeightBox_activated
MPerfWidget, 792
on_layoutBox_activated
SettingsWidget, 1018
on_lightBox_currentIndexChanged
RebindWidget, 985
on_lightButton_clicked
RebindWidget, 986
on_lockBox_currentIndexChanged
RebindWidget, 986
on_lockButton_clicked
RebindWidget, 986
on_loginItemBox_clicked
SettingsWidget, 1019
on_mAccelBox_clicked
ExtraSettingsWidget, 180
on_mb2Box_currentIndexChanged
RebindWidget, 987
on_mb2Button_clicked
RebindWidget, 987
on_mbBox_currentIndexChanged
RebindWidget, 987
on_mbButton_clicked
RebindWidget, 988
on_mediaBox_currentIndexChanged
RebindWidget, 988
on_mediaButton_clicked
RebindWidget, 988
on_modBox_currentIndexChanged
RebindWidget, 988
on_modButton_clicked
RebindWidget, 989
on_modeBox_currentIndexChanged
RebindWidget, 989
on_modeButton_clicked
RebindWidget, 989
on_modesList_currentItemChanged
KbWidget, 582
on_modesList_customContextMenuRequested
KbWidget, 583
on_modesList_itemChanged
KbWidget, 584
on_modesList_itemClicked
KbWidget, 585
on_nameBox_textEdited

KbAnimWidget, 358
on_numBox_currentIndexChanged
RebindWidget, 990
on_numButton_clicked
RebindWidget, 990
on_opacityBox_valueChanged
KbAnimWidget, 359
on_presetBox_activated
AnimAddDialog, 52
on_presetDelete_clicked
GradientDialog, 222
on_presetList_currentItemChanged
GradientDialog, 222
on_presetName_textEdited
GradientDialog, 223
on_presetSave_clicked
GradientDialog, 223
on_previewBox_clicked
AnimAddDialog, 53
on_profileBox_activated
KbWidget, 586
on_profileList_currentItemChanged
KbProfileDialog, 559
on_profileList_customContextMenuRequested
KbProfileDialog, 559
on_profileList_itemChanged
KbProfileDialog, 561
on_profileList_itemClicked
KbProfileDialog, 562
on_programKpBox_textChanged
RebindWidget, 990
on_programKpButton_clicked
RebindWidget, 991
on_programKpSIBox_clicked
RebindWidget, 991
on_programKrBox_textChanged
RebindWidget, 991
on_programKrButton_clicked
RebindWidget, 992
on_programKrSIBox_clicked
RebindWidget, 992
on_propertyButton_clicked
KbAnimWidget, 359
on_pteMacroBox_textChanged
RebindWidget, 992
on_pushButton_clicked
SettingsWidget, 1019
on_rb_delay_asTyped_toggled
RebindWidget, 993
on_rb_delay_default_toggled
RebindWidget, 993
on_rb_delay_no_toggled
RebindWidget, 994
on_repeatBox_valueChanged
AnimSettingDialog, 113
on_resetButton_clicked
KbBindWidget, 408
RebindWidget, 994
on_sAccelBox_clicked
ExtraSettingsWidget, 181
on_sSpeedBox_valueChanged
ExtraSettingsWidget, 181
on_selAllButton_clicked
ModeSelectDialog, 759
on_selNoneButton_clicked
ModeSelectDialog, 759
on_shiftBox_activated
SettingsWidget, 1019
on_showAnimBox_clicked
KbLightWidget, 472
on_spinBox_valueChanged
MPerfWidget, 793
on_stopOpacity_valueChanged
GradientDialog, 224
on_stopPos_valueChanged
GradientDialog, 224
on_tabWidget_currentChanged
KbWidget, 586
on_trayBox_clicked
ExtraSettingsWidget, 182
on_typingBox_currentIndexChanged
RebindWidget, 995
on_typingButton_clicked
RebindWidget, 995
on_unbindButton_clicked
RebindWidget, 995
on_wheelBox_currentIndexChanged
RebindWidget, 995
on_wheelButton_clicked
RebindWidget, 996
on_winBox_activated
SettingsWidget, 1020
on_xyBox_clicked
MPerfWidget, 793
once
AutoRun, 123
opacity
KbAnim, 328
opacityBox
Ui::KbAnimWidget, 347
Ui_KbAnimWidget, 1090
open
KbLight, 445
QuaGzipFile, 823, 824
QuaZIODevice, 830
QuaZip, 855
QuaZipFile, 896–898
operator bool
Key, 603
operator const QHash< QString, Key > &
KeyMap, 657
operator QString
KeyAction, 630
operator=
ColorMap, 158
KbPerf, 527

KeyAction, 630
QuaGzipFile, 824
QuaZip, 858
QuaZipDir, 879
QuaZipFile, 900
operator==
ColorMap, 159
QuaZipDir, 879
orderChanged
RListWidget, 1010
ordered8to3
led_keyboard.c, 1385
os.h
_DEFAULT_SOURCE, 1466
_GNU_SOURCE, 1466
euid_guard_start, 1467
euid_guard_stop, 1467
UINPUT_VERSION, 1467
os_closeusb
usb.h, 1563
usb_linux.c, 1587
os_inputclose
input.h, 1324
input_linux.c, 1330
os_inputmain
usb.h, 1564
usb_linux.c, 1587
os_inputopen
input.h, 1324
input_linux.c, 1331
os_keypress
input.h, 1325
input_linux.c, 1332
os_mousemove
input.h, 1326
input_linux.c, 1332
os_resetusb
usb.h, 1567
usb_linux.c, 1591
os_sendindicators
usb.h, 1568
usb_linux.c, 1592
os_setupindicators
input.h, 1327
input_linux.c, 1333
os_setupusb
usb.h, 1569
usb_linux.c, 1593
os_usbrecv
usb.h, 1572
usb_linux.c, 1595
os_usbsend
usb.h, 1574
usb_linux.c, 1596
osxLabel
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1066
osxLine
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1066
outbuf
ckb-mviz/main.c, 1422

p
QuaZip, 865
QuaZipFile, 906
PL
KeyMap, 638
POLLRATE
command.h, 1227
PROFILEID
command.h, 1228
PROFILENAME
command.h, 1228
PT_BR
KeyMap, 638
P_GLAIVE
usb.h, 1549
P_GLAIVE_STR
usb.h, 1549
P_HARPOON
usb.h, 1549
P_HARPOON_STR
usb.h, 1549
P_K63_NRGB
usb.h, 1550
P_K63_NRGB_STR
usb.h, 1550
P_K65
usb.h, 1550
P_K65_LUX
usb.h, 1550
P_K65_LUX_STR
usb.h, 1550
P_K65_NRGB
usb.h, 1550
P_K65_NRGB_STR
usb.h, 1550
P_K65_RFIRE
usb.h, 1550
P_K65_RFIRE_STR
usb.h, 1550
P_K65_STR
usb.h, 1550
P_K68
usb.h, 1550
P_K68_STR
usb.h, 1551
P_K70
usb.h, 1551
P_K70_LUX
usb.h, 1551
P_K70_LUX_NRGB
usb.h, 1551
P_K70_LUX_NRGB_STR
usb.h, 1551
P_K70_LUX_STR

usb.h, 1551
P_K70_NRGB
usb.h, 1551
P_K70_NRGB_STR
usb.h, 1551
P_K70_RFIRE
usb.h, 1551
P_K70_RFIRE_NRGB
usb.h, 1551
P_K70_RFIRE_NRGB_STR
usb.h, 1552
P_K70_RFIRE_STR
usb.h, 1552
P_K70_STR
usb.h, 1552
P_K95
usb.h, 1552
P_K95_NRGB
usb.h, 1552
P_K95_NRGB_STR
usb.h, 1552
P_K95_PLATINUM
usb.h, 1552
P_K95_PLATINUM_STR
usb.h, 1552
P_K95_STR
usb.h, 1552
P_M65
usb.h, 1552
P_M65_PRO
usb.h, 1552
P_M65_PRO_STR
usb.h, 1553
P_M65_STR
usb.h, 1553
P_SABRE_L
usb.h, 1553
P_SABRE_L_STR
usb.h, 1553
P_SABRE_N
usb.h, 1553
P_SABRE_N_STR
usb.h, 1553
P_SABRE_O
usb.h, 1553
P_SABRE_O2
usb.h, 1553
P_SABRE_O2_STR
usb.h, 1553
P_SABRE_O_STR
usb.h, 1553
P_SCIMITAR
usb.h, 1553
P_SCIMITAR_PRO
usb.h, 1554
P_SCIMITAR_PRO_STR
usb.h, 1554
P_SCIMITAR_STR
usb.h, 1554
P_STRAFE
usb.h, 1554
P_STRAFE_NRGB
usb.h, 1554
P_STRAFE_NRGB_2
usb.h, 1554
P_STRAFE_NRGB_2_STR
usb.h, 1554
P_STRAFE_NRGB_STR
usb.h, 1554
P_STRAFE_STR
usb.h, 1554
PATCH_COUNT
keymap.cpp, 1724
PR_NAME_LEN
structures.h, 1522
PROGRAM_PR_INDEF
KeyAction, 634
PROGRAM_PR_KPSTOP
KeyAction, 635
PROGRAM_PR_KRSTOP
KeyAction, 635
PROGRAM_PR_MULTI
KeyAction, 635
PROGRAM_RE_INDEF
KeyAction, 635
PROGRAM_RE_KPSTOP
KeyAction, 635
PROGRAM_RE_MULTI
KeyAction, 635
paintEvent
GradientDialogWidget, 236
KeyWidget, 673
param
AnimScript, 74
paramIterator
AnimScript, 76
parameter, 1319
KbAnim, 328, 329
parameter_t
input.h, 1320
parameters
AnimScript, 75
parseCommandLine
main.cpp, 1761
parseVersionString
KbManager, 485
pas
ckb-mviz/main.c, 1422
patch
keymap.cpp, 1724
patchABNT2
keymap.cpp, 1729
patchANSI
keymap.cpp, 1730
patchDE
keymap.cpp, 1736

patchDK
keymap.cpp, 1736
patchDvorak
keymap.cpp, 1736
patchES
keymap.cpp, 1736
patchEU
keymap.cpp, 1737
patchFR
keymap.cpp, 1737
patchISO
keymap.cpp, 1730
patchIT
keymap.cpp, 1737
patchJP
keymap.cpp, 1737
patchJP106
keymap.cpp, 1731
patchJP106fn
keymap.cpp, 1732
patchMX
keymap.cpp, 1737
patchNO
keymap.cpp, 1738
patchPTBR
keymap.cpp, 1738
patchSE
keymap.cpp, 1738
patchnonJP106
keymap.cpp, 1732
path
AnimScript, 76
QuaZipDir, 879
pcpx
_kiss_fft_guts.h, 1614
perf
KbBind, 386
KbMode, 500
KPerfWidget, 717
MPerfWidget, 798
period
ckb-rain/main.c, 1433
permissionsFromExternalAttr
quazipfileinfo.cpp, 1821
phase
ckb-random/main.c, 1438
pickColor
ColorButton, 148
pickGradient
GradientButton, 203
pidActive
main.cpp, 1763
play_macro
input.c, 1314
pollLabel
Ui::KbWidget, 601
Ui_KbWidget, 1118
pollLabel2
Ui::KbWidget, 601
Ui_KbWidget, 1118
pollUpdates
ExtraSettingsWidget, 182
SettingsWidget, 1020
pollrate
devcmd.__unnamed__, 162
Kb, 310
popDpi
KbPerf, 527
pos
QIODevice_descriptor, 803
QuaZipFile, 900
positions
KeyMap, 658
power
ckb-mviz/main.c, 1422
powers
ckb-mviz/main.c, 1422
preProgram
KeyAction, 634
prefsPath
gradientdialog.cpp, 1689
Kb, 310
Preset
GradientDialog::Preset, 800
preset
AnimScript, 76
presetBox
Ui::AnimAddDialog, 47
Ui_AnimAddDialog, 1043
presetDelete
Ui::GradientDialog, 213
Ui_GradientDialog, 1080
presetList
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
presetName
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
presetSave
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
PresetValue
AnimScript, 59
presets
AnimScript, 77
GradientDialog, 226
pressestofull
ckb-heat/main.c, 1416
prevMode
Kb, 310
prevProfile
Kb, 310
previewAnim
KbLight, 446
previewBox
Ui::AnimAddDialog, 47

Ui_AnimAddDialog, 1043
previousItems
RListWidget, 1012
printParams
AnimScript, 77
printRGB
KbLight, 447
printdpi
dpi.c, 1282
dpi.h, 1288
printname
profile.c, 1479
printrgb
led.c, 1345
led.h, 1352
printurl
ckb-anim.h, 1671
process
AnimScript, 85
processDownload
KbFirmware, 421
product_str
usb.c, 1536
usb.h, 1576
profile
KbBindWidget, 413
KPerfWidget, 717
MPerfWidget, 798
RebindWidget, 1005
profile.c
_freeprofile, 1468
allocprofile, 1469
cmd_erase, 1469
cmd_eraseprofile, 1470
cmd_id, 1470
cmd_name, 1471
cmd_profileid, 1471
cmd_profilename, 1472
freemode, 1472
freeprofile, 1473
gethwmodename, 1474
gethwprofilename, 1474
getid, 1475
getmodename, 1476
getprofilename, 1476
hwtonative, 1477
initmode, 1478
loadprofile, 1478
nativetohw, 1479
printname, 1479
setid, 1480
u16dec, 1481
u16enc, 1481
urldecode2, 1482
urlencode2, 1483
utf16to8, 1483
utf8to16, 1484
profile.h
allocprofile, 1485
cmd_erase, 1486
cmd_eraseprofile, 1486
cmd_hwload_kb, 1487
cmd_hwload_mouse, 1488
cmd_hwsave_kb, 1489
cmd_hwsave_mouse, 1490
cmd_id, 1491
cmd_name, 1491
cmd_profileid, 1492
cmd_profilename, 1492
freeprofile, 1493
gethwmodename, 1493
gethwprofilename, 1494
getid, 1495
getmodename, 1495
getprofilename, 1496
hwloadprofile, 1485
hwtonative, 1497
loadprofile, 1497
nativetohw, 1498
setid, 1498
profile_keyboard.c
cmd_hwload_kb, 1499
cmd_hwsave_kb, 1500
hwloadmode, 1501
profile_mouse.c
cmd_hwload_mouse, 1502
cmd_hwsave_mouse, 1503
profileAdded
Kb, 293
profileBox
Ui::KbWidget, 601
Ui_KbWidget, 1118
profileChanged
Kb, 293
KbWidget, 587
profileList
Ui::KbProfileDialog, 569
Ui_KbProfileDialog, 1109
profileList_reordered
KbProfileDialog, 562
profileRenamed
Kb, 293
profileid
devcmd.__unnamed__, 162
profilename
devcmd.__unnamed__, 162
profiles
Kb, 294
progTipLabel
Ui::RebindWidget, 968
Ui_RebindWidget, 1197
programAction
KeyAction, 630
programInfo
KeyAction, 630
programKpBox

Ui::RebindWidget, 966
Ui_RebindWidget, 1196
programKpButton
Ui::RebindWidget, 966
Ui_RebindWidget, 1196
programKpExtra
Ui::RebindWidget, 967
Ui_RebindWidget, 1196
programKpModeBox
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programKpSIBox
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programKrBox
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programKrButton
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programKrExtra
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programKrModeBox
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programKrSIBox
Ui::RebindWidget, 967
Ui_RebindWidget, 1197
programTab
Ui::RebindWidget, 968
Ui_RebindWidget, 1197
progressBar
Ui::FwUpgradeDialog, 200
Ui_FwUpgradeDialog, 1074
propertyButton
Ui::KbAnimWidget, 347
Ui_KbAnimWidget, 1090
pt_head
input.c, 1317
pt_tail
input.c, 1317
pteMacroBox
Ui::RebindWidget, 968
Ui_RebindWidget, 1198
pteMacroComment
Ui::RebindWidget, 968
Ui_RebindWidget, 1198
pteMacroText
Ui::RebindWidget, 968
Ui_RebindWidget, 1198
ptlist, 1319
ptlist_t
input.h, 1320
pushButton
Ui::SettingsWidget, 1034
Ui_SettingsWidget, 1211
pushDpi
KbPerf, 528, 529
pushSniper
KbPerf, 530
pushedDpis
KbPerf, 538
pwd
CkbSettings, 133, 134

q
QuaZipFilePrivate, 921
QuaZipPrivate, 939
qCleanupResources_binary
qrc_binary.cpp, 2064
qCleanupResources_image
qrc_image.cpp, 2067
qCleanupResources_text
qrc_text.cpp, 2069
QColorMap
colormap.h, 1682
QColorMapIterator
colormap.h, 1682
QDialog, 801
QIODevice, 802
QIODevice_descriptor, 803
pos, 803
QIODevice_descriptor, 803
QIODevice_descriptor, 803
qInitResources_binary
qrc_binary.cpp, 2064
qInitResources_image
qrc_image.cpp, 2067
qInitResources_text
qrc_text.cpp, 2070
QListWidget, 804
QMainWindow, 805
QMutableColorMapIterator
colormap.h, 1682
QObject, 806
QPushButton, 807
qRegisterResourceData
qrc_binary.cpp, 2065
qrc_image.cpp, 2068
qrc_text.cpp, 2070
QT_MOC_LITERAL
moc_animadddialog.cpp, 1968
moc_animscript.cpp, 1971
moc_animsettingdialog.cpp, 1973
moc_ckbsettingswriter.cpp, 1976
moc_colorbutton.cpp, 1978
moc_extrasettingswidget.cpp, 1981
moc_fwupgradedialog.cpp, 1984
moc_gradientbutton.cpp, 1987
moc_gradientdialog.cpp, 1989
moc_gradientdialogwidget.cpp, 1992
moc_kb.cpp, 1995
moc_kbanim.cpp, 1997
moc_kbanimwidget.cpp, 2000
moc_kbbind.cpp, 2002
moc_kbbindwidget.cpp, 2004

moc_kbfirmware.cpp, 2007
moc_kblight.cpp, 2010
moc_kblightwidget.cpp, 2013
moc_kbmanager.cpp, 2015
moc_kbmode.cpp, 2017
moc_kbperf.cpp, 2020
moc_kbprofile.cpp, 2022
moc_kbprofiledialog.cpp, 2025
moc_kbwidget.cpp, 2028
moc_keyaction.cpp, 2030
moc_keywidget.cpp, 2032
moc_kperfwidget.cpp, 2035
moc_layoutdialog.cpp, 2038
moc_macroreader.cpp, 2041
moc_mainwindow.cpp, 2044
moc_modeselectdialog.cpp, 2046
moc_mperfwidget.cpp, 2048
moc_quagzipfile.cpp, 2051
moc_quaziodevice.cpp, 2053
moc_quazipfile.cpp, 2055
moc_rebindwidget.cpp, 2058
moc_rlistwidget.cpp, 2060
moc_settingswidget.cpp, 2062
QThread, 808
QUAZIO_INBUFSIZE
quaziodevice.cpp, 1809
QUAZIO_OUTBUFSIZE
quaziodevice.cpp, 1809
QUAZIP_EXPORT
quazip_global.h, 1814
qUnregisterResourceData
qrc_binary.cpp, 2065
qrc_image.cpp, 2068
qrc_text.cpp, 2070
QWidget, 939
qioapi.cpp
call_zopen64, 1791
call_zseek64, 1791
call_ztell64, 1791
fill_qiodevice64_filefunc, 1792
fill_qiodevice_filefunc, 1793
fill_zlib_filefunc64_32_def_from_filefunc32, 1794
qiodevice64_seek_file_func, 1795
qiodevice64_tell_file_func, 1796
qiodevice_close_file_func, 1796
qiodevice_error_file_func, 1797
qiodevice_fakeclose_file_func, 1797
qiodevice_open_file_func, 1798
qiodevice_read_file_func, 1799
qiodevice_seek_file_func, 1799
qiodevice_tell_file_func, 1800
qiodevice_write_file_func, 1800
SEEK_CUR, 1791
SEEK_END, 1791
SEEK_SET, 1791
qiodevice64_seek_file_func
qioapi.cpp, 1795
qiodevice64_tell_file_func
qioapi.cpp, 1796
qiodevice_close_file_func
qioapi.cpp, 1796
qiodevice_error_file_func
qioapi.cpp, 1797
qiodevice_fakeclose_file_func
qioapi.cpp, 1797
qiodevice_open_file_func
qioapi.cpp, 1798
qiodevice_read_file_func
qioapi.cpp, 1799
qiodevice_seek_file_func
qioapi.cpp, 1799
qiodevice_tell_file_func
qioapi.cpp, 1800
qiodevice_write_file_func
qioapi.cpp, 1800
qrc_binary.cpp
qCleanupResources_binary, 2064
qInitResources_binary, 2064
qRegisterResourceData, 2065
qUnregisterResourceData, 2065
qt_resource_data, 2066
qt_resource_name, 2066
qt_resource_struct, 2066
qrc_image.cpp
qCleanupResources_image, 2067
qInitResources_image, 2067
qRegisterResourceData, 2068
qUnregisterResourceData, 2068
qt_resource_data, 2068
qt_resource_name, 2068
qt_resource_struct, 2069
qrc_text.cpp
qCleanupResources_text, 2069
qInitResources_text, 2070
qRegisterResourceData, 2070
qUnregisterResourceData, 2070
qt_resource_data, 2071
qt_resource_name, 2071
qt_resource_struct, 2071
qs_strcmp
colormap.cpp, 1680
qt_meta_stringdata_AnimAddDialog_t, 1967
qt_meta_stringdata_AnimScript_t, 1970
qt_meta_stringdata_AnimSettingDialog_t, 1973
qt_meta_stringdata_CkbSettingsWriter_t, 1975
qt_meta_stringdata_ColorButton_t, 1978
qt_meta_stringdata_ExtraSettingsWidget_t, 1980
qt_meta_stringdata_FwUpgradeDialog_t, 1983
qt_meta_stringdata_GradientButton_t, 1986
qt_meta_stringdata_GradientDialog_t, 1989
qt_meta_stringdata_GradientDialogWidget_t, 1991
qt_meta_stringdata_KPerfWidget_t, 2034
qt_meta_stringdata_Kb_t, 1994
qt_meta_stringdata_KbAnim_t, 1996
qt_meta_stringdata_KbAnimWidget_t, 1999
qt_meta_stringdata_KbBind_t, 2001

qt_meta_stringdata_KbBindWidget_t, 2004
qt_meta_stringdata_KbFirmware_t, 2006
qt_meta_stringdata_KbLight_t, 2009
qt_meta_stringdata_KbLightWidget_t, 2012
qt_meta_stringdata_KbManager_t, 2014
qt_meta_stringdata_KbMode_t, 2017
qt_meta_stringdata_KbPerf_t, 2019
qt_meta_stringdata_KbProfile_t, 2022
qt_meta_stringdata_KbProfileDialog_t, 2024
qt_meta_stringdata_KbWidget_t, 2027
qt_meta_stringdata_KeyAction_t, 2029
qt_meta_stringdata_KeyWidget_t, 2032
qt_meta_stringdata_LayoutDialog_t, 2037
qt_meta_stringdata_MPerfWidget_t, 2048
qt_meta_stringdata_MacroReader_t, 2040
qt_meta_stringdata_MacroReaderThread_t, 2040
qt_meta_stringdata_MainWindow_t, 2044
qt_meta_stringdata_ModeSelectDialog_t, 2045
qt_meta_stringdata_QuaGzipFile_t, 2050
qt_meta_stringdata_QuaZIODevice_t, 2053
qt_meta_stringdata_QuaZipFile_t, 2055
qt_meta_stringdata_RListWidget_t, 2059
qt_meta_stringdata_RebindWidget_t, 2057
qt_meta_stringdata_SettingsWidget_t, 2062
qt_meta_data_AnimAddDialog
moc_animadddialog.cpp, 1968
qt_meta_data_AnimScript
moc_animscript.cpp, 1971
qt_meta_data_AnimSettingDialog
moc_animsettingdialog.cpp, 1974
qt_meta_data_CkbSettingsWriter
moc_ckbsettingswriter.cpp, 1976
qt_meta_data_ColorButton
moc_colorbutton.cpp, 1979
qt_meta_data_ExtraSettingsWidget
moc_extrasettingswidget.cpp, 1981
qt_meta_data_FwUpgradeDialog
moc_fwupgradedialog.cpp, 1984
qt_meta_data_GradientButton
moc_gradientbutton.cpp, 1987
qt_meta_data_GradientDialog
moc_gradientdialog.cpp, 1990
qt_meta_data_GradientDialogWidget
moc_gradientdialogwidget.cpp, 1992
qt_meta_data_KPerfWidget
moc_kperfwidget.cpp, 2035
qt_meta_data_Kb
moc_kb.cpp, 1995
qt_meta_data_KbAnim
moc_kbanim.cpp, 1997
qt_meta_data_KbAnimWidget
moc_kbanimwidget.cpp, 2000
qt_meta_data_KbBind
moc_kbbind.cpp, 2002
qt_meta_data_KbBindWidget
moc_kbbindwidget.cpp, 2005
qt_meta_data_KbFirmware
moc_kbfirmware.cpp, 2007
qt_meta_data_KbLight
moc_kblight.cpp, 2010
qt_meta_data_KbLightWidget
moc_kblightwidget.cpp, 2013
qt_meta_data_KbManager
moc_kbmanager.cpp, 2015
qt_meta_data_KbMode
moc_kbmode.cpp, 2018
qt_meta_data_KbPerf
moc_kbperf.cpp, 2020
qt_meta_data_KbProfile
moc_kbprofile.cpp, 2023
qt_meta_data_KbProfileDialog
moc_kbprofiledialog.cpp, 2025
qt_meta_data_KbWidget
moc_kbwidget.cpp, 2028
qt_meta_data_KeyAction
moc_keyaction.cpp, 2030
qt_meta_data_KeyWidget
moc_keywidget.cpp, 2033
qt_meta_data_LayoutDialog
moc_layoutdialog.cpp, 2038
qt_meta_data_MPerfWidget
moc_mperfwidget.cpp, 2049
qt_meta_data_MacroReader
moc_macroreader.cpp, 2041
qt_meta_data_MacroReaderThread
moc_macroreader.cpp, 2042
qt_meta_data_MainWindow
moc_mainwindow.cpp, 2044
qt_meta_data_ModeSelectDialog
moc_modeselectdialog.cpp, 2046
qt_meta_data_QuaGzipFile
moc_quagzipfile.cpp, 2051
qt_meta_data_QuaZIODevice
moc_quaziodevice.cpp, 2054
qt_meta_data_QuaZipFile
moc_quazipfile.cpp, 2056
qt_meta_data_RListWidget
moc_rlistwidget.cpp, 2060
qt_meta_data_RebindWidget
moc_rebindwidget.cpp, 2058
qt_meta_data_SettingsWidget
moc_settingswidget.cpp, 2063
qt_meta_stringdata_AnimAddDialog
moc_animadddialog.cpp, 1969
qt_meta_stringdata_AnimScript
moc_animscript.cpp, 1972
qt_meta_stringdata_AnimSettingDialog
moc_animsettingdialog.cpp, 1974
qt_meta_stringdata_CkbSettingsWriter
moc_ckbsettingswriter.cpp, 1977
qt_meta_stringdata_ColorButton
moc_colorbutton.cpp, 1979
qt_meta_stringdata_ExtraSettingsWidget
moc_extrasettingswidget.cpp, 1981
qt_meta_stringdata_FwUpgradeDialog
moc_fwupgradedialog.cpp, 1985

qt_meta_stringdata_GradientButton
moc_gradientbutton.cpp, 1988
qt_meta_stringdata_GradientDialog
moc_gradientdialog.cpp, 1990
qt_meta_stringdata_GradientDialogWidget
moc_gradientdialogwidget.cpp, 1993
qt_meta_stringdata_KPerfWidget
moc_kperfwidget.cpp, 2036
qt_meta_stringdata_Kb
moc_kb.cpp, 1995
qt_meta_stringdata_KbAnim
moc_kbanim.cpp, 1998
qt_meta_stringdata_KbAnimWidget
moc_kbanimwidget.cpp, 2000
qt_meta_stringdata_KbBind
moc_kbbind.cpp, 2002
qt_meta_stringdata_KbBindWidget
moc_kbbindwidget.cpp, 2005
qt_meta_stringdata_KbFirmware
moc_kbfirmware.cpp, 2008
qt_meta_stringdata_KbLight
moc_kblight.cpp, 2011
qt_meta_stringdata_KbLightWidget
moc_kblightwidget.cpp, 2013
qt_meta_stringdata_KbManager
moc_kbmanager.cpp, 2015
qt_meta_stringdata_KbMode
moc_kbmode.cpp, 2018
qt_meta_stringdata_KbPerf
moc_kbperf.cpp, 2021
qt_meta_stringdata_KbProfile
moc_kbprofile.cpp, 2023
qt_meta_stringdata_KbProfileDialog
moc_kbprofiledialog.cpp, 2026
qt_meta_stringdata_KbWidget
moc_kbwidget.cpp, 2028
qt_meta_stringdata_KeyAction
moc_keyaction.cpp, 2031
qt_meta_stringdata_KeyWidget
moc_keywidget.cpp, 2033
qt_meta_stringdata_LayoutDialog
moc_layoutdialog.cpp, 2039
qt_meta_stringdata_MPerfWidget
moc_mperfwidget.cpp, 2049
qt_meta_stringdata_MacroReader
moc_macroreader.cpp, 2042
qt_meta_stringdata_MacroReaderThread
moc_macroreader.cpp, 2042
qt_meta_stringdata_MainWindow
moc_mainwindow.cpp, 2044
qt_meta_stringdata_ModeSelectDialog
moc_modeselectdialog.cpp, 2047
qt_meta_stringdata_QuaGzipFile
moc_quagzipfile.cpp, 2052
qt_meta_stringdata_QuaZIODevice
moc_quaziodevice.cpp, 2054
qt_meta_stringdata_QuaZipFile
moc_quazipfile.cpp, 2056
qt_meta_stringdata_RListWidget
moc_rlistwidget.cpp, 2060
qt_meta_stringdata_RebindWidget
moc_rebindwidget.cpp, 2058
qt_meta_stringdata_SettingsWidget
moc_settingswidget.cpp, 2063
qt_resource_data
qrc_binary.cpp, 2066
qrc_image.cpp, 2068
qrc_text.cpp, 2071
qt_resource_name
qrc_binary.cpp, 2066
qrc_image.cpp, 2068
qrc_text.cpp, 2071
qt_resource_struct
qrc_binary.cpp, 2066
qrc_image.cpp, 2069
qrc_text.cpp, 2071
QuaZip
csDefault, 836
csInsensitive, 836
csSensitive, 836
MAX_FILE_NAME_LENGTH, 836
mdAdd, 836
mdAppend, 836
mdCreate, 836
mdNotOpen, 836
mdUnzip, 836
QuaAdler32, 809
calculate, 811
checksum, 812
QuaAdler32, 811
QuaAdler32, 811
reset, 811
update, 812
value, 812
QuaChecksum32, 812
calculate, 814
reset, 814
update, 814
value, 814
QuaCrc32, 815
calculate, 817
checksum, 818
QuaCrc32, 817
QuaCrc32, 817
reset, 817
update, 818
value, 818
QuaGzipFile, 818
~QuaGzipFile, 822
close, 822
d, 825
flush, 823
getFileName, 823
isSequential, 823
open, 823, 824
operator=, 824

QuaGzipFile, 821, 822
QuaGzipFile, 821, 822
readData, 824
setFileName, 824
writeData, 824
QuaZIODevice, 825
~QuaZIODevice, 828
close, 828
d, 832
flush, 829
getIoDevice, 830
isSequential, 830
open, 830
QuaZIODevice, 828
QuaZIODevice, 828
readData, 831
writeData, 832
QuaZip, 833
~QuaZip, 837
CaseSensitivity, 836
close, 838
Constants, 836
convertCaseSensitivity, 839
getComment, 840
getCommentCodec, 840
getCurrentFileInfo, 841, 842
getCurrentFileName, 843
getEntriesCount, 844
getFileInfoList, 845
getFileInfoList64, 845
getFileNameCodec, 846
getFileNameList, 846
getIoDevice, 847
getMode, 847
getUnzFile, 848
getZipError, 849
getZipFile, 850
getZipName, 850
goToFirstFile, 851
goToNextFile, 852
hasCurrentFile, 853
isAutoClose, 853
isDataDescriptorWritingEnabled, 853
isOpen, 854
isZip64Enabled, 854
Mode, 836
open, 855
operator=, 858
p, 865
QuaZip, 837
QuaZipPrivate, 865
QuaZip, 837
QuaZipPrivate, 937
setAutoClose, 858
setComment, 859
setCommentCodec, 859
setCurrentFile, 859
setDataDescriptorWritingEnabled, 861
setDefaultFileNameCodec, 862
setFileNameCodec, 863
setIoDevice, 863
setZip64Enabled, 864
setZipName, 864
QuaZip_getFileInfo
quazip.cpp, 1811, 1812
QuaZipDir, 865
~QuaZipDir, 868
caseSensitivity, 869
cd, 869
cdUp, 871
count, 871
d, 882
dirName, 872
entryInfoList, 872, 873
entryInfoList64, 873, 874
entryList, 874, 875
exists, 875, 876
filePath, 877
filter, 877
isRoot, 878
nameFilters, 878
operator=, 879
operator==, 879
path, 879
QuaZipDir, 868
QuaZipDir, 868
relativeFilePath, 880
setCaseSensitivity, 880
setFilter, 881
setNameFilters, 881
setPath, 881
setSorting, 882
sorting, 882
QuaZipDir_convertInfoList
quazipdir.cpp, 1815, 1816
QuaZipDir_getFileInfo
quazipdir.cpp, 1816
QuaZipFile, 883
~QuaZipFile, 888
atEnd, 888
bytesAvailable, 889
close, 889
csize, 890
getActualFileName, 891
getCaseSensitivity, 892
getFileInfo, 892, 893
getFileName, 894
getZip, 894
getZipError, 894
getZipName, 895
isRaw, 895
isSequential, 896
open, 896–898
operator=, 900
p, 906
pos, 900

QuaZipFile, 886, 887
QuaZipFilePrivate, 905
QuaZipFile, 886, 887
QuaZipFilePrivate, 920
readData, 901
setFileName, 902
setZip, 902
setZipName, 903
size, 903
usize, 904
writeData, 905
QuaZipFileInfo, 906
comment, 908
compressedSize, 908
crc, 908
dateTime, 908
diskNumberStart, 908
externalAttr, 908
extra, 908
flags, 908
getPermissions, 907
internalAttr, 909
method, 909
name, 909
uncompressedSize, 909
versionCreated, 909
versionNeeded, 909
QuaZipFileInfo64, 909
comment, 915
compressedSize, 915
crc, 915
dateTime, 915
diskNumberStart, 915
externalAttr, 915
extra, 916
flags, 916
getNTFSaTime, 911
getNTFScTime, 912
getNTFSmTime, 912
getPermissions, 913
internalAttr, 916
isEncrypted, 914
method, 916
name, 916
toQuaZipFileInfo, 914
uncompressedSize, 916
versionCreated, 916
versionNeeded, 916
QuaZipFilePrivate, 917
~QuaZipFilePrivate, 919
caseSensitivity, 920
crc, 920
fileName, 920
internal, 920
q, 921
QuaZipFile, 920
QuaZipFilePrivate, 918, 919
QuaZipFile, 905
QuaZipFilePrivate, 918, 919
raw, 921
resetZipError, 919
setZipError, 919
uncompressedSize, 921
writePos, 921
zip, 921
zipError, 921
QuaZipNewInfo, 922
comment, 929
dateTime, 929
externalAttr, 929
extraGlobal, 930
extraLocal, 930
internalAttr, 930
name, 930
QuaZipNewInfo, 923, 924
QuaZipNewInfo, 923, 924
setFileDateTime, 925
setFileNTFSTimes, 928
setFileNTFSaTime, 925
setFileNTFScTime, 926
setFileNTFSmTime, 927
setFilePermissions, 928
setPermissions, 929
uncompressedSize, 930
QuaZipNewInfo_init
quazipnewinfo.cpp, 1823
QuaZipNewInfo_setPermissions
quazipnewinfo.cpp, 1824
QuaZipPrivate, 930
addCurrentFileToDirectoryMap, 933
autoClose, 937
clearDirectoryMap, 934
comment, 937
commentCodec, 937
dataDescriptorWritingEnabled, 937
defaultFileNameCodec, 938
directoryCaseInsensitive, 938
directoryCaseSensitive, 938
fileNameCodec, 938
getDefaultFileNameCodec, 935
getFileInfoList, 935
goToFirstUnmappedFile, 936
hasCurrentFile_f, 938
ioDevice, 938
lastMappedDirectoryEntry, 938
mode, 938
q, 939
QuaZip, 937
QuaZipPrivate, 932, 933
QuaZip, 865
QuaZipPrivate, 932, 933
zip64, 939
zipError, 939
zipName, 939
quantize8to3
led_keyboard.c, 1386

quaziodevice.cpp
QUAZIO_INBUFSIZE, 1809
QUAZIO_OUTBUFSIZE, 1809
quazip.cpp
QuaZip_getFileInfo, 1811, 1812
quazip.h
UNZ_OPENERROR, 1813
quazip_global.h
QUAZIP_EXPORT, 1814
UNUSED, 1815
quazipdir.cpp
QuaZipDir_convertInfoList, 1815, 1816
QuaZipDir_getFileInfo, 1816
quazipfileinfo.cpp
getNTFSTime, 1819
permissionsFromExternalAttr, 1821
quazipnewinfo.cpp
QuaZipNewInfo_init, 1823
QuaZipNewInfo_setPermissions, 1824
setNTFSTime, 1824
quit
ckb-daemon/main.c, 1399
quitApp
MainWindow, 746
quitWithLock
ckb-daemon/main.c, 1400

REBIND
command.h, 1228
RESTART
command.h, 1228
RGB
command.h, 1228
KPerfWidget, 709
README.md, 1219
RListWidget, 1007
change, 1009
currentData, 1012
DATA_ROLE, 1012
dragged, 1012
enter, 1009
orderChanged, 1010
previousItems, 1012
RListWidget, 1009
reorderTimer, 1012
rescanItems, 1010
RListWidget, 1009
timerTick, 1011
ROADMAP.md, 1219
randomBright
ckb-heat/main.c, 1416
randomize
ckb-gradient/main.c, 1410
ckb-ripple/main.c, 1444
raw
QuaZipFilePrivate, 921
raw2Mode
KPerfWidget, 713
rb_delay_asTyped
Ui::RebindWidget, 968
Ui_RebindWidget, 1198
rb_delay_default
Ui::RebindWidget, 968
Ui_RebindWidget, 1198
rb_delay_no
Ui::RebindWidget, 968
Ui_RebindWidget, 1198
rbWidget
Ui::KbBindWidget, 402
Ui_KbBindWidget, 1096
reInit
KbAnim, 330
readAnyFrame
AnimScript, 85
readData
QuaGzipFile, 824
QuaZIODevice, 831
QuaZipFile, 901
readFrame
AnimScript, 85
readMacro
MacroReaderThread, 732
readNotify
Kb, 295
readProcess
AnimScript, 78
readcmd
command.c, 1220
command.h, 1229
readlines
devnode.c, 1266
devnode.h, 1275
readlines_ctx
devnode.h, 1273
readlines_ctx_free
devnode.c, 1267
devnode.h, 1276
readlines_ctx_init
devnode.c, 1268
devnode.h, 1277
rebind
devcmd.__unnamed__, 162
RebindWidget, 971
~RebindWidget, 976
applyChanges, 976
bind, 1004
convertMacroBox, 978
fnKeys, 1005
helpStatus, 979
macReader, 1005
mediaKeys, 1005
modKeys, 1005
mouseExtKeys, 1005
mouseKeys, 1005
numKeys, 1005
on_animBox_currentIndexChanged, 980
on_animButton_clicked, 980

on_applyButton_clicked, 981
on_btnClearMacro_clicked, 981
on_btnStartMacro_clicked, 982
on_btnStopMacro_clicked, 982
on_cancelButton_clicked, 983
on_dpiBox_currentIndexChanged, 984
on_dpiButton_clicked, 985
on_fnBox_currentIndexChanged, 985
on_fnButton_clicked, 985
on_lightBox_currentIndexChanged, 985
on_lightButton_clicked, 986
on_lockBox_currentIndexChanged, 986
on_lockButton_clicked, 986
on_mb2Box_currentIndexChanged, 987
on_mb2Button_clicked, 987
on_mbBox_currentIndexChanged, 987
on_mbButton_clicked, 988
on_mediaBox_currentIndexChanged, 988
on_mediaButton_clicked, 988
on_modBox_currentIndexChanged, 988
on_modButton_clicked, 989
on_modeBox_currentIndexChanged, 989
on_modeButton_clicked, 989
on_numBox_currentIndexChanged, 990
on_numButton_clicked, 990
on_programKpBox_textChanged, 990
on_programKpButton_clicked, 991
on_programKpSIBox_clicked, 991
on_programKrBox_textChanged, 991
on_programKrButton_clicked, 992
on_programKrSIBox_clicked, 992
on_pteMacroBox_textChanged, 992
on_rb_delay_asTyped_toggled, 993
on_rb_delay_default_toggled, 993
on_rb_delay_no_toggled, 994
on_resetButton_clicked, 994
on_typingBox_currentIndexChanged, 995
on_typingButton_clicked, 995
on_unbindButton_clicked, 995
on_wheelBox_currentIndexChanged, 995
on_wheelButton_clicked, 996
profile, 1005
RebindWidget, 975
RebindWidget, 975
selection, 1005
setBind, 996
setBox, 999
setCorrectRadioButton, 1000
setSelection, 1001
TAB_ANIM, 1006
TAB_KB, 1006
TAB_MACRO, 1006
TAB_MOUSE, 1006
TAB_PROGRAM, 1006
TAB_SPECIAL, 1006
typingKeys, 1006
ui, 1006
wheelKeys, 1007
rebindwidget.cpp
    DPI_CUST_IDX, 1965
    DPI_OFFSET, 1965
rebuildBaseMap
    KbLight, 448
refreshList
    KbAnimWidget, 360
relProgram
    KeyAction, 635
relativeFilePath
    QuaZipDir, 880
remove
    CkbSettings, 135
removeAll
    KbProfile, 552
removeCache
    CkbSettings, 139
removeDev
    FwUpgradeDialog, 191
removeDevice
    MainWindow, 746
removeFile
    JlCompress, 259
reorderAnims
    KbAnimWidget, 361
reorderTimer
    RListWidget, 1012
repeatBox
    Ui::AnimSettingDialog, 97
    Ui_AnimSettingDialog, 1054
repeatKey
    KbAnim, 338
repeatLabel
    Ui::AnimSettingDialog, 97
    Ui_AnimSettingDialog, 1054
repeatLabel_2
    Ui::AnimSettingDialog, 97
    Ui_AnimSettingDialog, 1054
repeatMsec
    AnimScript, 85
    KbAnim, 338
repeatTime
    KbAnim, 338
repopulate
    KbProfileDialog, 563
rescanItems
    RListWidget, 1010
reset
    QuaAdler32, 811
    QuaChecksum32, 814
    QuaCrc32, 817
reset_stop
    ckb-daemon/main.c, 1404
    usb.c, 1541
resetAction
    KbBind, 387
resetButton
    Ui::KbBindWidget, 402

Ui::RebindWidget, 969
Ui_KbBindWidget, 1096
Ui_RebindWidget, 1198
resetIndicators
KbLight, 448
resetParams
KbAnim, 330
resetZipError
QuaZipFilePrivate, 919
resetusb
usb.h, 1554
restart
ckb-daemon/main.c, 1401
devcmd.__unnamed__, 162
notify.c, 1460
restartAnimation
KbLight, 449
restoreAction
MainWindow, 752
retranslateUi
Ui::AnimAddDialog, 44
Ui::AnimSettingDialog, 88
Ui::ExtraSettingsWidget, 165
Ui::FwUpgradeDialog, 196
Ui::GradientDialog, 208
Ui::KbAnimWidget, 340
Ui::KbBindWidget, 398
Ui::KbLightWidget, 457
Ui::KbProfileDialog, 567
Ui::KbWidget, 594
Ui::KPerfWidget, 688
Ui::LayoutDialog, 719
Ui::MainWindow, 735
Ui::ModeSelectDialog, 754
Ui::MPerfWidget, 764
Ui::RebindWidget, 943
Ui::SettingsWidget, 1026
Ui_AnimAddDialog, 1040
Ui_AnimSettingDialog, 1046
Ui_ExtraSettingsWidget, 1059
Ui_FwUpgradeDialog, 1070
Ui_GradientDialog, 1076
Ui_KbAnimWidget, 1084
Ui_KbBindWidget, 1093
Ui_KbLightWidget, 1099
Ui_KbProfileDialog, 1107
Ui_KbWidget, 1111
Ui_KPerfWidget, 1122
Ui_LayoutDialog, 1140
Ui_MainWindow, 1145
Ui_ModeSelectDialog, 1148
Ui_MPerfWidget, 1154
Ui_RebindWidget, 1174
Ui_SettingsWidget, 1203
retrigger
AnimScript, 79
revertusb
usb.c, 1537
usb.h, 1577
rgb, 1434
devcmd.__unnamed__, 162
rgbMode
KeyWidget, 678, 679
rgbcmp
led_keyboard.c, 1386
led_mouse.c, 1393
right
settingswidget.cpp, 2074
rm_recursive
devnode.c, 1268
rmdevpath
devnode.c, 1269
devnode.h, 1277
rmnotifynode
devnode.c, 1270
devnode.h, 1278
run
CkbSettingsWriter, 141
Kb, 299
MacroReaderThread, 732
runningPushIdx
KbPerf, 538

SABRE
KeyMap, 639
SCIMITAR
KeyMap, 639
SCROLL
KbPerf, 508
SCROLLSPEED
command.h, 1227
SE
KeyMap, 638
SET
KeyWidget, 665
SNAP
command.h, 1228
SPECIAL
KeyAction, 609
STRAFE
KeyMap, 639
SUBTRACT
KeyWidget, 665
SWITCH
command.h, 1227
S_CUSTOM
devnode.h, 1273
S_CUSTOM_R
devnode.h, 1273
S_GID_READ
devnode.c, 1259
S_MUL
_kiss_fft_guts.h, 1614
S_READ
devnode.h, 1273
S_READDIR
devnode.h, 1273

S_READWRITE
devnode.h, 1273
SABRE_HEIGHT
keymap.cpp, 1724
SABRE_WIDTH
keymap.cpp, 1724
sAccelBox
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1066
sAuthorLabel
Ui::AnimSettingDialog, 97
Ui_AnimSettingDialog, 1054
SCAN_KBD
daemon/keymap.h, 1744
SCAN_MOUSE
daemon/keymap.h, 1744
SCAN_SILENT
daemon/keymap.h, 1744
SCIM_HEIGHT
keymap.cpp, 1724
SCIM_WIDTH
keymap.cpp, 1724
SCROLL_ACCELERATED
structures.h, 1522
SCROLL_MAX
structures.h, 1522
SCROLL_MIN
structures.h, 1522
sDescLabel
Ui::AnimSettingDialog, 97
Ui_AnimSettingDialog, 1054
SEEK_CUR
qioapi.cpp, 1791
zip.c, 1906
SEEK_END
qioapi.cpp, 1791
zip.c, 1906
SEEK_SET
qioapi.cpp, 1791
zip.c, 1906
SERIAL_LEN
structures.h, 1522
SET_END
mperfwidget.cpp, 1772
SET_KEYBIT
structures.h, 1522
SET_START
mperfwidget.cpp, 1772
SGroup, 1036
∼SGroup, 1038
_settings, 1038
SGroup, 1038
SGroup, 1038
SHMEM_SIZE
main.cpp, 1758
SHMEM_SIZE_V015
main.cpp, 1758
SIZECENTRALDIRITEM
unzip.c, 1833
SIZECENTRALHEADER
zip.c, 1906
SIZEDATA_INDATABLOCK
zip.c, 1906
SIZEZIPLOCALHEADER
unzip.c, 1833
sLicenseLabel
Ui::AnimSettingDialog, 98
Ui_AnimSettingDialog, 1055
SNIPER
KbPerf, 538
sNameLabel
Ui::AnimSettingDialog, 98
Ui_AnimSettingDialog, 1055
sSpeedBox
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1067
sSpeedWidget
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1067
sToL
kblight.cpp, 1704
sVerLabel
Ui::AnimSettingDialog, 98
Ui_AnimSettingDialog, 1055
sYearLabel
Ui::AnimSettingDialog, 98
Ui_AnimSettingDialog, 1055
sabOverlay
keywidget.cpp, 1750
SabreKeys
keymap.cpp, 1738
save
Kb, 300
KbAnim, 331
KbBind, 388
KbLight, 450
KbMode, 500
KbPerf, 530
KbProfile, 553
saveBlob
FwUpgradeDialog, 192
saveGlobalRemap
KbBind, 389
savePath
FwUpgradeDialog, 193
savedpi
dpi.c, 1283
dpi.h, 1289
savergb_kb
led.h, 1354
led_keyboard.c, 1387
savergb_mouse
led.h, 1355
led_mouse.c, 1394
scan
AnimScript, 80

scanKeyboards
KbManager, 485
scanTimer
KbManager, 487
ScimKeys
keymap.cpp, 1739
scimOverlay
keywidget.cpp, 1751
script
KbAnim, 332
scriptName
KbAnim, 332
scripts
AnimAddDialog, 53
AnimScript, 85
scrollBox
Ui::KPerfWidget, 704
Ui_KPerfWidget, 1137
scrollColorOff
Ui::KPerfWidget, 704
Ui_KPerfWidget, 1137
scrollColorOn
Ui::KPerfWidget, 704
Ui_KPerfWidget, 1138
scrollSpeed
Kb, 301
selAllButton
Ui::ModeSelectDialog, 756
Ui_ModeSelectDialog, 1150
selLabel
Ui::KbLightWidget, 463
Ui_KbLightWidget, 1105
selNoneButton
Ui::ModeSelectDialog, 757
Ui_ModeSelectDialog, 1151
selectAll
KeyWidget, 679
selectLabel
Ui::KbBindWidget, 402
Ui_KbBindWidget, 1097
selected
GradientDialogWidget, 242
LayoutDialog, 725
selectedKeys
KbAnimWidget, 365
selectedOffset
GradientDialogWidget, 242
selectedPos
GradientDialogWidget, 242
selectedStop
GradientDialogWidget, 237
selection
KeyWidget, 684
ModeSelectDialog, 760
RebindWidget, 1005
selectionChanged
KeyWidget, 680
selectionColors
GradientDialogWidget, 238
selectionStack
Ui::KbAnimWidget, 347
Ui_KbAnimWidget, 1090
serialLabel
Ui::KbWidget, 601
Ui_KbWidget, 1118
set
CkbSettings, 135
setAction
KbBind, 390
setAnimation
KeyWidget, 680
setAnimationToSelection
KeyWidget, 681
setAutoClose
QuaZip, 858
setBind
KbBindWidget, 409
RebindWidget, 996
setBox
RebindWidget, 999
setCaseSensitivity
QuaZipDir, 880
setComment
QuaZip, 859
setCommentCodec
QuaZip, 859
setCorrectRadioButton
RebindWidget, 1000
setCurrent
KbAnimWidget, 362
setCurrentColor
GradientDialogWidget, 238
setCurrentFile
QuaZip, 859
setCurrentMode
Kb, 302, 303
setCurrentProfile
Kb, 304
setDataDescriptorWritingEnabled
QuaZip, 861
setDefaultFileNameCodec
QuaZip, 862
setDuration
AnimScript, 80
setFileDateTime
QuaZipNewInfo, 925
setFileNTFSTimes
QuaZipNewInfo, 928
setFileNTFSaTime
QuaZipNewInfo, 925
setFileNTFScTime
QuaZipNewInfo, 926
setFileNTFSmTime
QuaZipNewInfo, 927
setFileName
QuaGzipFile, 824

QuaZipFile, 902
setFileNameCodec
QuaZip, 863
setFilePermissions
QuaZipNewInfo, 928
setFilter
QuaZipDir, 881
setGlobalRemap
KbBind, 391
setIndicator
KbLight, 451
KbPerf, 531
setIoDevice
QuaZip, 863
setLabel
ColorButton, 148, 149
setLight
KbAnimWidget, 363
KbLightWidget, 472
setMonochrome
KbLightWidget, 473
setNTFSTime
quazipnewinfo.cpp, 1824
setNameFilters
QuaZipDir, 881
setNeedsSave
KbMode, 501
KbProfile, 554
setNeedsUpdate
KbBind, 392
KbMode, 501
KbPerf, 532
KbProfile, 554
setPath
QuaZipDir, 881
setPerf
KPerfWidget, 714
MPerfWidget, 794
setPermissions
QuaZipNewInfo, 929
setPreset
GradientDialog, 224
setSelectedKeys
KbAnimWidget, 364
setSelection
KeyWidget, 681
RebindWidget, 1001
setSorting
QuaZipDir, 882
setStatus
SettingsWidget, 1020
setStops
GradientDialogWidget, 239
setValue
CkbSettings, 136
setVersion
SettingsWidget, 1021
setZip
QuaZipFile, 902
setZip64Enabled
QuaZip, 864
setZipError
QuaZipFilePrivate, 919
setZipName
QuaZip, 864
QuaZipFile, 903
setactive
device.h, 1240
setactive_kb
device.h, 1242
device_keyboard.c, 1249
setactive_mouse
device.h, 1244
device_mouse.c, 1253
setid
profile.c, 1480
profile.h, 1498
setmodeindex
devcmd.__unnamed__, 162
setmodeindex_nrgb
device.h, 1245
device_keyboard.c, 1251
settingPath
autorun.cpp, 1648
settingTab
Ui::AnimSettingDialog, 97
Ui_AnimSettingDialog, 1055
settingWidgets
AnimSettingDialog, 119
settingsCacheMutex
ckbsettings.cpp, 1675
ckbsettingswriter.cpp, 1678
settingsGrid
Ui::AnimSettingDialog, 97
Ui_AnimSettingDialog, 1055
settingsMutex
ckbsettings.cpp, 1675
ckbsettingswriter.cpp, 1678
settingsUpdated
KbPerf, 532
SettingsWidget, 1012
~SettingsWidget, 1016
extra, 1023
MainWindow, 1023
on_altBox_activated, 1016
on_autoFWBox_clicked, 1017
on_capsBox_activated, 1017
on_ctrlBox_activated, 1017
on_extraButton_clicked, 1018
on_layoutBox_activated, 1018
on_loginItemBox_clicked, 1019
on_pushButton_clicked, 1019
on_shiftBox_activated, 1019
on_winBox_activated, 1020
pollUpdates, 1020
setStatus, 1020

setVersion, 1021
SettingsWidget, 1015
SettingsWidget, 1015
showLayoutDialog, 1021
ui, 1023
updateModifiers, 1022
settingsWidget
MainWindow, 752
settingswidget.cpp
devpath, 2074
modKeys, 2074
modNames, 2074
right, 2074
setupUi
Ui::AnimAddDialog, 45
Ui::AnimSettingDialog, 89
Ui::ExtraSettingsWidget, 166
Ui::FwUpgradeDialog, 196
Ui::GradientDialog, 209
Ui::KbAnimWidget, 341
Ui::KbBindWidget, 398
Ui::KbLightWidget, 458
Ui::KbProfileDialog, 568
Ui::KbWidget, 595
Ui::KPerfWidget, 689
Ui::LayoutDialog, 719
Ui::MainWindow, 735
Ui::ModeSelectDialog, 755
Ui::MPerfWidget, 765
Ui::RebindWidget, 946
Ui::SettingsWidget, 1027
Ui_AnimAddDialog, 1040
Ui_AnimSettingDialog, 1047
Ui_ExtraSettingsWidget, 1060
Ui_FwUpgradeDialog, 1070
Ui_GradientDialog, 1077
Ui_KbAnimWidget, 1084
Ui_KbBindWidget, 1093
Ui_KbLightWidget, 1100
Ui_KbProfileDialog, 1107
Ui_KbWidget, 1112
Ui_KPerfWidget, 1123
Ui_LayoutDialog, 1140
Ui_MainWindow, 1145
Ui_ModeSelectDialog, 1149
Ui_MPerfWidget, 1155
Ui_RebindWidget, 1176
Ui_SettingsWidget, 1204
setupusb
usb.c, 1538
usb.h, 1578
shareDimming
KbLight, 451, 452
shiftBox
Ui::SettingsWidget, 1034
Ui_SettingsWidget, 1211
shiftLabel
Ui::SettingsWidget, 1035
Ui_SettingsWidget, 1211
showAnimBox
Ui::KbLightWidget, 463
Ui_KbLightWidget, 1105
showFirstTab
KbWidget, 588
showFwUpdate
KbWidget, 588
showFwUpdateNotification
MainWindow, 747
showLastTab
KbWidget, 589
showLayoutDialog
SettingsWidget, 1021
showPreview
AnimAddDialog, 53
showWindow
MainWindow, 748
sidelightToggled
KeyWidget, 682
sighandler
ckb-daemon/main.c, 1402
sighandler2
ckb-daemon/main.c, 1403
size
QuaZipFile, 903
sliderXMapper
MPerfWidget, 798
sliderXMoved
MPerfWidget, 795
sliderYMapper
MPerfWidget, 798
sliderYMoved
MPerfWidget, 796
snap
devcmd.__unnamed__, 162
sniperDpi
KbPerf, 532, 533
sniperValue
KeyAction, 635
sorting
QuaZipDir, 882
spawn
ckb-rain/main.c, 1433
specialInfo
KeyAction, 631
speed
ckb-rain/main.c, 1433
spinBox
Ui::MPerfWidget, 778
Ui_MPerfWidget, 1167
src/ckb-daemon/command.c, 1219
src/ckb-daemon/command.h, 1224
src/ckb-daemon/device.c, 1233
src/ckb-daemon/device.h, 1237
src/ckb-daemon/device_keyboard.c, 1247
src/ckb-daemon/device_mouse.c, 1252
src/ckb-daemon/device_vtable.c, 1255

src/ckb-daemon/devnode.c, 1257
src/ckb-daemon/devnode.h, 1271
src/ckb-daemon/dpi.c, 1279
src/ckb-daemon/dpi.h, 1285
src/ckb-daemon/extra_mac.c, 1291
src/ckb-daemon/firmware.c, 1291
src/ckb-daemon/firmware.h, 1296
src/ckb-daemon/includes.h, 1299
src/ckb-daemon/input.c, 1303
src/ckb-daemon/input.h, 1317
src/ckb-daemon/input_linux.c, 1328
src/ckb-daemon/input_mac.c, 1335
src/ckb-daemon/input_mac_mouse.c, 1335
src/ckb-daemon/keymap.c, 1335
src/ckb-daemon/keymap.h, 1740
src/ckb-daemon/keymap_mac.h, 1340
src/ckb-daemon/led.c, 1341
src/ckb-daemon/led.h, 1346
src/ckb-daemon/led_keyboard.c, 1358
src/ckb-daemon/led_mouse.c, 1392
src/ckb-daemon/main.c, 1396
src/ckb-daemon/notify.c, 1452
src/ckb-daemon/notify.h, 1461
src/ckb-daemon/os.h, 1466
src/ckb-daemon/profile.c, 1467
src/ckb-daemon/profile.h, 1484
src/ckb-daemon/profile_keyboard.c, 1499
src/ckb-daemon/profile_mouse.c, 1502
src/ckb-daemon/structures.h, 1504
src/ckb-daemon/usb.c, 1523
src/ckb-daemon/usb.h, 1541
src/ckb-daemon/usb_linux.c, 1583
src/ckb-daemon/usb_mac.c, 1609
src/ckb-gradient/main.c, 1404
src/ckb-heat/main.c, 1411
src/ckb-mviz/_kiss_fft_guts.h, 1609
src/ckb-mviz/kiss_fft.c, 1614
src/ckb-mviz/kiss_fft.h, 1625
src/ckb-mviz/kiss_fftr.c, 1632
src/ckb-mviz/kiss_fftr.h, 1637
src/ckb-mviz/main.c, 1416
src/ckb-pinwheel/main.c, 1422
src/ckb-rain/main.c, 1427
src/ckb-random/main.c, 1433
src/ckb-ripple/main.c, 1438
src/ckb-wave/main.c, 1444
src/ckb/animadddialog.cpp, 1641
src/ckb/animadddialog.h, 1642
src/ckb/animscript.cpp, 1642
src/ckb/animscript.h, 1644
src/ckb/animsettingdialog.cpp, 1645
src/ckb/animsettingdialog.h, 1647
src/ckb/autorun.cpp, 1648
src/ckb/autorun.h, 1649
src/ckb/ckb-anim.h, 1649
src/ckb/ckbsettings.cpp, 1673
src/ckb/ckbsettings.h, 1676
src/ckb/ckbsettingswriter.cpp, 1676
src/ckb/ckbsettingswriter.h, 1678
src/ckb/colorbutton.cpp, 1678
src/ckb/colorbutton.h, 1679
src/ckb/colormap.cpp, 1680
src/ckb/colormap.h, 1681
src/ckb/extrasettingswidget.cpp, 1682
src/ckb/extrasettingswidget.h, 1683
src/ckb/fwupgradedialog.cpp, 1683
src/ckb/fwupgradedialog.h, 1686
src/ckb/gradientbutton.cpp, 1687
src/ckb/gradientbutton.h, 1688
src/ckb/gradientdialog.cpp, 1689
src/ckb/gradientdialog.h, 1689
src/ckb/gradientdialogwidget.cpp, 1690
src/ckb/gradientdialogwidget.h, 1690
src/ckb/kb.cpp, 1691
src/ckb/kb.h, 1692
src/ckb/kbanim.cpp, 1693
src/ckb/kbanim.h, 1695
src/ckb/kbanimwidget.cpp, 1696
src/ckb/kbanimwidget.h, 1696
src/ckb/kbbind.cpp, 1697
src/ckb/kbbind.h, 1698
src/ckb/kbbindwidget.cpp, 1699
src/ckb/kbbindwidget.h, 1699
src/ckb/kbfirmware.cpp, 1700
src/ckb/kbfirmware.h, 1701
src/ckb/kblight.cpp, 1702
src/ckb/kblight.h, 1705
src/ckb/kblightwidget.cpp, 1706
src/ckb/kblightwidget.h, 1706
src/ckb/kbmanager.cpp, 1707
src/ckb/kbmanager.h, 1708
src/ckb/kbmode.cpp, 1709
src/ckb/kbmode.h, 1709
src/ckb/kbperf.cpp, 1710
src/ckb/kbperf.h, 1710
src/ckb/kbprofile.cpp, 1711
src/ckb/kbprofile.h, 1712
src/ckb/kbprofiledialog.cpp, 1712
src/ckb/kbprofiledialog.h, 1713
src/ckb/kbwidget.cpp, 1714
src/ckb/kbwidget.h, 1714
src/ckb/keyaction.cpp, 1715
src/ckb/keyaction.h, 1716
src/ckb/keymap.cpp, 1717
src/ckb/keymap.h, 1739
src/ckb/keywidget.cpp, 1748
src/ckb/keywidget.h, 1751
src/ckb/kperfwidget.cpp, 1752
src/ckb/kperfwidget.h, 1752
src/ckb/layoutdialog.cpp, 1754
src/ckb/layoutdialog.h, 1755
src/ckb/macroreader.cpp, 1756
src/ckb/macroreader.h, 1756
src/ckb/main.cpp, 1757
src/ckb/mainwindow.cpp, 1763
src/ckb/mainwindow.h, 1764

src/ckb/media.h, 1766
src/ckb/media_linux.cpp, 1768
src/ckb/media_mac.m, 1769
src/ckb/modeselectdialog.cpp, 1770
src/ckb/modeselectdialog.h, 1771
src/ckb/mperfwidget.cpp, 1772
src/ckb/mperfwidget.h, 1774
src/ckb/quazip/JlCompress.cpp, 1787
src/ckb/quazip/JlCompress.h, 1789
src/ckb/quazip/crypt.h, 1776
src/ckb/quazip/ioapi.h, 1780
src/ckb/quazip/qioapi.cpp, 1789
src/ckb/quazip/quaadler32.cpp, 1801
src/ckb/quazip/quaadler32.h, 1802
src/ckb/quazip/quachecksum32.h, 1803
src/ckb/quazip/quacrc32.cpp, 1804
src/ckb/quazip/quacrc32.h, 1805
src/ckb/quazip/quagzipfile.cpp, 1806
src/ckb/quazip/quagzipfile.h, 1807
src/ckb/quazip/quaziodevice.cpp, 1808
src/ckb/quazip/quaziodevice.h, 1809
src/ckb/quazip/quazip.cpp, 1810
src/ckb/quazip/quazip.h, 1812
src/ckb/quazip/quazip_global.h, 1813
src/ckb/quazip/quazipdir.cpp, 1815
src/ckb/quazip/quazipdir.h, 1817
src/ckb/quazip/quazipfile.cpp, 1817
src/ckb/quazip/quazipfile.h, 1818
src/ckb/quazip/quazipfileinfo.cpp, 1819
src/ckb/quazip/quazipfileinfo.h, 1821
src/ckb/quazip/quazipnewinfo.cpp, 1822
src/ckb/quazip/quazipnewinfo.h, 1826
src/ckb/quazip/unzip.c, 1827
src/ckb/quazip/unzip.h, 1878
src/ckb/quazip/zip.c, 1898
src/ckb/quazip/zip.h, 1954
src/ckb/rebindwidget.cpp, 1965
src/ckb/rebindwidget.h, 1965
src/ckb/release/moc_animadddialog.cpp, 1966
src/ckb/release/moc_animscript.cpp, 1969
src/ckb/release/moc_animsettingdialog.cpp, 1972
src/ckb/release/moc_ckbsettingswriter.cpp, 1975
src/ckb/release/moc_colorbutton.cpp, 1977
src/ckb/release/moc_extrasettingswidget.cpp, 1979
src/ckb/release/moc_fwupgradedialog.cpp, 1982
src/ckb/release/moc_gradientbutton.cpp, 1985
src/ckb/release/moc_gradientdialog.cpp, 1988
src/ckb/release/moc_gradientdialogwidget.cpp, 1991
src/ckb/release/moc_kb.cpp, 1993
src/ckb/release/moc_kbanim.cpp, 1996
src/ckb/release/moc_kbanimwidget.cpp, 1998
src/ckb/release/moc_kbbind.cpp, 2001
src/ckb/release/moc_kbbindwidget.cpp, 2003
src/ckb/release/moc_kbfirmware.cpp, 2005
src/ckb/release/moc_kblight.cpp, 2008
src/ckb/release/moc_kblightwidget.cpp, 2011
src/ckb/release/moc_kbmanager.cpp, 2014
src/ckb/release/moc_kbmode.cpp, 2016
src/ckb/release/moc_kbperf.cpp, 2018
src/ckb/release/moc_kbprofile.cpp, 2021
src/ckb/release/moc_kbprofiledialog.cpp, 2023
src/ckb/release/moc_kbwidget.cpp, 2027
src/ckb/release/moc_keyaction.cpp, 2029
src/ckb/release/moc_keywidget.cpp, 2031
src/ckb/release/moc_kperfwidget.cpp, 2033
src/ckb/release/moc_layoutdialog.cpp, 2036
src/ckb/release/moc_macroreader.cpp, 2039
src/ckb/release/moc_mainwindow.cpp, 2043
src/ckb/release/moc_modeselectdialog.cpp, 2045
src/ckb/release/moc_mperfwidget.cpp, 2047
src/ckb/release/moc_quagzipfile.cpp, 2050
src/ckb/release/moc_quaziodevice.cpp, 2052
src/ckb/release/moc_quazipfile.cpp, 2054
src/ckb/release/moc_rebindwidget.cpp, 2056
src/ckb/release/moc_rlistwidget.cpp, 2059
src/ckb/release/moc_settingswidget.cpp, 2061
src/ckb/release/qrc_binary.cpp, 2063
src/ckb/release/qrc_image.cpp, 2066
src/ckb/release/qrc_text.cpp, 2069
src/ckb/rlistwidget.cpp, 2072
src/ckb/rlistwidget.h, 2072
src/ckb/settingswidget.cpp, 2073
src/ckb/settingswidget.h, 2075
src/ckb/ui_animadddialog.h, 2076
src/ckb/ui_animsettingdialog.h, 2076
src/ckb/ui_extrasettingswidget.h, 2077
src/ckb/ui_fwupgradedialog.h, 2078
src/ckb/ui_gradientdialog.h, 2079
src/ckb/ui_kbanimwidget.h, 2080
src/ckb/ui_kbbindwidget.h, 2081
src/ckb/ui_kblightwidget.h, 2082
src/ckb/ui_kbprofiledialog.h, 2083
src/ckb/ui_kbwidget.h, 2084
src/ckb/ui_kperfwidget.h, 2085
src/ckb/ui_layoutdialog.h, 2086
src/ckb/ui_mainwindow.h, 2087
src/ckb/ui_modeselectdialog.h, 2088
src/ckb/ui_mperfwidget.h, 2089
src/ckb/ui_rebindwidget.h, 2090
src/ckb/ui_settingswidget.h, 2091
stages
    MPerfWidget, 799
standardMaps
    keymap.cpp, 1739
start
    devcmd.__unnamed__, 162
start_dev
    device.c, 1236
    device.h, 1246
start_kb_nrgb
    device.h, 1246
    device_keyboard.c, 1251
startWorkInAThread
    MacroReader, 728
stateChange
    MainWindow, 748

statusLabel
Ui::FwUpgradeDialog, 200
Ui_FwUpgradeDialog, 1074
stop
AnimScript, 81
KbAnim, 333
KbManager, 487
stopBox
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
stopCheck
AnimSettingDialog, 119
stopColor
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
stopCount
GradientDialogWidget, 240
stopOpacity
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
stopPos
Ui::GradientDialog, 213
Ui_GradientDialog, 1081
stopPreview
KbLight, 452
stopRect
GradientDialogWidget, 240
stopTime
KbAnim, 338
stopped
AnimScript, 85
stops
GradientDialogWidget, 241
storageName
Key, 604
strLayout
KeyMap, 658
strModel
KeyMap, 659
strcmpcasenosensitive_internal
unzip.c, 1834
strtrim
usb_linux.c, 1598
structures.h
CLEAR_KEYBIT, 1518
DPI_COUNT, 1518
FEAT_ADJRATE, 1518
FEAT_ANSI, 1518
FEAT_BIND, 1518
FEAT_COMMON, 1518
FEAT_FWUPDATE, 1518
FEAT_FWVERSION, 1518
FEAT_HWLOAD, 1519
FEAT_ISO, 1519
FEAT_LMASK, 1519
FEAT_MONOCHROME, 1519
FEAT_MOUSEACCEL, 1519
FEAT_NOTIFY, 1519
FEAT_POLLRATE, 1519
FEAT_RGB, 1519
FEAT_STD_NRGB, 1519
FEAT_STD_RGB, 1520
HAS_ANY_FEATURE, 1520
HAS_FEATURES, 1520
HWMODE_K70, 1520
HWMODE_K95, 1520
HWMODE_MAX, 1520
I_CAPS, 1520
I_NUM, 1520
I_SCROLL, 1520
IFACE_MAX, 1520
KB_NAME_LEN, 1521
LIFT_MAX, 1521
LIFT_MIN, 1521
MACRO_MAX, 1521
MD_NAME_LEN, 1521
MODE_COUNT, 1521
MSG_SIZE, 1521
NEEDS_FW_UPDATE, 1521
OUTFIFO_MAX, 1521
PR_NAME_LEN, 1522
SCROLL_ACCELERATED, 1522
SCROLL_MAX, 1522
SCROLL_MIN, 1522
SERIAL_LEN, 1522
SET_KEYBIT, 1522
vtable_keyboard, 1522
vtable_keyboard_nonrgb, 1522
vtable_mouse, 1522
Subtract
KbAnim, 314
symmetric
ckb-pinwheel/main.c, 1426
ckb-ripple/main.c, 1444
ckb-wave/main.c, 1452

TOGGLE
KeyWidget, 665
TAB_ANIM
RebindWidget, 1006
TAB_KB
RebindWidget, 1006
TAB_MACRO
RebindWidget, 1006
TAB_MOUSE
RebindWidget, 1006
TAB_PROGRAM
RebindWidget, 1006
TAB_SPECIAL
RebindWidget, 1006
TEST_RESET
usb_linux.c, 1585
TRUE
ckb-anim.h, 1660
TRY_WITH_RESET
command.c, 1220
TRYFREE

unzip.c, 1833
zip.c, 1906
tabWidget
Ui::AnimSettingDialog, 98
Ui::KbWidget, 601
Ui::MainWindow, 737
Ui::RebindWidget, 969
Ui_AnimSettingDialog, 1055
Ui_KbWidget, 1118
Ui_MainWindow, 1147
Ui_RebindWidget, 1198
tableDownload
KbFirmware, 424
tableName
kbfirmware.cpp, 1701
target
ckb-random/main.c, 1438
tick
ckb-rain/main.c, 1433
timeGrid
Ui::AnimSettingDialog, 98
Ui_AnimSettingDialog, 1055
timeTab
Ui::AnimSettingDialog, 98
Ui_AnimSettingDialog, 1055
timerTick
MainWindow, 749
RListWidget, 1011
timespec_add
ckb-daemon/main.c, 1403
includes.h, 1303
timespec_eq
includes.h, 1302
timespec_ge
includes.h, 1302
timespec_gt
includes.h, 1302
timespec_le
includes.h, 1302
timespec_lt
includes.h, 1302
tm_unz_s, 1880
tm_zip_s, 1956
tm_unz
unzip.h, 1888
tm_zip
zip.h, 1960
toQuaZipFileInfo
QuaZipFileInfo64, 914
toStorage
KeyMap, 659
toString
GradientButton, 204
toggleSidelight
KbLightWidget, 474
toggleTrayIcon
MainWindow, 750
top
ckb-wave/main.c, 1452
trayBox
Ui::ExtraSettingsWidget, 173
Ui_ExtraSettingsWidget, 1067
trayIcon
MainWindow, 752
trayIconMenu
MainWindow, 752
trigger
KbAnim, 334
txtBuffer
Ui::RebindWidget, 969
Ui_RebindWidget, 1199
Type
KeyAction, 609
type
KeyAction, 631
typingBox
Ui::RebindWidget, 969
Ui_RebindWidget, 1199
typingButton
Ui::RebindWidget, 969
Ui_RebindWidget, 1199
typingKeys
RebindWidget, 1006

u16dec
profile.c, 1481
u16enc
profile.c, 1481
UNBIND
command.h, 1228
UNBOUND
KeyAction, 609
UNKNOWN
KbFirmware, 416
media.h, 1766
UNMUTED
media.h, 1766
US
KeyMap, 638
US_DVORAK
KeyMap, 638
UINPUT_VERSION
os.h, 1467
UNUSED
quazip_global.h, 1815
UNZ_AUTO_CLOSE
unzip.h, 1886
UNZ_BADZIPFILE
unzip.h, 1886
UNZ_BUFSIZE
unzip.c, 1833
UNZ_CRCERROR
unzip.h, 1886
UNZ_DEFAULT_FLAGS
unzip.h, 1886
UNZ_EOF
unzip.h, 1887

UNZ_ERRNO
unzip.h, 1887
UNZ_INTERNALERROR
unzip.h, 1887
UNZ_MAXFILENAMEINZIP
unzip.c, 1833
UNZ_OK
unzip.h, 1887
UNZ_OPENERROR
quazip.h, 1813
UNZ_PARAMERROR
unzip.h, 1887
USB_DELAY_DEFAULT
usb.h, 1555
uchar
includes.h, 1303
udev
usb_linux.c, 1608
udev_enum
usb_linux.c, 1599
udevthread
usb_linux.c, 1608
Ui, 41
ui
AnimAddDialog, 54
AnimSettingDialog, 119
ExtraSettingsWidget, 183
FwUpgradeDialog, 193
GradientDialog, 227
KbAnimWidget, 365
KbBindWidget, 413
KbLightWidget, 476
KbProfileDialog, 565
KbWidget, 592
KPerfWidget, 717
LayoutDialog, 725
MainWindow, 752
ModeSelectDialog, 760
MPerfWidget, 799
RebindWidget, 1006
SettingsWidget, 1023
Ui::AnimAddDialog, 43
animBox, 46
buttonBox, 47
gridLayout, 47
horizontalLayout, 47
horizontalSpacer, 47
horizontalSpacer_2, 47
label, 47
label_2, 47
presetBox, 47
previewBox, 47
retranslateUi, 44
setupUi, 45
verticalSpacer_2, 48
verticalSpacer_3, 48
Ui::AnimSettingDialog, 86
animName, 93
buttonBox, 93
delayBox, 93
gridLayout, 94
gridLayout_2, 94
horizontalSpacer, 94
horizontalSpacer_2, 94
horizontalSpacer_3, 94
horizontalSpacer_4, 94
horizontalSpacer_5, 94
horizontalSpacer_6, 94
infoTab, 94
kpDelayBox, 94
kpModeStopBox, 95
kpReleaseBox, 95
kpRepeatBox, 95
kpRepeatLabel, 95
kpRepeatLabel_2, 95
label, 95
label_10, 95
label_11, 95
label_12, 95
label_13, 96
label_2, 96
label_3, 96
label_4, 96
label_5, 96
label_6, 96
label_7, 96
label_8, 96
label_9, 96
line, 97
line_2, 97
repeatBox, 97
repeatLabel, 97
repeatLabel_2, 97
retranslateUi, 88
sAuthorLabel, 97
sDescLabel, 97
sLicenseLabel, 98
sNameLabel, 98
sVerLabel, 98
sYearLabel, 98
settingTab, 97
settingsGrid, 97
setupUi, 89
tabWidget, 98
timeGrid, 98
timeTab, 98
verticalSpacer, 98
verticalSpacer_2, 98
verticalSpacer_3, 99
Ui::ExtraSettingsWidget, 162
animCountLabel, 170
animPathLabel, 170
animScanButton, 170
brightnessBox, 170
buttonBox, 170
delayBox, 170

ditherBox, 170
fpsBox, 171
fpsWarnLabel, 171
gridLayout, 171
horizontalLayout, 171
horizontalLayout_2, 171
horizontalSpacer, 171
horizontalSpacer_2, 171
horizontalSpacer_3, 171
label, 171
label_10, 172
label_11, 172
label_2, 172
label_3, 172
label_5, 172
label_6, 172
line, 172
line_2, 172
line_3, 172
mAccelBox, 173
osxLabel, 173
osxLine, 173
retranslateUi, 165
sAccelBox, 173
sSpeedBox, 173
sSpeedWidget, 173
setupUi, 166
trayBox, 173
verticalSpacer, 173
verticalSpacer_10, 174
verticalSpacer_11, 174
verticalSpacer_2, 174
verticalSpacer_3, 174
verticalSpacer_4, 174
verticalSpacer_5, 174
verticalSpacer_6, 174
verticalSpacer_7, 174
verticalSpacer_8, 174
verticalSpacer_9, 175
Ui::FwUpgradeDialog, 193
actionButton, 198
cancelButton, 198
curLabel, 198
devLabel, 198
gridLayout, 199
gridLayout_2, 199
horizontalSpacer, 199
horizontalSpacer_2, 199
horizontalSpacer_3, 199
horizontalSpacer_4, 199
label_2, 199
label_4, 199
newLabel, 199
progressBar, 200
retranslateUi, 196
setupUi, 196
statusLabel, 200
Ui::GradientDialog, 206
buttonBox, 211
gridLayout, 211
gridLayout_2, 211
gridLayout_3, 212
groupBox_2, 212
horizontalLayout, 212
horizontalLayout_2, 212
horizontalSpacer, 212
horizontalSpacer_2, 212
horizontalSpacer_3, 212
label, 212
label_2, 212
presetDelete, 213
presetList, 213
presetName, 213
presetSave, 213
retranslateUi, 208
setupUi, 209
stopBox, 213
stopColor, 213
stopOpacity, 213
stopPos, 213
verticalSpacer, 214
verticalSpacer_2, 214
widget, 214
Ui::KPerfWidget, 684
capsBox, 696
capsColorOff, 696
capsColorOn, 696
copyButton, 696
gridLayout, 696
horizontalLayout, 697
horizontalLayout_2, 697
horizontalSpacer, 697
horizontalSpacer_2, 697
horizontalSpacer_3, 697
horizontalSpacer_4, 697
horizontalSpacer_5, 697
horizontalSpacer_6, 697
horizontalSpacer_7, 697
horizontalSpacer_8, 698
intensityBox, 698
k95Label1, 698
k95Label2, 698
k95Label3, 698
k95Label4, 698
k95Label5, 698
k95Label6, 698
k95Line, 698
k95Spacer, 699
label, 699
label_10, 699
label_11, 699
label_12, 699
label_14, 699
label_15, 699
label_16, 699
label_17, 699

label_18, 700
label_19, 700
label_2, 700
label_20, 700
label_21, 700
label_22, 700
label_23, 700
label_24, 700
label_3, 700
label_4, 701
label_5, 701
label_6, 701
label_7, 701
label_8, 701
lightBox, 701
lightColor1, 701
lightColor2, 701
lightColor3, 701
line, 702
line_2, 702
line_3, 702
lockBox, 702
lockColorOff, 702
lockColorOn, 702
macroBox, 702
macroColorOff, 702
macroColorOn, 702
modeBox, 703
modeColorOff, 703
modeColorOn, 703
muteBox, 703
muteColorNA, 703
muteColorOff, 703
muteColorOn, 703
numBox, 703
numColorOff, 703
numColorOn, 704
retranslateUi, 688
scrollBox, 704
scrollColorOff, 704
scrollColorOn, 704
setupUi, 689
verticalSpacer, 704
verticalSpacer_2, 704
verticalSpacer_3, 704
verticalSpacer_4, 704
Ui::KbAnimWidget, 338
aCopyLabel, 344
aMissingLabel, 344
aMissingPage, 344
aNameLabel, 344
aVerLabel, 345
animList, 344
animPage, 344
blendBox, 345
deleteButton, 345
gridLayout, 345
gridLayout_2, 345
gridLayout_3, 345
gridLayout_4, 345
horizontalLayout, 345
horizontalSpacer, 345
horizontalSpacer_2, 346
horizontalSpacer_3, 346
horizontalSpacer_4, 346
horizontalSpacer_5, 346
keyButton, 346
label_5, 346
label_6, 346
label_7, 346
nameBox, 346
noAnimLabel, 347
noSelPage, 347
opacityBox, 347
propertyButton, 347
retranslateUi, 340
selectionStack, 347
setupUi, 341
verticalLayout, 347
verticalSpacer, 347
verticalSpacer_2, 347
verticalSpacer_3, 347
Ui::KbBindWidget, 396
copyButton, 401
gridLayout, 401
gridLayout_2, 401
horizontalLayout_2, 401
horizontalSpacer, 401
horizontalSpacer_2, 401
keyWidget, 401
label, 401
label_2, 402
line, 402
line_2, 402
rbWidget, 402
resetButton, 402
retranslateUi, 398
selectLabel, 402
setupUi, 398
verticalLayout, 402
verticalSpacer, 402
widget, 402
Ui::KbLightWidget, 455
animButton, 461
animWidget, 461
bgButton, 461
brightnessBox, 461
gridLayout, 461
gridLayout_2, 461
gridLayout_4, 461
horizontalLayout, 462
horizontalLayout_2, 462
horizontalLayout_3, 462
horizontalSpacer_3, 462
horizontalSpacer_4, 462
horizontalSpacer_5, 462

horizontalSpacer_6, 462
keyWidget, 462
label, 462
label_2, 463
label_3, 463
line, 463
line_2, 463
retranslateUi, 457
selLabel, 463
setupUi, 458
showAnimBox, 463
verticalLayout, 463
verticalSpacer_2, 463
verticalSpacer_3, 463
widget, 464
Ui::KbProfileDialog, 565
buttonBox, 569
gridLayout, 569
horizontalSpacer, 569
horizontalSpacer_2, 569
profileList, 569
retranslateUi, 567
setupUi, 568
Ui::KbWidget, 592
bindTab, 598
bindWidget, 598
devTab, 598
fwLabel, 598
fwUpdButton, 598
fwUpdLabel, 599
fwUpdLayout, 599
gridLayout, 599
gridLayout_2, 599
gridLayout_3, 599
gridLayout_4, 599
gridLayout_5, 599
gridLayout_7, 599
horizontalSpacer_2, 599
horizontalSpacer_3, 600
hwSaveButton, 600
kPerfTab, 600
kPerfWidget, 600
label_3, 600
label_4, 600
label_9, 600
lightTab, 600
lightWidget, 600
mPerfTab, 601
mPerfWidget, 601
modesList, 601
pollLabel, 601
pollLabel2, 601
profileBox, 601
retranslateUi, 594
serialLabel, 601
setupUi, 595
tabWidget, 601
verticalSpacer, 601
Ui::LayoutDialog, 717
buttonBox, 721
comboBox, 721
gridLayout, 721
horizontalSpacer, 721
horizontalSpacer_2, 721
label, 721
label_2, 722
retranslateUi, 719
setupUi, 719
verticalSpacer, 722
verticalSpacer_2, 722
verticalSpacer_3, 722
verticalSpacer_4, 722
Ui::MPerfWidget, 761
aSnapBox, 773
copyButton, 773
eBox1, 773
eBox2, 773
eBox3, 773
eBox4, 773
eBox5, 773
gridLayout, 773
horizontalLayout, 774
horizontalLayout_2, 774
horizontalLayout_4, 774
horizontalSpacer, 774
horizontalSpacer_2, 774
horizontalSpacer_3, 774
horizontalSpacer_4, 774
horizontalSpacer_5, 774
horizontalSpacer_6, 774
horizontalSpacer_7, 775
iButton0, 775
iButton1, 775
iButton2, 775
iButton3, 775
iButton4, 775
iButton5, 775
iButtonO, 775
indicBox, 775
lHeightBox, 777
label, 776
label_11, 776
label_12, 776
label_13, 776
label_14, 776
label_15, 776
label_2, 776
label_25, 776
label_3, 776
label_4, 777
label_5, 777
label_6, 777
label_7, 777
label_8, 777
line, 777
line_2, 777

line_3, 777
retranslateUi, 764
setupUi, 765
spinBox, 778
verticalLayout, 778
verticalSpacer, 778
verticalSpacer_2, 778
verticalSpacer_3, 778
xBox0, 778
xBox1, 778
xBox2, 778
xBox3, 778
xBox4, 779
xBox5, 779
xSlider0, 779
xSlider1, 779
xSlider2, 779
xSlider3, 779
xSlider4, 779
xSlider5, 779
xyBox, 779
yBox0, 780
yBox1, 780
yBox2, 780
yBox3, 780
yBox4, 780
yBox5, 780
ySlider0, 780
ySlider1, 780
ySlider2, 780
ySlider3, 781
ySlider4, 781
ySlider5, 781
Ui::MainWindow, 733
actionAbout, 736
actionExit, 736
centralWidget, 736
gridLayout, 737
retranslateUi, 735
setupUi, 735
tabWidget, 737
Ui::ModeSelectDialog, 753
buttonBox, 756
gridLayout, 756
label, 756
modeList, 756
retranslateUi, 754
selAllButton, 756
selNoneButton, 757
setupUi, 755
Ui::RebindWidget, 940
actionTab, 956
animBox, 956
animButton, 956
animKrBox, 957
animOnceBox, 957
animTab, 957
applyButton, 957
btnClearMacro, 957
btnStartMacro, 957
btnStopMacro, 957
cancelButton, 957
delayButtons, 957
dpiBox, 958
dpiButton, 958
dpiCustLabel, 958
dpiCustXBox, 958
dpiCustYBox, 958
dpiWarning, 958
fnBox, 958
fnButton, 958
gridLayout, 959
gridLayout_10, 959
gridLayout_2, 959
gridLayout_3, 959
gridLayout_4, 959
gridLayout_5, 959
gridLayout_8, 959
groupBox_2, 959
groupBox_3, 959
groupBox_4, 960
horizontalLayout, 960
horizontalLayout_2, 960
horizontalLayout_3, 960
horizontalSpacer, 960
horizontalSpacer_10, 960
horizontalSpacer_11, 960
horizontalSpacer_12, 960
horizontalSpacer_13, 960
horizontalSpacer_14, 961
horizontalSpacer_15, 961
horizontalSpacer_16, 961
horizontalSpacer_17, 961
horizontalSpacer_18, 961
horizontalSpacer_19, 961
horizontalSpacer_2, 961
horizontalSpacer_20, 961
horizontalSpacer_21, 961
horizontalSpacer_22, 962
horizontalSpacer_23, 962
horizontalSpacer_24, 962
horizontalSpacer_25, 962
horizontalSpacer_26, 962
horizontalSpacer_3, 962
horizontalSpacer_4, 962
horizontalSpacer_5, 962
horizontalSpacer_6, 962
horizontalSpacer_7, 963
horizontalSpacer_8, 963
horizontalSpacer_9, 963
keyTab, 963
label, 963
layoutWidget, 963
layoutWidget1, 963
layoutWidget2, 963
layoutWidget_2, 963

lbl_macro, 964
lightBox, 964
lightButton, 964
lightWrapBox, 964
lockBox, 964
lockButton, 964
macroTab, 964
mb2Box, 964
mb2Button, 965
mbBox, 965
mbButton, 965
mediaBox, 965
mediaButton, 965
modBox, 965
modButton, 965
modeBox, 966
modeButton, 966
modeWrapBox, 966
mouseTab, 966
numBox, 966
numButton, 966
progTipLabel, 968
programKpBox, 966
programKpButton, 966
programKpExtra, 967
programKpModeBox, 967
programKpSIBox, 967
programKrBox, 967
programKrButton, 967
programKrExtra, 967
programKrModeBox, 967
programKrSIBox, 967
programTab, 968
pteMacroBox, 968
pteMacroComment, 968
pteMacroText, 968
rb_delay_asTyped, 968
rb_delay_default, 968
rb_delay_no, 968
resetButton, 969
retranslateUi, 943
setupUi, 946
tabWidget, 969
txtBuffer, 969
typingBox, 969
typingButton, 969
unbindButton, 969
verticalLayout, 969
verticalLayout_2, 969
verticalSpacer, 970
verticalSpacer_10, 970
verticalSpacer_11, 970
verticalSpacer_2, 970
verticalSpacer_3, 970
verticalSpacer_4, 970
verticalSpacer_5, 970
verticalSpacer_6, 970
verticalSpacer_7, 970
verticalSpacer_8, 971
wheelBox, 971
wheelButton, 971
Ui::SettingsWidget, 1024
altBox, 1031
altLabel, 1031
autoFWBox, 1031
capsBox, 1031
capsLabel, 1032
ctrlBox, 1032
ctrlLabel, 1032
devicesLabel, 1032
extraButton, 1032
gridLayout, 1032
gridLayout_3, 1032
gridLayout_5, 1032
horizontalLayout, 1032
horizontalLayout_2, 1033
horizontalSpacer, 1033
horizontalSpacer_10, 1033
horizontalSpacer_2, 1033
horizontalSpacer_7, 1033
horizontalSpacer_9, 1033
label, 1033
label_2, 1033
label_3, 1033
label_5, 1033
label_6, 1034
label_8, 1034
layoutBox, 1034
line, 1034
line_2, 1034
line_3, 1034
loginItemBox, 1034
pushButton, 1034
retranslateUi, 1026
setupUi, 1027
shiftBox, 1034
shiftLabel, 1035
versionLabel, 1035
verticalSpacer, 1035
verticalSpacer_10, 1035
verticalSpacer_2, 1035
verticalSpacer_3, 1035
verticalSpacer_4, 1035
verticalSpacer_5, 1035
verticalSpacer_6, 1035
verticalSpacer_7, 1036
verticalSpacer_8, 1036
winBox, 1036
winLabel, 1036
Ui_AnimAddDialog, 1039
animBox, 1042
buttonBox, 1042
gridLayout, 1042
horizontalLayout, 1042
horizontalSpacer, 1042
horizontalSpacer_2, 1042

label, 1043
label_2, 1043
presetBox, 1043
previewBox, 1043
retranslateUi, 1040
setupUi, 1040
verticalSpacer_2, 1043
verticalSpacer_3, 1043
Ui_AnimSettingDialog, 1043
animName, 1050
buttonBox, 1050
delayBox, 1051
gridLayout, 1051
gridLayout_2, 1051
horizontalSpacer, 1051
horizontalSpacer_2, 1051
horizontalSpacer_3, 1051
horizontalSpacer_4, 1051
horizontalSpacer_5, 1051
horizontalSpacer_6, 1051
infoTab, 1052
kpDelayBox, 1052
kpModeStopBox, 1052
kpReleaseBox, 1052
kpRepeatBox, 1052
kpRepeatLabel, 1052
kpRepeatLabel_2, 1052
label, 1052
label_10, 1052
label_11, 1053
label_12, 1053
label_13, 1053
label_2, 1053
label_3, 1053
label_4, 1053
label_5, 1053
label_6, 1053
label_7, 1053
label_8, 1054
label_9, 1054
line, 1054
line_2, 1054
repeatBox, 1054
repeatLabel, 1054
repeatLabel_2, 1054
retranslateUi, 1046
sAuthorLabel, 1054
sDescLabel, 1054
sLicenseLabel, 1055
sNameLabel, 1055
sVerLabel, 1055
sYearLabel, 1055
settingTab, 1055
settingsGrid, 1055
setupUi, 1047
tabWidget, 1055
timeGrid, 1055
timeTab, 1055
verticalSpacer, 1056
verticalSpacer_2, 1056
verticalSpacer_3, 1056
Ui_ExtraSettingsWidget, 1056
animCountLabel, 1063
animPathLabel, 1064
animScanButton, 1064
brightnessBox, 1064
buttonBox, 1064
delayBox, 1064
ditherBox, 1064
fpsBox, 1064
fpsWarnLabel, 1064
gridLayout, 1064
horizontalLayout, 1064
horizontalLayout_2, 1065
horizontalSpacer, 1065
horizontalSpacer_2, 1065
horizontalSpacer_3, 1065
label, 1065
label_10, 1065
label_11, 1065
label_2, 1065
label_3, 1065
label_5, 1066
label_6, 1066
line, 1066
line_2, 1066
line_3, 1066
mAccelBox, 1066
osxLabel, 1066
osxLine, 1066
retranslateUi, 1059
sAccelBox, 1066
sSpeedBox, 1067
sSpeedWidget, 1067
setupUi, 1060
trayBox, 1067
verticalSpacer, 1067
verticalSpacer_10, 1067
verticalSpacer_11, 1067
verticalSpacer_2, 1067
verticalSpacer_3, 1067
verticalSpacer_4, 1067
verticalSpacer_5, 1068
verticalSpacer_6, 1068
verticalSpacer_7, 1068
verticalSpacer_8, 1068
verticalSpacer_9, 1068
Ui_FwUpgradeDialog, 1068
actionButton, 1072
cancelButton, 1073
curLabel, 1073
devLabel, 1073
gridLayout, 1073
gridLayout_2, 1073
horizontalSpacer, 1073
horizontalSpacer_2, 1073

horizontalSpacer_3, 1073
horizontalSpacer_4, 1073
label_2, 1074
label_4, 1074
newLabel, 1074
progressBar, 1074
retranslateUi, 1070
setupUi, 1070
statusLabel, 1074
Ui_GradientDialog, 1074
buttonBox, 1079
gridLayout, 1079
gridLayout_2, 1079
gridLayout_3, 1079
groupBox_2, 1080
horizontalLayout, 1080
horizontalLayout_2, 1080
horizontalSpacer, 1080
horizontalSpacer_2, 1080
horizontalSpacer_3, 1080
label, 1080
label_2, 1080
presetDelete, 1080
presetList, 1081
presetName, 1081
presetSave, 1081
retranslateUi, 1076
setupUi, 1077
stopBox, 1081
stopColor, 1081
stopOpacity, 1081
stopPos, 1081
verticalSpacer, 1081
verticalSpacer_2, 1082
widget, 1082
Ui_KPerfWidget, 1119
capsBox, 1130
capsColorOff, 1130
capsColorOn, 1130
copyButton, 1130
gridLayout, 1130
horizontalLayout, 1130
horizontalLayout_2, 1130
horizontalSpacer, 1130
horizontalSpacer_2, 1131
horizontalSpacer_3, 1131
horizontalSpacer_4, 1131
horizontalSpacer_5, 1131
horizontalSpacer_6, 1131
horizontalSpacer_7, 1131
horizontalSpacer_8, 1131
intensityBox, 1131
k95Label1, 1131
k95Label2, 1132
k95Label3, 1132
k95Label4, 1132
k95Label5, 1132
k95Label6, 1132
k95Line, 1132
k95Spacer, 1132
label, 1132
label_10, 1132
label_11, 1133
label_12, 1133
label_14, 1133
label_15, 1133
label_16, 1133
label_17, 1133
label_18, 1133
label_19, 1133
label_2, 1133
label_20, 1134
label_21, 1134
label_22, 1134
label_23, 1134
label_24, 1134
label_3, 1134
label_4, 1134
label_5, 1134
label_6, 1134
label_7, 1135
label_8, 1135
lightBox, 1135
lightColor1, 1135
lightColor2, 1135
lightColor3, 1135
line, 1135
line_2, 1135
line_3, 1135
lockBox, 1136
lockColorOff, 1136
lockColorOn, 1136
macroBox, 1136
macroColorOff, 1136
macroColorOn, 1136
modeBox, 1136
modeColorOff, 1136
modeColorOn, 1136
muteBox, 1137
muteColorNA, 1137
muteColorOff, 1137
muteColorOn, 1137
numBox, 1137
numColorOff, 1137
numColorOn, 1137
retranslateUi, 1122
scrollBox, 1137
scrollColorOff, 1137
scrollColorOn, 1138
setupUi, 1123
verticalSpacer, 1138
verticalSpacer_2, 1138
verticalSpacer_3, 1138
verticalSpacer_4, 1138
Ui_KbAnimWidget, 1082
aCopyLabel, 1087

aMissingLabel, 1087
aMissingPage, 1087
aNameLabel, 1087
aVerLabel, 1088
animList, 1087
animPage, 1088
blendBox, 1088
deleteButton, 1088
gridLayout, 1088
gridLayout_2, 1088
gridLayout_3, 1088
gridLayout_4, 1088
horizontalLayout, 1088
horizontalSpacer, 1089
horizontalSpacer_2, 1089
horizontalSpacer_3, 1089
horizontalSpacer_4, 1089
horizontalSpacer_5, 1089
keyButton, 1089
label_5, 1089
label_6, 1089
label_7, 1089
nameBox, 1090
noAnimLabel, 1090
noSelPage, 1090
opacityBox, 1090
propertyButton, 1090
retranslateUi, 1084
selectionStack, 1090
setupUi, 1084
verticalLayout, 1090
verticalSpacer, 1090
verticalSpacer_2, 1090
verticalSpacer_3, 1091
Ui_KbBindWidget, 1091
copyButton, 1095
gridLayout, 1095
gridLayout_2, 1095
horizontalLayout_2, 1095
horizontalSpacer, 1096
horizontalSpacer_2, 1096
keyWidget, 1096
label, 1096
label_2, 1096
line, 1096
line_2, 1096
rbWidget, 1096
resetButton, 1096
retranslateUi, 1093
selectLabel, 1097
setupUi, 1093
verticalLayout, 1097
verticalSpacer, 1097
widget, 1097
Ui_KbLightWidget, 1097
animButton, 1102
animWidget, 1103
bgButton, 1103
brightnessBox, 1103
gridLayout, 1103
gridLayout_2, 1103
gridLayout_4, 1103
horizontalLayout, 1103
horizontalLayout_2, 1103
horizontalLayout_3, 1103
horizontalSpacer_3, 1104
horizontalSpacer_4, 1104
horizontalSpacer_5, 1104
horizontalSpacer_6, 1104
keyWidget, 1104
label, 1104
label_2, 1104
label_3, 1104
line, 1104
line_2, 1105
retranslateUi, 1099
selLabel, 1105
setupUi, 1100
showAnimBox, 1105
verticalLayout, 1105
verticalSpacer_2, 1105
verticalSpacer_3, 1105
widget, 1105
Ui_KbProfileDialog, 1105
buttonBox, 1108
gridLayout, 1108
horizontalSpacer, 1109
horizontalSpacer_2, 1109
profileList, 1109
retranslateUi, 1107
setupUi, 1107
Ui_KbWidget, 1109
bindTab, 1115
bindWidget, 1115
devTab, 1115
fwLabel, 1115
fwUpdButton, 1115
fwUpdLabel, 1115
fwUpdLayout, 1116
gridLayout, 1116
gridLayout_2, 1116
gridLayout_3, 1116
gridLayout_4, 1116
gridLayout_5, 1116
gridLayout_7, 1116
horizontalSpacer_2, 1116
horizontalSpacer_3, 1116
hwSaveButton, 1117
kPerfTab, 1117
kPerfWidget, 1117
label_3, 1117
label_4, 1117
label_9, 1117
lightTab, 1117
lightWidget, 1117
mPerfTab, 1118

mPerfWidget, 1118
modesList, 1117
pollLabel, 1118
pollLabel2, 1118
profileBox, 1118
retranslateUi, 1111
serialLabel, 1118
setupUi, 1112
tabWidget, 1118
verticalSpacer, 1118
Ui_LayoutDialog, 1138
buttonBox, 1142
comboBox, 1142
gridLayout, 1142
horizontalSpacer, 1142
horizontalSpacer_2, 1142
label, 1142
label_2, 1143
retranslateUi, 1140
setupUi, 1140
verticalSpacer, 1143
verticalSpacer_2, 1143
verticalSpacer_3, 1143
verticalSpacer_4, 1143
Ui_MPerfWidget, 1151
aSnapBox, 1162
copyButton, 1162
eBox1, 1163
eBox2, 1163
eBox3, 1163
eBox4, 1163
eBox5, 1163
gridLayout, 1163
horizontalLayout, 1163
horizontalLayout_2, 1163
horizontalLayout_4, 1163
horizontalSpacer, 1163
horizontalSpacer_2, 1164
horizontalSpacer_3, 1164
horizontalSpacer_4, 1164
horizontalSpacer_5, 1164
horizontalSpacer_6, 1164
horizontalSpacer_7, 1164
iButton0, 1164
iButton1, 1164
iButton2, 1164
iButton3, 1165
iButton4, 1165
iButton5, 1165
iButtonO, 1165
indicBox, 1165
lHeightBox, 1167
label, 1165
label_11, 1165
label_12, 1165
label_13, 1165
label_14, 1166
label_15, 1166
label_2, 1166
label_25, 1166
label_3, 1166
label_4, 1166
label_5, 1166
label_6, 1166
label_7, 1166
label_8, 1167
line, 1167
line_2, 1167
line_3, 1167
retranslateUi, 1154
setupUi, 1155
spinBox, 1167
verticalLayout, 1167
verticalSpacer, 1167
verticalSpacer_2, 1167
verticalSpacer_3, 1168
xBox0, 1168
xBox1, 1168
xBox2, 1168
xBox3, 1168
xBox4, 1168
xBox5, 1168
xSlider0, 1168
xSlider1, 1168
xSlider2, 1169
xSlider3, 1169
xSlider4, 1169
xSlider5, 1169
xyBox, 1169
yBox0, 1169
yBox1, 1169
yBox2, 1169
yBox3, 1169
yBox4, 1170
yBox5, 1170
ySlider0, 1170
ySlider1, 1170
ySlider2, 1170
ySlider3, 1170
ySlider4, 1170
ySlider5, 1170
Ui_MainWindow, 1143
actionAbout, 1146
actionExit, 1146
centralWidget, 1146
gridLayout, 1146
retranslateUi, 1145
setupUi, 1145
tabWidget, 1147
Ui_ModeSelectDialog, 1147
buttonBox, 1150
gridLayout, 1150
label, 1150
modeList, 1150
retranslateUi, 1148
selAllButton, 1150

selNoneButton, 1151
setupUi, 1149
Ui_RebindWidget, 1171
actionTab, 1186
animBox, 1186
animButton, 1187
animKrBox, 1187
animOnceBox, 1187
animTab, 1187
applyButton, 1187
btnClearMacro, 1187
btnStartMacro, 1187
btnStopMacro, 1187
cancelButton, 1187
delayButtons, 1188
dpiBox, 1188
dpiButton, 1188
dpiCustLabel, 1188
dpiCustXBox, 1188
dpiCustYBox, 1188
dpiWarning, 1188
fnBox, 1188
fnButton, 1189
gridLayout, 1189
gridLayout_10, 1189
gridLayout_2, 1189
gridLayout_3, 1189
gridLayout_4, 1189
gridLayout_5, 1189
gridLayout_8, 1189
groupBox_2, 1189
groupBox_3, 1190
groupBox_4, 1190
horizontalLayout, 1190
horizontalLayout_2, 1190
horizontalLayout_3, 1190
horizontalSpacer, 1190
horizontalSpacer_10, 1190
horizontalSpacer_11, 1190
horizontalSpacer_12, 1190
horizontalSpacer_13, 1191
horizontalSpacer_14, 1191
horizontalSpacer_15, 1191
horizontalSpacer_16, 1191
horizontalSpacer_17, 1191
horizontalSpacer_18, 1191
horizontalSpacer_19, 1191
horizontalSpacer_2, 1191
horizontalSpacer_20, 1191
horizontalSpacer_21, 1192
horizontalSpacer_22, 1192
horizontalSpacer_23, 1192
horizontalSpacer_24, 1192
horizontalSpacer_25, 1192
horizontalSpacer_26, 1192
horizontalSpacer_3, 1192
horizontalSpacer_4, 1192
horizontalSpacer_5, 1192
horizontalSpacer_6, 1193
horizontalSpacer_7, 1193
horizontalSpacer_8, 1193
horizontalSpacer_9, 1193
keyTab, 1193
label, 1193
layoutWidget, 1193
layoutWidget1, 1193
layoutWidget2, 1193
layoutWidget_2, 1194
lbl_macro, 1194
lightBox, 1194
lightButton, 1194
lightWrapBox, 1194
lockBox, 1194
lockButton, 1194
macroTab, 1194
mb2Box, 1195
mb2Button, 1195
mbBox, 1195
mbButton, 1195
mediaBox, 1195
mediaButton, 1195
modBox, 1195
modButton, 1195
modeBox, 1195
modeButton, 1196
modeWrapBox, 1196
mouseTab, 1196
numBox, 1196
numButton, 1196
progTipLabel, 1197
programKpBox, 1196
programKpButton, 1196
programKpExtra, 1196
programKpModeBox, 1197
programKpSIBox, 1197
programKrBox, 1197
programKrButton, 1197
programKrExtra, 1197
programKrModeBox, 1197
programKrSIBox, 1197
programTab, 1197
pteMacroBox, 1198
pteMacroComment, 1198
pteMacroText, 1198
rb_delay_asTyped, 1198
rb_delay_default, 1198
rb_delay_no, 1198
resetButton, 1198
retranslateUi, 1174
setupUi, 1176
tabWidget, 1198
txtBuffer, 1199
typingBox, 1199
typingButton, 1199
unbindButton, 1199
verticalLayout, 1199

verticalLayout_2, 1199
verticalSpacer, 1199
verticalSpacer_10, 1199
verticalSpacer_11, 1199
verticalSpacer_2, 1200
verticalSpacer_3, 1200
verticalSpacer_4, 1200
verticalSpacer_5, 1200
verticalSpacer_6, 1200
verticalSpacer_7, 1200
verticalSpacer_8, 1200
wheelBox, 1200
wheelButton, 1200
Ui_SettingsWidget, 1201
altBox, 1208
altLabel, 1208
autoFWBox, 1208
capsBox, 1208
capsLabel, 1208
ctrlBox, 1209
ctrlLabel, 1209
devicesLabel, 1209
extraButton, 1209
gridLayout, 1209
gridLayout_3, 1209
gridLayout_5, 1209
horizontalLayout, 1209
horizontalLayout_2, 1209
horizontalSpacer, 1209
horizontalSpacer_10, 1210
horizontalSpacer_2, 1210
horizontalSpacer_7, 1210
horizontalSpacer_9, 1210
label, 1210
label_2, 1210
label_3, 1210
label_5, 1210
label_6, 1210
label_8, 1211
layoutBox, 1211
line, 1211
line_2, 1211
line_3, 1211
loginItemBox, 1211
pushButton, 1211
retranslateUi, 1203
setupUi, 1204
shiftBox, 1211
shiftLabel, 1211
versionLabel, 1212
verticalSpacer, 1212
verticalSpacer_10, 1212
verticalSpacer_2, 1212
verticalSpacer_3, 1212
verticalSpacer_4, 1212
verticalSpacer_5, 1212
verticalSpacer_6, 1212
verticalSpacer_7, 1212
verticalSpacer_8, 1213
winBox, 1213
winLabel, 1213
uiUpdated
KPerfWidget, 715
uinputopen
input_linux.c, 1334
unbind
devcmd.__unnamed__, 162
unbindButton
Ui::RebindWidget, 969
Ui_RebindWidget, 1199
uncompressedSize
QuaZipFileInfo, 909
QuaZipFileInfo64, 916
QuaZipFilePrivate, 921
QuaZipNewInfo, 930
unz64_file_pos_s, 1885
unz64_s, 1831
unz64_file_pos
unzip.h, 1888
unz64local_CheckCurrentFileCoherencyHeader
unzip.c, 1834
unz64local_DosDateToTmuDate
unzip.c, 1836
unz64local_GetCurrentFileInfoInternal
unzip.c, 1837
unz64local_SearchCentralDir
unzip.c, 1844
unz64local_SearchCentralDir64
unzip.c, 1845
unz64local_getByte
unzip.c, 1837
unz64local_getLong
unzip.c, 1841
unz64local_getLong64
unzip.c, 1842
unz64local_getShort
unzip.c, 1843
unz_file_info64_internal_s, 1830
unz_file_info64_s, 1882
unz_file_info_s, 1883
unz_file_pos_s, 1885
unz_global_info64_s, 1880
unz_global_info_s, 1881
unz_copyright
unzip.c, 1877
unz_file_info
unzip.h, 1888
unz_file_info64
unzip.h, 1888
unz_file_info64_internal
unzip.c, 1834
unz_file_pos
unzip.h, 1888
unz_global_info
unzip.h, 1888
unz_global_info64

unzip.h, 1888
unzClearFlags
unzip.c, 1847
unzip.h, 1889
unzClose
unzip.c, 1848
unzCloseCurrentFile
unzip.c, 1849
unzFile
unzip.h, 1888
unzGetCurrentFileInfo
unzip.c, 1850
unzGetCurrentFileInfo64
unzip.c, 1851
unzGetCurrentFileZStreamPos64
unzip.c, 1852
unzGetFilePos
unzip.c, 1852
unzip.h, 1889
unzGetFilePos64
unzip.c, 1853
unzip.h, 1890
unzGetGlobalComment
unzip.c, 1853
unzGetGlobalInfo
unzip.c, 1854
unzGetGlobalInfo64
unzip.c, 1854
unzGetLocalExtrafield
unzip.c, 1855
unzGetOffset
unzip.c, 1855
unzip.h, 1890
unzGetOffset64
unzip.c, 1856
unzip.h, 1891
unzGoToFilePos
unzip.c, 1856
unzip.h, 1891
unzGoToFilePos64
unzip.c, 1857
unzip.h, 1892
unzGoToFirstFile
unzip.c, 1858
unzGoToNextFile
unzip.c, 1859
unzLocateFile
unzip.c, 1860
unzOpen
unzip.c, 1861
unzOpen2
unzip.c, 1861
unzOpen2_64
unzip.c, 1862
unzOpen64
unzip.c, 1863
unzOpenCurrentFile
unzip.c, 1863
unzOpenCurrentFile2
unzip.c, 1864
unzOpenCurrentFile3
unzip.c, 1864
unzOpenCurrentFilePassword
unzip.c, 1867
unzOpenInternal
unzip.c, 1868
unzip.h, 1893
unzReadCurrentFile
unzip.c, 1871
unzSetFlags
unzip.c, 1874
unzip.h, 1896
unzSetOffset
unzip.c, 1875
unzip.h, 1897
unzSetOffset64
unzip.c, 1875
unzip.h, 1897
unzStringFileNameCompare
unzip.c, 1876
unzeof
unzip.c, 1850
unzip.c
_Z_OF, 1834
ALLOC, 1832
BUFREADCOMMENT, 1833
local, 1833
SIZECENTRALDIRITEM, 1833
SIZEZIPLOCALHEADER, 1833
strcmpcasenosensitive_internal, 1834
TRYFREE, 1833
UNZ_BUFSIZE, 1833
UNZ_MAXFILENAMEINZIP, 1833
unz64local_CheckCurrentFileCoherencyHeader, 1834
unz64local_DosDateToTmuDate, 1836
unz64local_GetCurrentFileInfoInternal, 1837
unz64local_SearchCentralDir, 1844
unz64local_SearchCentralDir64, 1845
unz64local_getByte, 1837
unz64local_getLong, 1841
unz64local_getLong64, 1842
unz64local_getShort, 1843
unz_copyright, 1877
unz_file_info64_internal, 1834
unzClearFlags, 1847
unzClose, 1848
unzCloseCurrentFile, 1849
unzGetCurrentFileInfo, 1850
unzGetCurrentFileInfo64, 1851
unzGetCurrentFileZStreamPos64, 1852
unzGetFilePos, 1852
unzGetFilePos64, 1853
unzGetGlobalComment, 1853
unzGetGlobalInfo, 1854
unzGetGlobalInfo64, 1854

unzGetLocalExtrafield, 1855
unzGetOffset, 1855
unzGetOffset64, 1856
unzGoToFilePos, 1856
unzGoToFilePos64, 1857
unzGoToFirstFile, 1858
unzGoToNextFile, 1859
unzLocateFile, 1860
unzOpen, 1861
unzOpen2, 1861
unzOpen2_64, 1862
unzOpen64, 1863
unzOpenCurrentFile, 1863
unzOpenCurrentFile2, 1864
unzOpenCurrentFile3, 1864
unzOpenCurrentFilePassword, 1867
unzOpenInternal, 1868
unzReadCurrentFile, 1871
unzSetFlags, 1874
unzSetOffset, 1875
unzSetOffset64, 1875
unzStringFileNameCompare, 1876
unzeof, 1850
unztell, 1876
unztell64, 1877
z_crc_t, 1834
unzip.h
_Z_OF, 1888, 1889
tm_unz, 1888
UNZ_AUTO_CLOSE, 1886
UNZ_BADZIPFILE, 1886
UNZ_CRCERROR, 1886
UNZ_DEFAULT_FLAGS, 1886
UNZ_EOF, 1887
UNZ_ERRNO, 1887
UNZ_INTERNALERROR, 1887
UNZ_OK, 1887
UNZ_PARAMERROR, 1887
unz64_file_pos, 1888
unz_file_info, 1888
unz_file_info64, 1888
unz_file_pos, 1888
unz_global_info, 1888
unz_global_info64, 1888
unzClearFlags, 1889
unzFile, 1888
unzGetFilePos, 1889
unzGetFilePos64, 1890
unzGetOffset, 1890
unzGetOffset64, 1891
unzGoToFilePos, 1891
unzGoToFilePos64, 1892
unzOpenInternal, 1893
unzSetFlags, 1896
unzSetOffset, 1897
unzSetOffset64, 1897
Z_BZIP2ED, 1887
unztell
unzip.c, 1876
unztell64
unzip.c, 1877
update
KbBind, 392
KbPerf, 534
QuaAdler32, 812
QuaChecksum32, 814
QuaCrc32, 818
update_keys
crypt.h, 1779
updateBind
KbBindWidget, 410
updateFwButton
KbWidget, 589
updateImage
ColorButton, 149
GradientButton, 205
updateLayout
Kb, 305
updateLight
KbLightWidget, 475
updateMapper
AnimSettingDialog, 119
KPerfWidget, 717
updateModifiers
SettingsWidget, 1022
updateParam
AnimSettingDialog, 114
updateParams
KbAnim, 335
updatePresets
GradientDialog, 225
updateProfileList
KbWidget, 590
updateSelDisplay
KbBindWidget, 412
updateStops
AnimSettingDialog, 117
updateVersion
MainWindow, 751
updateconnected
devnode.c, 1270
devnode.h, 1279
updated
KbBind, 394
KbLight, 453
KbMode, 502
updatedpi
devcmd.__unnamed__, 162
dpi.c, 1284
dpi.h, 1290
updateindicators
devcmd.__unnamed__, 162
updateindicators_kb
input.c, 1316
input.h, 1328
updatergb

devcmd.__unnamed__, 162
updatergb_kb
led.h, 1356
led_keyboard.c, 1388
updatergb_mouse
led.h, 1357
led_mouse.c, 1394
url
KbFirmware::FW, 184
urlParam
animscript.cpp, 1643
urldecode
ckb-anim.h, 1672
urldecode2
profile.c, 1482
urlencode2
profile.c, 1483
usb.c
_resetusb, 1524
_setupusb, 1524
_usbrecv, 1528
_usbsend, 1530
closeusb, 1531
devmain, 1533
features_mask, 1541
get_vtable, 1535
hwload_mode, 1541
product_str, 1536
reset_stop, 1541
revertusb, 1537
setupusb, 1538
usb_tryreset, 1539
usbmutex, 1541
vendor_str, 1540
usb.h
_nk95cmd, 1556
_resetusb, 1557
_usbrecv, 1558
_usbsend, 1559
closeusb, 1561
DELAY_LONG, 1546
DELAY_MEDIUM, 1546
DELAY_SHORT, 1546
IS_FULLRANGE, 1546
IS_GLAIVE, 1546
IS_HARPOON, 1546
IS_K63, 1546
IS_K65, 1547
IS_K68, 1547
IS_K70, 1547
IS_K95, 1547
IS_M65, 1547
IS_MONOCHROME, 1547
IS_MONOCHROME_DEV, 1547
IS_MOUSE, 1547
IS_MOUSE_DEV, 1548
IS_PLATINUM, 1548
IS_RGB, 1548
IS_RGB_DEV, 1548
IS_SABRE, 1548
IS_SCIMITAR, 1548
IS_STRAFE, 1548
IS_V2_OVERRIDE, 1548
NK95_HWOFF, 1548
NK95_HWON, 1549
NK95_M1, 1549
NK95_M2, 1549
NK95_M3, 1549
nk95cmd, 1549
os_closeusb, 1563
os_inputmain, 1564
os_resetusb, 1567
os_sendindicators, 1568
os_setupusb, 1569
os_usbrecv, 1572
os_usbsend, 1574
P_GLAIVE, 1549
P_GLAIVE_STR, 1549
P_HARPOON, 1549
P_HARPOON_STR, 1549
P_K63_NRGB, 1550
P_K63_NRGB_STR, 1550
P_K65, 1550
P_K65_LUX, 1550
P_K65_LUX_STR, 1550
P_K65_NRGB, 1550
P_K65_NRGB_STR, 1550
P_K65_RFIRE, 1550
P_K65_RFIRE_STR, 1550
P_K65_STR, 1550
P_K68, 1550
P_K68_STR, 1551
P_K70, 1551
P_K70_LUX, 1551
P_K70_LUX_NRGB, 1551
P_K70_LUX_NRGB_STR, 1551
P_K70_LUX_STR, 1551
P_K70_NRGB, 1551
P_K70_NRGB_STR, 1551
P_K70_RFIRE, 1551
P_K70_RFIRE_NRGB, 1551
P_K70_RFIRE_STR, 1552
P_K70_STR, 1552
P_K95, 1552
P_K95_NRGB, 1552
P_K95_NRGB_STR, 1552
P_K95_PLATINUM, 1552
P_K95_PLATINUM_STR, 1552
P_K95_STR, 1552
P_M65, 1552
P_M65_PRO, 1552
P_M65_PRO_STR, 1553
P_M65_STR, 1553
P_SABRE_L, 1553
P_SABRE_L_STR, 1553
P_SABRE_N, 1553

P_SABRE_N_STR, 1553
P_SABRE_O, 1553
P_SABRE_O2, 1553
P_SABRE_O2_STR, 1553
P_SABRE_O_STR, 1553
P_SCIMITAR, 1553
P_SCIMITAR_PRO, 1554
P_SCIMITAR_PRO_STR, 1554
P_SCIMITAR_STR, 1554
P_STRAFE, 1554
P_STRAFE_NRGB, 1554
P_STRAFE_NRGB_2, 1554
P_STRAFE_NRGB_STR, 1554
P_STRAFE_STR, 1554
product_str, 1576
resetusb, 1554
revertusb, 1577
setupusb, 1578
USB_DELAY_DEFAULT, 1555
usb_tryreset, 1579
usbkill, 1580
usbmain, 1581
usbrecv, 1555
usbsend, 1555
V_CORSAIR, 1555
V_CORSAIR_STR, 1555
vendor_str, 1582
usb_add_device
usb_linux.c, 1600
usb_linux.c
_nk95cmd, 1586
DEBUG, 1585
kbsyspath, 1608
models, 1608
N_MODELS, 1585
os_closeusb, 1587
os_inputmain, 1587
os_resetusb, 1591
os_sendindicators, 1592
os_setupusb, 1593
os_usbrecv, 1595
os_usbsend, 1596
strtrim, 1598
TEST_RESET, 1585
udev, 1608
udev_enum, 1599
udevthread, 1608
usb_add_device, 1600
usb_rm_device, 1601
usbadd, 1602
usbclaim, 1604
usbkill, 1605
usbmain, 1605
usbthread, 1609
usbunclaim, 1607
usb_rm_device
usb_linux.c, 1601
usb_tryreset
usb.c, 1539
usb.h, 1579
UsbId, 1213
guid, 1218
guidString, 1215
hwModified, 1218
hwModifiedString, 1216
modified, 1218
modifiedString, 1216, 1217
newGuid, 1217
newModified, 1217
UsbId, 1215
UsbId, 1215
usbModel
Kb, 310
usbSerial
Kb, 311
usbadd
usb_linux.c, 1602
usbclaim
usb_linux.c, 1604
usbdevice, 1516
usbid, 1506
usbinput, 1515
usbkill
usb.h, 1580
usb_linux.c, 1605
usbmain
usb.h, 1581
usb_linux.c, 1605
usbmode, 1512
usbmutex
usb.c, 1541
usbprofile, 1513
usbrecv
usb.h, 1555
usbsend
usb.h, 1555
usbthread
usb_linux.c, 1609
usbunclaim
usb_linux.c, 1607
useopacity
ckb-random/main.c, 1438
ushort
includes.h, 1303
usize
QuaZipFile, 904
utf16to8
profile.c, 1483
utf8to16
profile.c, 1484

V_CORSAIR
usb.h, 1555
V_CORSAIR_STR
usb.h, 1555
VERSIONMADEBY
zip.c, 1906

value
CkbSettings, 137
KeyAction, 632
QuaAdler32, 812
QuaChecksum32, 814
QuaCrc32, 818
vendor_str
usb.c, 1540
usb.h, 1582
verifyFw
fwupgradedialog.cpp, 1685
version
AnimScript, 82
versionCreated
QuaZipFileInfo, 909
QuaZipFileInfo64, 916
versionForBoard
KbFirmware, 423
versionLabel
Ui::SettingsWidget, 1035
Ui_SettingsWidget, 1212
versionNeeded
QuaZipFileInfo, 909
QuaZipFileInfo64, 916
versionUpdated
KbManager, 488
verticalLayout
Ui::KbAnimWidget, 347
Ui::KbBindWidget, 402
Ui::KbLightWidget, 463
Ui::MPerfWidget, 778
Ui::RebindWidget, 969
Ui_KbAnimWidget, 1090
Ui_KbBindWidget, 1097
Ui_KbLightWidget, 1105
Ui_MPerfWidget, 1167
Ui_RebindWidget, 1199
verticalLayout_2
Ui::RebindWidget, 969
Ui_RebindWidget, 1199
verticalSpacer
Ui::AnimSettingDialog, 98
Ui::ExtraSettingsWidget, 173
Ui::GradientDialog, 214
Ui::KbAnimWidget, 347
Ui::KbBindWidget, 402
Ui::KbWidget, 601
Ui::KPerfWidget, 704
Ui::LayoutDialog, 722
Ui::MPerfWidget, 778
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_AnimSettingDialog, 1056
Ui_ExtraSettingsWidget, 1067
Ui_GradientDialog, 1081
Ui_KbAnimWidget, 1090
Ui_KbBindWidget, 1097
Ui_KbWidget, 1118
Ui_KPerfWidget, 1138
Ui_LayoutDialog, 1143
Ui_MPerfWidget, 1167
Ui_RebindWidget, 1199
Ui_SettingsWidget, 1212
verticalSpacer_10
Ui::ExtraSettingsWidget, 174
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_ExtraSettingsWidget, 1067
Ui_RebindWidget, 1199
Ui_SettingsWidget, 1212
verticalSpacer_11
Ui::ExtraSettingsWidget, 174
Ui::RebindWidget, 970
Ui_ExtraSettingsWidget, 1067
Ui_RebindWidget, 1199
verticalSpacer_2
Ui::AnimAddDialog, 48
Ui::AnimSettingDialog, 98
Ui::ExtraSettingsWidget, 174
Ui::GradientDialog, 214
Ui::KbAnimWidget, 347
Ui::KbLightWidget, 463
Ui::KPerfWidget, 704
Ui::LayoutDialog, 722
Ui::MPerfWidget, 778
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_AnimAddDialog, 1043
Ui_AnimSettingDialog, 1056
Ui_ExtraSettingsWidget, 1067
Ui_GradientDialog, 1082
Ui_KbAnimWidget, 1090
Ui_KbLightWidget, 1105
Ui_KPerfWidget, 1138
Ui_LayoutDialog, 1143
Ui_MPerfWidget, 1167
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1212
verticalSpacer_3
Ui::AnimAddDialog, 48
Ui::AnimSettingDialog, 99
Ui::ExtraSettingsWidget, 174
Ui::KbAnimWidget, 347
Ui::KbLightWidget, 463
Ui::KPerfWidget, 704
Ui::LayoutDialog, 722
Ui::MPerfWidget, 778
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_AnimAddDialog, 1043
Ui_AnimSettingDialog, 1056
Ui_ExtraSettingsWidget, 1067
Ui_KbAnimWidget, 1091
Ui_KbLightWidget, 1105
Ui_KPerfWidget, 1138
Ui_LayoutDialog, 1143

Ui_MPerfWidget, 1168
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1212
verticalSpacer_4
Ui::ExtraSettingsWidget, 174
Ui::KPerfWidget, 704
Ui::LayoutDialog, 722
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_ExtraSettingsWidget, 1067
Ui_KPerfWidget, 1138
Ui_LayoutDialog, 1143
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1212
verticalSpacer_5
Ui::ExtraSettingsWidget, 174
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_ExtraSettingsWidget, 1068
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1212
verticalSpacer_6
Ui::ExtraSettingsWidget, 174
Ui::RebindWidget, 970
Ui::SettingsWidget, 1035
Ui_ExtraSettingsWidget, 1068
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1212
verticalSpacer_7
Ui::ExtraSettingsWidget, 174
Ui::RebindWidget, 970
Ui::SettingsWidget, 1036
Ui_ExtraSettingsWidget, 1068
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1212
verticalSpacer_8
Ui::ExtraSettingsWidget, 174
Ui::RebindWidget, 971
Ui::SettingsWidget, 1036
Ui_ExtraSettingsWidget, 1068
Ui_RebindWidget, 1200
Ui_SettingsWidget, 1213
verticalSpacer_9
Ui::ExtraSettingsWidget, 175
Ui_ExtraSettingsWidget, 1068
vtable_keyboard
device_vtable.c, 1257
structures.h, 1522
vtable_keyboard_nonrgb
device_vtable.c, 1257
structures.h, 1522
vtable_mouse
device_vtable.c, 1257
structures.h, 1522

wheelBox
Ui::RebindWidget, 971
Ui_RebindWidget, 1200
wheelButton
Ui::RebindWidget, 971
Ui_RebindWidget, 1200
wheelKeys
RebindWidget, 1007
widget
Ui::GradientDialog, 214
Ui::KbBindWidget, 402
Ui::KbLightWidget, 464
Ui_GradientDialog, 1082
Ui_KbBindWidget, 1097
Ui_KbLightWidget, 1105
width
ckb-wave/main.c, 1452
Key, 605
KeyMap, 660
winBox
Ui::SettingsWidget, 1036
Ui_SettingsWidget, 1213
winLabel
Ui::SettingsWidget, 1036
Ui_SettingsWidget, 1213
winLock
KbBind, 394, 395
Write_EndOfCentralDirectoryRecord
zip.c, 1914
Write_GlobalComment
zip.c, 1915
Write_LocalFileHeader
zip.c, 1916
Write_Zip64EndOfCentralDirectoryLocator
zip.c, 1918
Write_Zip64EndOfCentralDirectoryRecord
zip.c, 1919
writeCache
CkbSettings, 139
writeData
QuaGzipFile, 824
QuaZIODevice, 832
QuaZipFile, 905
writePos
QuaZipFilePrivate, 921
writeProfileHeader
Kb, 306

x
ckb-pinwheel/main.c, 1426
Key, 605
x_offset
ckb-pinwheel/main.c, 1427
ckb-ripple/main.c, 1444
xBox0
Ui::MPerfWidget, 778
Ui_MPerfWidget, 1168
xBox1
Ui::MPerfWidget, 778
Ui_MPerfWidget, 1168
xBox2
Ui::MPerfWidget, 778
Ui_MPerfWidget, 1168

xBox3
Ui::MPerfWidget, 778
Ui_MPerfWidget, 1168
xBox4
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1168
xBox5
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1168
xSlider0
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1168
xSlider1
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1168
xSlider2
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1169
xSlider3
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1169
xSlider4
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1169
xSlider5
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1169
xyBox
Ui::MPerfWidget, 779
Ui_MPerfWidget, 1169
xyLinkPath
mperfwidget.cpp, 1774

y
ckb-pinwheel/main.c, 1427
Key, 605
YES
KbFirmware, 416
y_offset
ckb-pinwheel/main.c, 1427
ckb-ripple/main.c, 1444
yBox0
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1169
yBox1
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1169
yBox2
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1169
yBox3
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1169
yBox4
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1170
yBox5
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1170
ySlider0
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1170
ySlider1
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1170
ySlider2
Ui::MPerfWidget, 780
Ui_MPerfWidget, 1170
ySlider3
Ui::MPerfWidget, 781
Ui_MPerfWidget, 1170
ySlider4
Ui::MPerfWidget, 781
Ui_MPerfWidget, 1170
ySlider5
Ui::MPerfWidget, 781
Ui_MPerfWidget, 1170
year
AnimScript, 82

Z_BUFSIZE
zip.c, 1906
Z_BZIP2ED
unzip.h, 1887
zip.h, 1959
Z_MAXFILENAMEINZIP
zip.c, 1906
z_crc_t
unzip.c, 1834
zip.c, 1907
ZCALLBACK
ioapi.h, 1784
ZCLOSE64
ioapi.h, 1784
ZERROR64
ioapi.h, 1784
ZFAKECLOSE64
ioapi.h, 1784
ZIP64ENDHEADERMAGIC
zip.c, 1906
ZIP64ENDLOCHEADERMAGIC
zip.c, 1907
ZIP_AUTO_CLOSE
zip.h, 1959
ZIP_BADZIPFILE
zip.h, 1959
ZIP_DEFAULT_FLAGS
zip.h, 1959
ZIP_EOF
zip.h, 1959
ZIP_ERRNO
zip.h, 1959
ZIP_INTERNALERROR
zip.h, 1959
ZIP_OK
zip.h, 1959
ZIP_PARAMERROR
zip.h, 1960

ZIP_SEQUENTIAL
zip.h, 1960
ZOPEN64
ioapi.h, 1785
ZPOS64_T
ioapi.h, 1786
ZREAD64
ioapi.h, 1785
ZSEEK64
ioapi.h, 1785
ZTELL64
ioapi.h, 1786
ZWRITE64
ioapi.h, 1786
zdecode
crypt.h, 1778
zencode
crypt.h, 1778
zip
QuaZipFilePrivate, 921
zip.c
_Z_OF, 1907
ALLOC, 1904
add_data_in_datablock, 1907
allocate_new_datablock, 1908
BUFREADCOMMENT, 1905
CENTRALHEADERMAGIC, 1905
DESCRIPTORHEADERMAGIC, 1905
ENDHEADERMAGIC, 1905
free_datablock, 1909
free_linkedlist, 1909
init_linkedlist, 1910
LOCALHEADERMAGIC, 1905
linkedlist_data, 1907
linkedlist_datablock_internal, 1907
LoadCentralDirectoryRecord, 1911
local, 1905
MAKEULONG64, 1905
SEEK_CUR, 1906
SEEK_END, 1906
SEEK_SET, 1906
SIZECENTRALHEADER, 1906
SIZEDATA_INDATABLOCK, 1906
TRYFREE, 1906
VERSIONMADEBY, 1906
Write_EndOfCentralDirectoryRecord, 1914
Write_GlobalComment, 1915
Write_LocalFileHeader, 1916
Write_Zip64EndOfCentralDirectoryLocator, 1918
Write_Zip64EndOfCentralDirectoryRecord, 1919
Z_BUFSIZE, 1906
Z_MAXFILENAMEINZIP, 1906
z_crc_t, 1907
ZIP64ENDHEADERMAGIC, 1906
ZIP64ENDLOCHEADERMAGIC, 1907
zip64FlushWriteBuffer, 1920
zip64local_SearchCentralDir, 1926
zip64local_SearchCentralDir64, 1927
zip64local_TmzDateToDosDate, 1929
zip64local_getByte, 1921
zip64local_getLong, 1922
zip64local_getLong64, 1923
zip64local_getShort, 1924
zip64local_putValue, 1925
zip64local_putValue_inmemory, 1926
zip_copyright, 1954
zipClearFlags, 1930
zipClose, 1930
zipCloseFileInZip, 1932
zipCloseFileInZipRaw, 1933
zipCloseFileInZipRaw64, 1933
zipOpen, 1938
zipOpen2, 1938
zipOpen2_64, 1939
zipOpen3, 1940
zipOpen64, 1942
zipOpenNewFileInZip, 1943
zipOpenNewFileInZip2, 1943
zipOpenNewFileInZip2_64, 1944
zipOpenNewFileInZip3, 1944
zipOpenNewFileInZip3_64, 1945
zipOpenNewFileInZip4, 1945
zipOpenNewFileInZip4_64, 1946
zipOpenNewFileInZip64, 1950
zipRemoveExtraInfoBlock, 1950
zipSetFlags, 1951
zipWriteInFileInZip, 1952
zip.h
_Z_OF, 1960, 1961
DEF_MEM_LEVEL, 1959
tm_zip, 1960
Z_BZIP2ED, 1959
ZIP_AUTO_CLOSE, 1959
ZIP_BADZIPFILE, 1959
ZIP_DEFAULT_FLAGS, 1959
ZIP_EOF, 1959
ZIP_ERRNO, 1959
ZIP_INTERNALERROR, 1959
ZIP_OK, 1959
ZIP_PARAMERROR, 1960
ZIP_SEQUENTIAL, 1960
zipClearFlags, 1961
zipFile, 1960
zipOpen3, 1962
zipSetFlags, 1964
zipcharpc, 1960
zip64
QuaZipPrivate, 939
zip64_internal, 1904
zip64FlushWriteBuffer
zip.c, 1920
zip64local_SearchCentralDir
zip.c, 1926
zip64local_SearchCentralDir64
zip.c, 1927
zip64local_TmzDateToDosDate

zip.c, 1929
zip64local_getByte
zip.c, 1921
zip64local_getLong
zip.c, 1922
zip64local_getLong64
zip.c, 1923
zip64local_getShort
zip.c, 1924
zip64local_putValue
zip.c, 1925
zip64local_putValue_inmemory
zip.c, 1926
zip_fileinfo, 1957
zip_copyright
zip.c, 1954
zipClearFlags
zip.c, 1930
zip.h, 1961
zipClose
zip.c, 1930
zipCloseFileInZip
zip.c, 1932
zipCloseFileInZipRaw
zip.c, 1933
zipCloseFileInZipRaw64
zip.c, 1933
zipError
QuaZipFilePrivate, 921
QuaZipPrivate, 939
zipFile
zip.h, 1960
zipName
QuaZipPrivate, 939
zipOpen
zip.c, 1938
zipOpen2
zip.c, 1938
zipOpen2_64
zip.c, 1939
zipOpen3
zip.c, 1940
zip.h, 1962
zipOpen64
zip.c, 1942
zipOpenNewFileInZip
zip.c, 1943
zipOpenNewFileInZip2
zip.c, 1943
zipOpenNewFileInZip2_64
zip.c, 1944
zipOpenNewFileInZip3
zip.c, 1944
zipOpenNewFileInZip3_64
zip.c, 1945
zipOpenNewFileInZip4
zip.c, 1945
zipOpenNewFileInZip4_64
zip.c, 1946
zipOpenNewFileInZip64
zip.c, 1950
zipRemoveExtraInfoBlock
zip.c, 1950
zipSetFlags
zip.c, 1951
zip.h, 1964
zipWriteInFileInZip
zip.c, 1952
zipcharpc
zip.h, 1960
zlib_filefunc64_32_def_s, 1783
zlib_filefunc64_def_s, 1782
zlib_filefunc_def_s, 1782
zlib_filefunc64_32_def
ioapi.h, 1786
zlib_filefunc64_def
ioapi.h, 1786
zlib_filefunc_def
ioapi.h, 1786