ckb-next

0.2.7

Generated by Doxygen 1.8.6

Thu Apr 27 2017 21:14:35

Contents

1 Todo List **1**

2 Deprecated List **3**

3 Namespace Index **5**

3.1 Namespace List 5

4 Hierarchical Index **7**

4.1 Class Hierarchy 7

5 Data Structure Index **11**

5.1 Data Structures 11

6 File Index **13**

6.1 File List 13

7 Namespace Documentation **17**

7.1 Ui Namespace Reference 17

8 Data Structure Documentation **19**

8.1 AnimAddDialog Class Reference 19

8.1.1 Detailed Description 20

8.1.2 Constructor & Destructor Documentation 21

8.1.2.1 AnimAddDialog 21

8.1.2.2 ∼AnimAddDialog 21

8.1.3 Member Function Documentation 21

8.1.3.1 chosenPreset 21

8.1.3.2 chosenScript 22

8.1.3.3 on_animBox_activated 22

8.1.3.4 on_presetBox_activated 23

8.1.3.5 on_previewBox_clicked 24

8.1.4 Field Documentation 24

8.1.4.1 _keys 24

8.1.4.2 _light 24

8.1.4.3 scripts . . . 24
8.1.4.4 showPreview . . . 25
8.1.4.5 ui . . . 25
8.2 AnimScript Class Reference . . . 25
8.2.1 Detailed Description . . . 29
8.2.2 Data Structure Documentation . . . 29
8.2.2.1 struct AnimScript::Param . . . 29
8.2.3 Member Typedef Documentation . . . 30
8.2.3.1 PresetValue . . . 30
8.2.4 Constructor & Destructor Documentation . . . 30
8.2.4.1 ∼AnimScript . . . 30
8.2.4.2 AnimScript . . . 30
8.2.4.3 AnimScript . . . 31
8.2.5 Member Function Documentation . . . 31
8.2.5.1 advance . . . 31
8.2.5.2 author . . . 31
8.2.5.3 begin . . . 32
8.2.5.4 colors . . . 33
8.2.5.5 copy . . . 33
8.2.5.6 copyright . . . 34
8.2.5.7 count . . . 35
8.2.5.8 description . . . 35
8.2.5.9 end . . . 35
8.2.5.10 frame . . . 36
8.2.5.11 guid . . . 37
8.2.5.12 hasFrame . . . 37
8.2.5.13 hasKeypress . . . 38
8.2.5.14 hasParam . . . 38
8.2.5.15 init . . . 39
8.2.5.16 keypress . . . 40
8.2.5.17 license . . . 41
8.2.5.18 list . . . 42
8.2.5.19 load . . . 42
8.2.5.20 name . . . 45
8.2.5.21 param . . . 45
8.2.5.22 parameters . . . 46
8.2.5.23 paramIterator . . . 47
8.2.5.24 path . . . 47
8.2.5.25 preset . . . 48
8.2.5.26 presets . . . 48

8.2.5.27 printParams . . . 49
8.2.5.28 readProcess . . . 49
8.2.5.29 retrigger . . . 50
8.2.5.30 scan . . . 51
8.2.5.31 setDuration . . . 52
8.2.5.32 stop . . . 52
8.2.5.33 version . . . 53
8.2.5.34 year . . . 53
8.2.6 Field Documentation . . . 54
8.2.6.1 _colorBuffer . . . 54
8.2.6.2 _colors . . . 54
8.2.6.3 _info . . . 54
8.2.6.4 _keys . . . 54
8.2.6.5 _map . . . 54
8.2.6.6 _paramValues . . . 54
8.2.6.7 _path . . . 55
8.2.6.8 _presets . . . 55
8.2.6.9 _presetValues . . . 55
8.2.6.10 durationMsec . . . 55
8.2.6.11 firstFrame . . . 55
8.2.6.12 inFrame . . . 55
8.2.6.13 initialized . . . 55
8.2.6.14 KP_NAME . . . 55
8.2.6.15 KP_NONE . . . 55
8.2.6.16 KP_POSITION . . . 56
8.2.6.17 lastFrame . . . 56
8.2.6.18 minX . . . 56
8.2.6.19 minY . . . 56
8.2.6.20 process . . . 56
8.2.6.21 readAnyFrame . . . 56
8.2.6.22 readFrame . . . 56
8.2.6.23 repeatMsec . . . 56
8.2.6.24 scripts . . . 56
8.2.6.25 stopped . . . 57
8.3 AnimSettingDialog Class Reference . . . 57
8.3.1 Detailed Description . . . 59
8.3.2 Constructor & Destructor Documentation . . . 59
8.3.2.1 AnimSettingDialog . . . 59
8.3.2.2 ~AnimSettingDialog . . . 64
8.3.3 Member Function Documentation . . . 64

8.3.3.1 angleDialChanged . . . 65
8.3.3.2 angleSpinnerChanged . . . 65
8.3.3.3 hLine . . . 66
8.3.3.4 name . . . 67
8.3.3.5 newDuration . . . 67
8.3.3.6 on_delayBox_valueChanged . . . 68
8.3.3.7 on_kpDelayBox_valueChanged . . . 69
8.3.3.8 on_kpReleaseBox_stateChanged . . . 69
8.3.3.9 on_kpRepeatBox_valueChanged . . . 70
8.3.3.10 on_repeatBox_valueChanged . . . 71
8.3.3.11 updateParam . . . 71
8.3.3.12 updateStops . . . 74
8.3.4 Field Documentation . . . 75
8.3.4.1 _anim . . . 75
8.3.4.2 angleDialMapper . . . 75
8.3.4.3 angleSpinnerMapper . . . 75
8.3.4.4 angleSpinners . . . 76
8.3.4.5 hasRepeat . . . 76
8.3.4.6 kpStopCheck . . . 76
8.3.4.7 lastDuration . . . 76
8.3.4.8 settingWidgets . . . 76
8.3.4.9 stopCheck . . . 76
8.3.4.10 ui . . . 76
8.3.4.11 updateMapper . . . 76
8.4 AutoRun Class Reference . . . 77
8.4.1 Detailed Description . . . 77
8.4.2 Member Function Documentation . . . 77
8.4.2.1 available . . . 77
8.4.2.2 disable . . . 78
8.4.2.3 enable . . . 78
8.4.2.4 isEnabled . . . 79
8.4.2.5 once . . . 80
8.5 CkbSettings Class Reference . . . 81
8.5.1 Detailed Description . . . 83
8.5.2 Constructor & Destructor Documentation . . . 83
8.5.2.1 CkbSettings . . . 83
8.5.2.2 CkbSettings . . . 83
8.5.2.3 CkbSettings . . . 84
8.5.2.4 ~CkbSettings . . . 84
8.5.3 Member Function Documentation . . . 84

8.5.3.1 beginGroup . . . 84
8.5.3.2 childGroups . . . 84
8.5.3.3 childKeys . . . 85
8.5.3.4 cleanUp . . . 86
8.5.3.5 contains . . . 87
8.5.3.6 containsGroup . . . 87
8.5.3.7 endGroup . . . 88
8.5.3.8 get . . . 88
8.5.3.9 isBusy . . . 89
8.5.3.10 pwd . . . 90
8.5.3.11 pwd . . . 91
8.5.3.12 remove . . . 92
8.5.3.13 set . . . 92
8.5.3.14 setValue . . . 94
8.5.3.15 value . . . 94
8.5.4 Field Documentation . . . 95
8.5.4.1 backing . . . 95
8.5.4.2 groups . . . 95
8.5.4.3 removeCache . . . 96
8.5.4.4 writeCache . . . 96
8.6 CkbSettingsWriter Class Reference . . . 96
8.6.1 Detailed Description . . . 97
8.6.2 Constructor & Destructor Documentation . . . 97
8.6.2.1 CkbSettingsWriter . . . 97
8.6.2.2 ~CkbSettingsWriter . . . 98
8.6.3 Member Function Documentation . . . 98
8.6.3.1 run . . . 98
8.6.4 Field Documentation . . . 98
8.6.4.1 _backing . . . 98
8.6.4.2 _removals . . . 98
8.6.4.3 _updates . . . 98
8.7 ColorButton Class Reference . . . 98
8.7.1 Detailed Description . . . 101
8.7.2 Constructor & Destructor Documentation . . . 101
8.7.2.1 ColorButton . . . 101
8.7.3 Member Function Documentation . . . 101
8.7.3.1 allowAlpha . . . 101
8.7.3.2 allowAlpha . . . 102
8.7.3.3 bigIcons . . . 102
8.7.3.4 bigIcons . . . 103

8.7.3.5 color . . . 103
8.7.3.6 color . . . 104
8.7.3.7 colorChanged . . . 105
8.7.3.8 pickColor . . . 105
8.7.3.9 setLabel . . . 105
8.7.3.10 setLabel . . . 106
8.7.3.11 updateImage . . . 106
8.7.4 Field Documentation . . . 107
8.7.4.1 _alpha . . . 107
8.7.4.2 _bigIcons . . . 107
8.7.4.3 _color . . . 107
8.7.4.4 _setLabel . . . 107
8.8 ColorMap Class Reference . . . 107
8.8.1 Detailed Description . . . 109
8.8.2 Constructor & Destructor Documentation . . . 109
8.8.2.1 ColorMap . . . 109
8.8.2.2 ~ColorMap . . . 109
8.8.2.3 ColorMap . . . 109
8.8.3 Member Function Documentation . . . 110
8.8.3.1 alloc . . . 110
8.8.3.2 clear . . . 110
8.8.3.3 colorForName . . . 111
8.8.3.4 colorForName . . . 112
8.8.3.5 colors . . . 112
8.8.3.6 colors . . . 113
8.8.3.7 count . . . 113
8.8.3.8 deAlloc . . . 113
8.8.3.9 init . . . 114
8.8.3.10 keyNames . . . 115
8.8.3.11 operator= . . . 115
8.8.4 Field Documentation . . . 116
8.8.4.1 _colors . . . 116
8.8.4.2 _count . . . 116
8.8.4.3 _keyNames . . . 116
8.8.4.4 _mapCount . . . 116
8.9 ExtraSettingsWidget Class Reference . . . 116
8.9.1 Detailed Description . . . 119
8.9.2 Constructor & Destructor Documentation . . . 119
8.9.2.1 ExtraSettingsWidget . . . 119
8.9.2.2 ~ExtraSettingsWidget . . . 120

8.9.3 Member Function Documentation . . . 120
8.9.3.1 on_animScanButton_clicked . . . 120
8.9.3.2 on_brightnessBox_clicked . . . 121
8.9.3.3 on_delayBox_clicked . . . 121
8.9.3.4 on_ditherBox_clicked . . . 122
8.9.3.5 on_fpsBox_valueChanged . . . 122
8.9.3.6 on_mAccelBox_clicked . . . 122
8.9.3.7 on_sAccelBox_clicked . . . 123
8.9.3.8 on_sSpeedBox_valueChanged . . . 123
8.9.3.9 on_trayBox_clicked . . . 124
8.9.3.10 pollUpdates . . . 124
8.9.4 Field Documentation . . . 125
8.9.4.1 ui . . . 125
8.10 KbFirmware::FW Struct Reference . . . 125
8.10.1 Detailed Description . . . 126
8.10.2 Constructor & Destructor Documentation . . . 126
8.10.2.1 FW . . . 126
8.10.3 Field Documentation . . . 126
8.10.3.1 ckbVersion . . . 126
8.10.3.2 fileName . . . 126
8.10.3.3 fwVersion . . . 126
8.10.3.4 hash . . . 126
8.10.3.5 url . . . 126
8.11 FwUpgradeDialog Class Reference . . . 126
8.11.1 Detailed Description . . . 128
8.11.2 Constructor & Destructor Documentation . . . 129
8.11.2.1 FwUpgradeDialog . . . 129
8.11.2.2 ~FwUpgradeDialog . . . 129
8.11.3 Member Function Documentation . . . 129
8.11.3.1 cleanBlob . . . 130
8.11.3.2 closeEvent . . . 130
8.11.3.3 exec . . . 130
8.11.3.4 fwUpdateFinished . . . 131
8.11.3.5 fwUpdateProgress . . . 132
8.11.3.6 on_actionButton_clicked . . . 133
8.11.3.7 on_cancelButton_clicked . . . 133
8.11.3.8 removeDev . . . 133
8.11.3.9 saveBlob . . . 134
8.11.4 Field Documentation . . . 134
8.11.4.1 blob . . . 134

8.11.4.2 evLoop . . . 135
8.11.4.3 exitSuccess . . . 135
8.11.4.4 kb . . . 135
8.11.4.5 savePath . . . 135
8.11.4.6 ui . . . 135
8.12 GradientButton Class Reference . . . 135
8.12.1 Detailed Description . . . 137
8.12.2 Constructor & Destructor Documentation . . . 137
8.12.2.1 GradientButton . . . 137
8.12.3 Member Function Documentation . . . 138
8.12.3.1 fromString . . . 138
8.12.3.2 gradientChanged . . . 139
8.12.3.3 pickGradient . . . 139
8.12.3.4 toString . . . 140
8.12.3.5 updateImage . . . 141
8.12.4 Field Documentation . . . 141
8.12.4.1 _alpha . . . 141
8.12.4.2 _stops . . . 141
8.13 GradientDialog Class Reference . . . 141
8.13.1 Detailed Description . . . 144
8.13.2 Constructor & Destructor Documentation . . . 144
8.13.2.1 GradientDialog . . . 144
8.13.2.2 ~GradientDialog . . . 145
8.13.3 Member Function Documentation . . . 145
8.13.3.1 addPreset . . . 146
8.13.3.2 colorChanged . . . 146
8.13.3.3 currentChanged . . . 147
8.13.3.4 getGradient . . . 148
8.13.3.5 makeIcon . . . 148
8.13.3.6 on_presetDelete_clicked . . . 149
8.13.3.7 on_presetList_currentItemChanged . . . 149
8.13.3.8 on_presetName_textEdited . . . 150
8.13.3.9 on_presetSave_clicked . . . 150
8.13.3.10 on_stopOpacity_valueChanged . . . 151
8.13.3.11 on_stopPos_valueChanged . . . 151
8.13.3.12 setPreset . . . 152
8.13.3.13 updatePresets . . . 153
8.13.4 Field Documentation . . . 153
8.13.4.1 currentPreset . . . 153
8.13.4.2 presets . . . 154

8.13.4.3 ui . . . 154
8.14 GradientDialogWidget Class Reference . . . 154
8.14.1 Detailed Description . . . 157
8.14.2 Constructor & Destructor Documentation . . . 157
8.14.2.1 GradientDialogWidget . . . 157
8.14.3 Member Function Documentation . . . 157
8.14.3.1 colorAt . . . 157
8.14.3.2 current . . . 158
8.14.3.3 currentChanged . . . 158
8.14.3.4 fillRect . . . 158
8.14.3.5 makeStops . . . 159
8.14.3.6 mouseMoveEvent . . . 160
8.14.3.7 mousePressEvent . . . 161
8.14.3.8 mouseReleaseEvent . . . 162
8.14.3.9 moveCurrent . . . 162
8.14.3.10 paintEvent . . . 163
8.14.3.11 selectedStop . . . 164
8.14.3.12 selectionColors . . . 165
8.14.3.13 setCurrentColor . . . 165
8.14.3.14 setStops . . . 166
8.14.3.15 stopCount . . . 166
8.14.3.16 stopRect . . . 167
8.14.3.17 stops . . . 167
8.14.4 Field Documentation . . . 168
8.14.4.1 _colors . . . 168
8.14.4.2 _current . . . 168
8.14.4.3 _stops . . . 168
8.14.4.4 selected . . . 168
8.14.4.5 selectedOffset . . . 168
8.14.4.6 selectedPos . . . 168
8.15 JlCompress Class Reference . . . 169
8.15.1 Detailed Description . . . 170
8.15.2 Member Function Documentation . . . 170
8.15.2.1 compressDir . . . 170
8.15.2.2 compressFile . . . 171
8.15.2.3 compressFile . . . 172
8.15.2.4 compressFiles . . . 173
8.15.2.5 compressSubDir . . . 175
8.15.2.6 extractDir . . . 177
8.15.2.7 extractFile . . . 179

8.15.2.8 extractFile . . . 181
8.15.2.9 extractFiles . . . 182
8.15.2.10 getFileList . . . 183
8.15.2.11 removeFile . . . 185
8.16 Kb Class Reference . . . 186
8.16.1 Detailed Description . . . 191
8.16.2 Constructor & Destructor Documentation . . . 191
8.16.2.1 ~Kb . . . 191
8.16.2.2 Kb . . . 192
8.16.3 Member Function Documentation . . . 194
8.16.3.1 appendProfile . . . 194
8.16.3.2 autoSave . . . 194
8.16.3.3 currentBind . . . 194
8.16.3.4 currentLight . . . 195
8.16.3.5 currentMode . . . 196
8.16.3.6 currentPerf . . . 196
8.16.3.7 currentProfile . . . 197
8.16.3.8 deleteHw . . . 197
8.16.3.9 deletePrevious . . . 198
8.16.3.10 dither . . . 198
8.16.3.11 dither . . . 199
8.16.3.12 find . . . 199
8.16.3.13 frameRate . . . 200
8.16.3.14 frameRate . . . 200
8.16.3.15 frameUpdate . . . 201
8.16.3.16 fwUpdate . . . 202
8.16.3.17 fwUpdateFinished . . . 203
8.16.3.18 fwUpdateProgress . . . 203
8.16.3.19 getKeyMap . . . 203
8.16.3.20 getMacroNumber . . . 204
8.16.3.21 getMacroPath . . . 204
8.16.3.22 hwProfile . . . 205
8.16.3.23 hwProfile . . . 205
8.16.3.24 hwSave . . . 206
8.16.3.25 indexOf . . . 207
8.16.3.26 infoUpdated . . . 208
8.16.3.27 isKeyboard . . . 208
8.16.3.28 isMouse . . . 209
8.16.3.29 isOpen . . . 209
8.16.3.30 layout . . . 210

8.16.3.31 layout . . . 210
8.16.3.32 load . . . 211
8.16.3.33 macroDelay . . . 212
8.16.3.34 macroDelay . . . 212
8.16.3.35 matches . . . 212
8.16.3.36 modeChanged . . . 213
8.16.3.37 model . . . 213
8.16.3.38 modeRenamed . . . 213
8.16.3.39 mouseAccel . . . 214
8.16.3.40 mouseAccel . . . 214
8.16.3.41 needsSave . . . 214
8.16.3.42 newMode . . . 215
8.16.3.43 newMode . . . 216
8.16.3.44 newProfile . . . 216
8.16.3.45 newProfile . . . 217
8.16.3.46 profileAdded . . . 218
8.16.3.47 profileChanged . . . 218
8.16.3.48 profileRenamed . . . 218
8.16.3.49 profiles . . . 218
8.16.3.50 profiles . . . 219
8.16.3.51 readNotify . . . 219
8.16.3.52 run . . . 224
8.16.3.53 save . . . 225
8.16.3.54 scrollSpeed . . . 226
8.16.3.55 scrollSpeed . . . 226
8.16.3.56 setCurrentMode . . . 227
8.16.3.57 setCurrentMode . . . 227
8.16.3.58 setCurrentMode . . . 228
8.16.3.59 setCurrentProfile . . . 229
8.16.3.60 updateLayout . . . 230
8.16.3.61 writeProfileHeader . . . 231
8.16.4 Friends And Related Function Documentation . . . 232
8.16.4.1 KbManager . . . 232
8.16.5 Field Documentation . . . 232
8.16.5.1 _currentMode . . . 232
8.16.5.2 _currentProfile . . . 232
8.16.5.3 _delay . . . 232
8.16.5.4 _dither . . . 232
8.16.5.5 _frameRate . . . 232
8.16.5.6 _hwProfile . . . 233

8.16.5.7 _layout . . . 233
8.16.5.8 _model . . . 233
8.16.5.9 _mouseAccel . . . 233
8.16.5.10 _needsSave . . . 233
8.16.5.11 _profiles . . . 233
8.16.5.12 _scrollSpeed . . . 233
8.16.5.13 cmd . . . 233
8.16.5.14 cmdpath . . . 233
8.16.5.15 devpath . . . 234
8.16.5.16 features . . . 234
8.16.5.17 firmware . . . 234
8.16.5.18 fwUpdPath . . . 234
8.16.5.19 hwLoading . . . 234
8.16.5.20 HWMODE_MAX . . . 234
8.16.5.21 hwModeCount . . . 234
8.16.5.22 iState . . . 234
8.16.5.23 lastAutoSave . . . 234
8.16.5.24 macroNumber . . . 235
8.16.5.25 macroPath . . . 235
8.16.5.26 monochrome . . . 235
8.16.5.27 notifyNumber . . . 235
8.16.5.28 notifyPath . . . 235
8.16.5.29 pollrate . . . 235
8.16.5.30 prefsPath . . . 235
8.16.5.31 prevMode . . . 235
8.16.5.32 prevProfile . . . 235
8.16.5.33 usbModel . . . 236
8.16.5.34 usbSerial . . . 236
8.17 KbAnim Class Reference . . . 236
8.17.1 Detailed Description . . . 239
8.17.2 Member Enumeration Documentation . . . 239
8.17.2.1 Mode . . . 239
8.17.3 Constructor & Destructor Documentation . . . 240
8.17.3.1 KbAnim . . . 240
8.17.3.2 KbAnim . . . 241
8.17.3.3 KbAnim . . . 242
8.17.4 Member Function Documentation . . . 242
8.17.4.1 blend . . . 242
8.17.4.2 catchUp . . . 244
8.17.4.3 commitParams . . . 245

8.17.4.4 effectiveParams . . . 246
8.17.4.5 guid . . . 246
8.17.4.6 hasParameter . . . 247
8.17.4.7 isActive . . . 247
8.17.4.8 isRunning . . . 247
8.17.4.9 keypress . . . 248
8.17.4.10 keys . . . 249
8.17.4.11 keys . . . 250
8.17.4.12 map . . . 250
8.17.4.13 map . . . 251
8.17.4.14 mode . . . 251
8.17.4.15 mode . . . 252
8.17.4.16 name . . . 252
8.17.4.17 name . . . 253
8.17.4.18 needsSave . . . 253
8.17.4.19 newId . . . 253
8.17.4.20 opacity . . . 253
8.17.4.21 opacity . . . 254
8.17.4.22 parameter . . . 254
8.17.4.23 parameter . . . 255
8.17.4.24 reInit . . . 255
8.17.4.25 resetParams . . . 256
8.17.4.26 save . . . 257
8.17.4.27 script . . . 258
8.17.4.28 scriptName . . . 258
8.17.4.29 stop . . . 259
8.17.4.30 trigger . . . 260
8.17.4.31 updateParams . . . 261
8.17.5 Field Documentation . . . 262
8.17.5.1 _guid . . . 262
8.17.5.2 _isActive . . . 262
8.17.5.3 _isActiveKp . . . 262
8.17.5.4 _keys . . . 262
8.17.5.5 _map . . . 262
8.17.5.6 _mode . . . 262
8.17.5.7 _name . . . 262
8.17.5.8 _needsSave . . . 263
8.17.5.9 _opacity . . . 263
8.17.5.10 _parameters . . . 263
8.17.5.11 _script . . . 263

8.17.5.12 _scriptGuid . . . 263
8.17.5.13 _scriptName . . . 263
8.17.5.14 _tempParameters . . . 263
8.17.5.15 kpRepeatMsec . . . 263
8.17.5.16 kpRepeatTime . . . 263
8.17.5.17 kpStopTime . . . 264
8.17.5.18 repeatKey . . . 264
8.17.5.19 repeatMsec . . . 264
8.17.5.20 repeatTime . . . 264
8.17.5.21 stopTime . . . 264
8.18 KbAnimWidget Class Reference . . . 264
8.18.1 Detailed Description . . . 267
8.18.2 Constructor & Destructor Documentation . . . 267
8.18.2.1 KbAnimWidget . . . 267
8.18.2.2 ~KbAnimWidget . . . 268
8.18.3 Member Function Documentation . . . 268
8.18.3.1 addAnim . . . 268
8.18.3.2 animChanged . . . 269
8.18.3.3 clearSelection . . . 269
8.18.3.4 didUpdateSelection . . . 270
8.18.3.5 duplicateAnim . . . 270
8.18.3.6 on_animList_currentItemChanged . . . 271
8.18.3.7 on_animList_customContextMenuRequested . . . 272
8.18.3.8 on_animList_itemChanged . . . 273
8.18.3.9 on_blendBox_activated . . . 273
8.18.3.10 on_deleteButton_clicked . . . 274
8.18.3.11 on_keyButton_clicked . . . 275
8.18.3.12 on_nameBox_textEdited . . . 275
8.18.3.13 on_opacityBox_valueChanged . . . 276
8.18.3.14 on_propertyButton_clicked . . . 276
8.18.3.15 refreshList . . . 277
8.18.3.16 reorderAnims . . . 278
8.18.3.17 setCurrent . . . 279
8.18.3.18 setLight . . . 281
8.18.3.19 setSelectedKeys . . . 281
8.18.4 Field Documentation . . . 281
8.18.4.1 animations . . . 281
8.18.4.2 current . . . 282
8.18.4.3 light . . . 282
8.18.4.4 noReorder . . . 282

8.18.4.5 selectedKeys . . . 282
8.18.4.6 ui . . . 282
8.19 KbBind Class Reference . . . 282
8.19.1 Detailed Description . . . 286
8.19.2 Constructor & Destructor Documentation . . . 286
8.19.2.1 KbBind . . . 286
8.19.2.2 KbBind . . . 286
8.19.3 Member Function Documentation . . . 287
8.19.3.1 action . . . 287
8.19.3.2 bindAction . . . 288
8.19.3.3 defaultAction . . . 288
8.19.3.4 devParent . . . 289
8.19.3.5 didLoad . . . 290
8.19.3.6 friendlyActionName . . . 290
8.19.3.7 friendlyName . . . 290
8.19.3.8 getMacroNumber . . . 291
8.19.3.9 getMacroPath . . . 292
8.19.3.10 globalRemap . . . 292
8.19.3.11 handleNotificationChannel . . . 293
8.19.3.12 isISO . . . 294
8.19.3.13 isKeyboard . . . 294
8.19.3.14 isMouse . . . 295
8.19.3.15 keyEvent . . . 295
8.19.3.16 layoutChanged . . . 296
8.19.3.17 light . . . 297
8.19.3.18 load . . . 297
8.19.3.19 loadGlobalRemap . . . 299
8.19.3.20 map . . . 299
8.19.3.21 map . . . 300
8.19.3.22 modeParent . . . 300
8.19.3.23 needsSave . . . 301
8.19.3.24 noAction . . . 301
8.19.3.25 noAction . . . 302
8.19.3.26 perf . . . 302
8.19.3.27 resetAction . . . 303
8.19.3.28 resetAction . . . 304
8.19.3.29 save . . . 304
8.19.3.30 saveGlobalRemap . . . 305
8.19.3.31 setAction . . . 306
8.19.3.32 setAction . . . 307

8.19.3.33 setGlobalRemap . . . 307
8.19.3.34 setNeedsUpdate . . . 308
8.19.3.35 update . . . 308
8.19.3.36 updated . . . 310
8.19.3.37 winLock . . . 310
8.19.3.38 winLock . . . 311
8.19.4 Friends And Related Function Documentation . . . 311
8.19.4.1 KeyAction . . . 311
8.19.5 Field Documentation . . . 311
8.19.5.1 _bind . . . 311
8.19.5.2 _devParent . . . 311
8.19.5.3 _globalRemap . . . 311
8.19.5.4 _map . . . 311
8.19.5.5 _needsSave . . . 312
8.19.5.6 _needsUpdate . . . 312
8.19.5.7 _winLock . . . 312
8.19.5.8 globalRemapTime . . . 312
8.19.5.9 lastCmd . . . 312
8.19.5.10 lastGlobalRemapTime . . . 312
8.20 KbBindWidget Class Reference . . . 312
8.20.1 Detailed Description . . . 314
8.20.2 Constructor & Destructor Documentation . . . 315
8.20.2.1 KbBindWidget . . . 315
8.20.2.2 ~KbBindWidget . . . 315
8.20.3 Member Function Documentation . . . 315
8.20.3.1 newLayout . . . 315
8.20.3.2 newSelection . . . 316
8.20.3.3 on_copyButton_clicked . . . 316
8.20.3.4 on_resetButton_clicked . . . 317
8.20.3.5 setBind . . . 318
8.20.3.6 updateBind . . . 318
8.20.3.7 updateSelDisplay . . . 319
8.20.4 Field Documentation . . . 320
8.20.4.1 bind . . . 320
8.20.4.2 currentSelection . . . 320
8.20.4.3 profile . . . 320
8.20.4.4 ui . . . 320
8.21 KbFirmware Class Reference . . . 321
8.21.1 Detailed Description . . . 323
8.21.2 Member Enumeration Documentation . . . 323

8.21.2.1 anonymous enum . . . 323
8.21.3 Constructor & Destructor Documentation . . . 323
8.21.3.1 KbFirmware . . . 323
8.21.3.2 ~KbFirmware . . . 323
8.21.4 Member Function Documentation . . . 324
8.21.4.1 _checkUpdates . . . 324
8.21.4.2 _fileForBoard . . . 324
8.21.4.3 _latestForBoard . . . 325
8.21.4.4 checkUpdates . . . 326
8.21.4.5 dataForBoard . . . 327
8.21.4.6 downloaded . . . 327
8.21.4.7 downloadFinished . . . 328
8.21.4.8 hasDownloaded . . . 328
8.21.4.9 initManager . . . 329
8.21.4.10 processDownload . . . 329
8.21.4.11 versionForBoard . . . 331
8.21.5 Field Documentation . . . 331
8.21.5.1 fwTable . . . 331
8.21.5.2 fwTableHash . . . 331
8.21.5.3 hasGPG . . . 331
8.21.5.4 instance . . . 332
8.21.5.5 lastCheck . . . 332
8.21.5.6 lastFinished . . . 332
8.21.5.7 networkManager . . . 332
8.21.5.8 tableDownload . . . 332
8.22 KbLight Class Reference . . . 332
8.22.1 Detailed Description . . . 335
8.22.2 Member Typedef Documentation . . . 336
8.22.2.1 AnimList . . . 336
8.22.3 Constructor & Destructor Documentation . . . 336
8.22.3.1 KbLight . . . 336
8.22.3.2 KbLight . . . 336
8.22.3.3 ~KbLight . . . 337
8.22.4 Member Function Documentation . . . 337
8.22.4.1 addAnim . . . 337
8.22.4.2 animKeypress . . . 338
8.22.4.3 animList . . . 338
8.22.4.4 animList . . . 339
8.22.4.5 base . . . 339
8.22.4.6 close . . . 340

8.22.4.7 color . . . 341
8.22.4.8 color . . . 342
8.22.4.9 colorMap . . . 342
8.22.4.10 didLoad . . . 343
8.22.4.11 dimming . . . 343
8.22.4.12 dimming . . . 344
8.22.4.13 duplicateAnim . . . 344
8.22.4.14 findAnim . . . 345
8.22.4.15 findAnimIdx . . . 345
8.22.4.16 frameDisplayed . . . 346
8.22.4.17 frameUpdate . . . 346
8.22.4.18 isStarted . . . 348
8.22.4.19 load . . . 349
8.22.4.20 map . . . 350
8.22.4.21 map . . . 351
8.22.4.22 needsSave . . . 352
8.22.4.23 open . . . 352
8.22.4.24 previewAnim . . . 353
8.22.4.25 printRGB . . . 354
8.22.4.26 rebuildBaseMap . . . 355
8.22.4.27 resetIndicators . . . 356
8.22.4.28 restartAnimation . . . 356
8.22.4.29 save . . . 357
8.22.4.30 setIndicator . . . 358
8.22.4.31 shareDimming . . . 359
8.22.4.32 shareDimming . . . 359
8.22.4.33 stopPreview . . . 360
8.22.4.34 updated . . . 360
8.22.5 Field Documentation . . . 360
8.22.5.1 _animList . . . 360
8.22.5.2 _animMap . . . 361
8.22.5.3 _colorMap . . . 361
8.22.5.4 _dimming . . . 361
8.22.5.5 _indicatorList . . . 361
8.22.5.6 _indicatorMap . . . 361
8.22.5.7 _map . . . 361
8.22.5.8 _needsMapRefresh . . . 361
8.22.5.9 _needsSave . . . 361
8.22.5.10 _previewAnim . . . 361
8.22.5.11 _qColorMap . . . 362

8.22.5.12 _start . . . 362
8.22.5.13 lastFrameSignal . . . 362
8.22.5.14 MAX_DIM . . . 362
8.23 KbLightWidget Class Reference . . . 362
8.23.1 Detailed Description . . . 365
8.23.2 Constructor & Destructor Documentation . . . 365
8.23.2.1 KbLightWidget . . . 365
8.23.2.2 ~KbLightWidget . . . 365
8.23.3 Member Function Documentation . . . 366
8.23.3.1 changeAnim . . . 366
8.23.3.2 changeAnimKeys . . . 366
8.23.3.3 changeColor . . . 367
8.23.3.4 newSelection . . . 367
8.23.3.5 on_animButton_clicked . . . 368
8.23.3.6 on_bgButton_clicked . . . 369
8.23.3.7 on_brightnessBox_activated . . . 369
8.23.3.8 on_showAnimBox_clicked . . . 370
8.23.3.9 setLight . . . 370
8.23.3.10 setMonochrome . . . 371
8.23.3.11 toggleSidelight . . . 371
8.23.3.12 updateLight . . . 372
8.23.4 Field Documentation . . . 372
8.23.4.1 currentSelection . . . 373
8.23.4.2 light . . . 373
8.23.4.3 ui . . . 373
8.24 KbManager Class Reference . . . 373
8.24.1 Detailed Description . . . 376
8.24.2 Constructor & Destructor Documentation . . . 376
8.24.2.1 KbManager . . . 376
8.24.3 Member Function Documentation . . . 377
8.24.3.1 ckbDaemonVersion . . . 377
8.24.3.2 ckbDaemonVersionF . . . 377
8.24.3.3 ckbGuiVersion . . . 378
8.24.3.4 ckbGuiVersionF . . . 378
8.24.3.5 devices . . . 379
8.24.3.6 eventTimer . . . 379
8.24.3.7 fps . . . 379
8.24.3.8 init . . . 380
8.24.3.9 kbConnected . . . 381
8.24.3.10 kbDisconnected . . . 381

8.24.3.11 kbManager . . . 381
8.24.3.12 parseVersionString . . . 382
8.24.3.13 scanKeyboards . . . 382
8.24.3.14 scanTimer . . . 384
8.24.3.15 stop . . . 384
8.24.3.16 versionUpdated . . . 385
8.24.4 Field Documentation . . . 385
8.24.4.1 _daemonVersion . . . 385
8.24.4.2 _devices . . . 385
8.24.4.3 _eventTimer . . . 385
8.24.4.4 _guiVersion . . . 385
8.24.4.5 _kbManager . . . 385
8.24.4.6 _scanTimer . . . 385
8.25 KbMode Class Reference . . . 386
8.25.1 Detailed Description . . . 388
8.25.2 Constructor & Destructor Documentation . . . 388
8.25.2.1 KbMode . . . 388
8.25.2.2 KbMode . . . 389
8.25.2.3 KbMode . . . 389
8.25.2.4 ~KbMode . . . 390
8.25.3 Member Function Documentation . . . 390
8.25.3.1 bind . . . 390
8.25.3.2 doUpdate . . . 391
8.25.3.3 id . . . 391
8.25.3.4 id . . . 391
8.25.3.5 keyMap . . . 392
8.25.3.6 light . . . 393
8.25.3.7 name . . . 393
8.25.3.8 name . . . 394
8.25.3.9 needsSave . . . 394
8.25.3.10 newId . . . 395
8.25.3.11 perf . . . 395
8.25.3.12 save . . . 396
8.25.3.13 setNeedsSave . . . 397
8.25.3.14 setNeedsUpdate . . . 398
8.25.3.15 updated . . . 398
8.25.4 Field Documentation . . . 398
8.25.4.1 _bind . . . 399
8.25.4.2 _id . . . 399
8.25.4.3 _light . . . 399

8.25.4.4 _name . . . 399
8.25.4.5 _needsSave . . . 399
8.25.4.6 _perf . . . 399
8.26 KbPerf Class Reference . . . 399
8.26.1 Detailed Description . . . 403
8.26.2 Member Enumeration Documentation . . . 403
8.26.2.1 height . . . 403
8.26.2.2 i_hw . . . 404
8.26.2.3 indicator . . . 404
8.26.3 Constructor & Destructor Documentation . . . 404
8.26.3.1 KbPerf . . . 404
8.26.3.2 KbPerf . . . 405
8.26.4 Member Function Documentation . . . 406
8.26.4.1 _curDpi . . . 406
8.26.4.2 angleSnap . . . 406
8.26.4.3 angleSnap . . . 407
8.26.4.4 applyIndicators . . . 407
8.26.4.5 bind . . . 409
8.26.4.6 curDpi . . . 409
8.26.4.7 curDpi . . . 410
8.26.4.8 curDpi . . . 410
8.26.4.9 curDpiIdx . . . 411
8.26.4.10 curDpiIdx . . . 411
8.26.4.11 didLoad . . . 412
8.26.4.12 dpi . . . 412
8.26.4.13 dpi . . . 413
8.26.4.14 dpi . . . 413
8.26.4.15 dpiColor . . . 414
8.26.4.16 dpiColor . . . 414
8.26.4.17 dpiDown . . . 414
8.26.4.18 dpiEnabled . . . 415
8.26.4.19 dpiEnabled . . . 415
8.26.4.20 dpiIndicator . . . 416
8.26.4.21 dpiIndicator . . . 416
8.26.4.22 dpiUp . . . 416
8.26.4.23 getIndicator . . . 417
8.26.4.24 iOpacity . . . 418
8.26.4.25 iOpacity . . . 418
8.26.4.26 liftHeight . . . 418
8.26.4.27 liftHeight . . . 419

8.26.4.28 light . . . 419
8.26.4.29 lightIndicator . . . 420
8.26.4.30 load . . . 420
8.26.4.31 modeParent . . . 422
8.26.4.32 needsSave . . . 423
8.26.4.33 operator= . . . 423
8.26.4.34 popDpi . . . 424
8.26.4.35 pushDpi . . . 424
8.26.4.36 pushDpi . . . 425
8.26.4.37 pushSniper . . . 426
8.26.4.38 save . . . 426
8.26.4.39 setIndicator . . . 427
8.26.4.40 setNeedsUpdate . . . 428
8.26.4.41 settingsUpdated . . . 428
8.26.4.42 sniperDpi . . . 429
8.26.4.43 sniperDpi . . . 429
8.26.4.44 sniperDpi . . . 430
8.26.4.45 update . . . 430
8.26.5 Field Documentation . . . 432
8.26.5.1 _angleSnap . . . 432
8.26.5.2 _dpiIndicator . . . 432
8.26.5.3 _iOpacity . . . 432
8.26.5.4 _liftHeight . . . 432
8.26.5.5 _needsSave . . . 432
8.26.5.6 _needsUpdate . . . 432
8.26.5.7 DPI_COUNT . . . 433
8.26.5.8 DPI_MAX . . . 433
8.26.5.9 DPI_MIN . . . 433
8.26.5.10 dpiClr . . . 433
8.26.5.11 dpiCurIdx . . . 433
8.26.5.12 dpiCurX . . . 433
8.26.5.13 dpiCurY . . . 433
8.26.5.14 dpiLastIdx . . . 433
8.26.5.15 dpiOn . . . 433
8.26.5.16 dpiX . . . 434
8.26.5.17 dpiY . . . 434
8.26.5.18 HW_I_COUNT . . . 434
8.26.5.19 hwIType . . . 434
8.26.5.20 I_COUNT . . . 434
8.26.5.21 iColor . . . 434

8.26.5.22 iEnable . . . 434
8.26.5.23 light100Color . . . 434
8.26.5.24 muteNAColor . . . 434
8.26.5.25 OTHER . . . 435
8.26.5.26 pushedDpis . . . 435
8.26.5.27 runningPushIdx . . . 435
8.26.5.28 SNIPER . . . 435
8.27 KbProfile Class Reference . . . 435
8.27.1 Detailed Description . . . 438
8.27.2 Member Typedef Documentation . . . 438
8.27.2.1 ModeList . . . 438
8.27.3 Constructor & Destructor Documentation . . . 438
8.27.3.1 KbProfile . . . 438
8.27.3.2 KbProfile . . . 439
8.27.3.3 KbProfile . . . 439
8.27.4 Member Function Documentation . . . 440
8.27.4.1 append . . . 440
8.27.4.2 currentMode . . . 440
8.27.4.3 currentMode . . . 441
8.27.4.4 find . . . 441
8.27.4.5 id . . . 442
8.27.4.6 id . . . 442
8.27.4.7 indexOf . . . 443
8.27.4.8 insert . . . 443
8.27.4.9 keyMap . . . 444
8.27.4.10 keyMap . . . 444
8.27.4.11 modeCount . . . 445
8.27.4.12 modes . . . 445
8.27.4.13 modes . . . 446
8.27.4.14 move . . . 446
8.27.4.15 name . . . 447
8.27.4.16 name . . . 447
8.27.4.17 needsSave . . . 448
8.27.4.18 newId . . . 448
8.27.4.19 removeAll . . . 449
8.27.4.20 save . . . 450
8.27.4.21 setNeedsSave . . . 451
8.27.4.22 setNeedsUpdate . . . 451
8.27.5 Field Documentation . . . 452
8.27.5.1 _currentMode . . . 452

8.27.5.2 _id . . . 452
8.27.5.3 _keyMap . . . 452
8.27.5.4 _modes . . . 452
8.27.5.5 _name . . . 452
8.27.5.6 _needsSave . . . 452
8.28 KbProfileDialog Class Reference . . . 453
8.28.1 Detailed Description . . . 455
8.28.2 Constructor & Destructor Documentation . . . 455
8.28.2.1 KbProfileDialog . . . 455
8.28.2.2 ~KbProfileDialog . . . 455
8.28.3 Member Function Documentation . . . 455
8.28.3.1 addNewProfileItem . . . 455
8.28.3.2 on_profileList_currentItemChanged . . . 456
8.28.3.3 on_profileList_customContextMenuRequested . . . 456
8.28.3.4 on_profileList_itemChanged . . . 458
8.28.3.5 on_profileList_itemClicked . . . 459
8.28.3.6 profileList_reordered . . . 460
8.28.3.7 repopulate . . . 460
8.28.4 Field Documentation . . . 461
8.28.4.1 device . . . 461
8.28.4.2 GUID . . . 462
8.28.4.3 NEW_FLAG . . . 462
8.28.4.4 ui . . . 462
8.29 KbWidget Class Reference . . . 462
8.29.1 Detailed Description . . . 465
8.29.2 Constructor & Destructor Documentation . . . 465
8.29.2.1 KbWidget . . . 465
8.29.2.2 ~KbWidget . . . 466
8.29.3 Member Function Documentation . . . 467
8.29.3.1 addNewModeItem . . . 467
8.29.3.2 devUpdate . . . 467
8.29.3.3 modeChanged . . . 468
8.29.3.4 modeIcon . . . 469
8.29.3.5 modesList_reordered . . . 470
8.29.3.6 modeUpdate . . . 471
8.29.3.7 name . . . 471
8.29.3.8 on_fwUpdButton_clicked . . . 472
8.29.3.9 on_hwSaveButton_clicked . . . 473
8.29.3.10 on_modesList_currentItemChanged . . . 474
8.29.3.11 on_modesList_customContextMenuRequested . . . 475

8.29.3.12 on_modesList_itemChanged . . . 476
8.29.3.13 on_modesList_itemClicked . . . 477
8.29.3.14 on_profileBox_activated . . . 477
8.29.3.15 on_tabWidget_currentChanged . . . 478
8.29.3.16 profileChanged . . . 478
8.29.3.17 showFirstTab . . . 480
8.29.3.18 showFwUpdate . . . 480
8.29.3.19 showLastTab . . . 480
8.29.3.20 updateFwButton . . . 481
8.29.3.21 updateProfileList . . . 481
8.29.4 Field Documentation . . . 483
8.29.4.1 currentMode . . . 483
8.29.4.2 device . . . 483
8.29.4.3 GUID . . . 483
8.29.4.4 hasShownNewFW . . . 483
8.29.4.5 lastAutoSave . . . 483
8.29.4.6 NEW_FLAG . . . 483
8.29.4.7 ui . . . 483
8.30 Key Struct Reference . . . 484
8.30.1 Detailed Description . . . 484
8.30.2 Member Function Documentation . . . 485
8.30.2.1 friendlyName . . . 485
8.30.2.2 operator bool . . . 485
8.30.2.3 operator! . . . 485
8.30.2.4 storageName . . . 486
8.30.3 Field Documentation . . . 486
8.30.3.1 _friendlyName . . . 486
8.30.3.2 _storageName . . . 486
8.30.3.3 hasLed . . . 486
8.30.3.4 hasScan . . . 486
8.30.3.5 height . . . 486
8.30.3.6 name . . . 487
8.30.3.7 width . . . 487
8.30.3.8 x . . . 487
8.30.3.9 y . . . 487
8.31 KeyAction Class Reference . . . 487
8.31.1 Detailed Description . . . 491
8.31.2 Member Enumeration Documentation . . . 491
8.31.2.1 Type . . . 491
8.31.3 Constructor & Destructor Documentation . . . 492

8.31.3.1 KeyAction . . . 492
8.31.3.2 KeyAction . . . 492
8.31.3.3 ~KeyAction . . . 492
8.31.3.4 KeyAction . . . 492
8.31.4 Member Function Documentation . . . 492
8.31.4.1 adjustDisplay . . . 492
8.31.4.2 animAction . . . 493
8.31.4.3 animInfo . . . 494
8.31.4.4 defaultAction . . . 494
8.31.4.5 dpiAction . . . 495
8.31.4.6 dpiInfo . . . 496
8.31.4.7 driverName . . . 497
8.31.4.8 friendlyName . . . 497
8.31.4.9 isAnim . . . 498
8.31.4.10 isDPI . . . 499
8.31.4.11 isMacro . . . 499
8.31.4.12 isMedia . . . 500
8.31.4.13 isMouse . . . 500
8.31.4.14 isNormal . . . 500
8.31.4.15 isProgram . . . 501
8.31.4.16 isSpecial . . . 501
8.31.4.17 isUnbound . . . 502
8.31.4.18 isValidMacro . . . 502
8.31.4.19 keyEvent . . . 503
8.31.4.20 keyRelease . . . 506
8.31.4.21 lightAction . . . 506
8.31.4.22 lockAction . . . 507
8.31.4.23 macroAction . . . 507
8.31.4.24 macroContent . . . 508
8.31.4.25 macroDisplay . . . 508
8.31.4.26 macroFullLine . . . 509
8.31.4.27 macroLine . . . 510
8.31.4.28 modeAction . . . 510
8.31.4.29 noAction . . . 511
8.31.4.30 operator QString . . . 511
8.31.4.31 operator= . . . 511
8.31.4.32 programAction . . . 511
8.31.4.33 programInfo . . . 512
8.31.4.34 specialInfo . . . 512
8.31.4.35 type . . . 513

8.31.4.36 value . . . 513
8.31.5 Field Documentation . . . 514
8.31.5.1 _value . . . 514
8.31.5.2 DPI_CUSTOM . . . 514
8.31.5.3 DPI_DOWN . . . 514
8.31.5.4 DPI_SNIPER . . . 514
8.31.5.5 DPI_UP . . . 514
8.31.5.6 LIGHT_DOWN . . . 514
8.31.5.7 LIGHT_DOWN_WRAP . . . 515
8.31.5.8 LIGHT_UP . . . 515
8.31.5.9 LIGHT_UP_WRAP . . . 515
8.31.5.10 LOCK_OFF . . . 515
8.31.5.11 LOCK_ON . . . 515
8.31.5.12 LOCK_TOGGLE . . . 515
8.31.5.13 MODE_NEXT . . . 515
8.31.5.14 MODE_NEXT_WRAP . . . 515
8.31.5.15 MODE_PREV . . . 515
8.31.5.16 MODE_PREV_WRAP . . . 516
8.31.5.17 preProgram . . . 516
8.31.5.18 PROGRAM_PR_INDEF . . . 516
8.31.5.19 PROGRAM_PR_KPSTOP . . . 516
8.31.5.20 PROGRAM_PR_KRSTOP . . . 516
8.31.5.21 PROGRAM_PR_MULTI . . . 516
8.31.5.22 PROGRAM_RE_INDEF . . . 516
8.31.5.23 PROGRAM_RE_KPSTOP . . . 516
8.31.5.24 PROGRAM_RE_MULTI . . . 516
8.31.5.25 relProgram . . . 517
8.31.5.26 sniperValue . . . 517
8.32 KeyMap Class Reference . . . 517
8.32.1 Detailed Description . . . 520
8.32.2 Member Enumeration Documentation . . . 520
8.32.2.1 Layout . . . 520
8.32.2.2 Model . . . 520
8.32.3 Constructor & Destructor Documentation . . . 521
8.32.3.1 KeyMap . . . 521
8.32.3.2 KeyMap . . . 521
8.32.4 Member Function Documentation . . . 522
8.32.4.1 byPosition . . . 522
8.32.4.2 contains . . . 522
8.32.4.3 count . . . 523

8.32.4.4 friendlyName . . . 523
8.32.4.5 fromName . . . 525
8.32.4.6 fromStorage . . . 525
8.32.4.7 getLayout . . . 526
8.32.4.8 getLayout . . . 527
8.32.4.9 getModel . . . 527
8.32.4.10 getModel . . . 528
8.32.4.11 height . . . 528
8.32.4.12 isISO . . . 528
8.32.4.13 isISO . . . 529
8.32.4.14 isKeyboard . . . 530
8.32.4.15 isKeyboard . . . 530
8.32.4.16 isMouse . . . 531
8.32.4.17 isMouse . . . 531
8.32.4.18 key . . . 532
8.32.4.19 keys . . . 532
8.32.4.20 layout . . . 533
8.32.4.21 layoutNames . . . 534
8.32.4.22 locale . . . 534
8.32.4.23 map . . . 535
8.32.4.24 model . . . 535
8.32.4.25 modelHeight . . . 536
8.32.4.26 modelWidth . . . 536
8.32.4.27 name . . . 537
8.32.4.28 operator const QHash< QString, Key > & . . . 537
8.32.4.29 operator[] . . . 537
8.32.4.30 positions . . . 538
8.32.4.31 strLayout . . . 538
8.32.4.32 strModel . . . 539
8.32.4.33 toStorage . . . 539
8.32.4.34 width . . . 540
8.32.5 Field Documentation . . . 540
8.32.5.1 _keys . . . 540
8.32.5.2 keyHeight . . . 540
8.32.5.3 keyLayout . . . 540
8.32.5.4 keyModel . . . 541
8.32.5.5 keyWidth . . . 541
8.33 KeyWidget Class Reference . . . 541
8.33.1 Detailed Description . . . 544
8.33.2 Member Typedef Documentation . . . 544

8.33.2.1 BindMap . . . 544
8.33.3 Member Enumeration Documentation . . . 545
8.33.3.1 anonymous enum . . . 545
8.33.4 Constructor & Destructor Documentation . . . 545
8.33.4.1 KeyWidget . . . 545
8.33.5 Member Function Documentation . . . 545
8.33.5.1 bindMap . . . 545
8.33.5.2 bindMap . . . 545
8.33.5.3 clearAnimation . . . 546
8.33.5.4 clearSelection . . . 546
8.33.5.5 colorMap . . . 546
8.33.5.6 colorMap . . . 546
8.33.5.7 displayColorMap . . . 546
8.33.5.8 drawInfo . . . 547
8.33.5.9 map . . . 547
8.33.5.10 map . . . 548
8.33.5.11 monochrome . . . 548
8.33.5.12 monochrome . . . 548
8.33.5.13 mouseMoveEvent . . . 549
8.33.5.14 mousePressEvent . . . 550
8.33.5.15 mouseReleaseEvent . . . 551
8.33.5.16 paintEvent . . . 552
8.33.5.17 rgbMode . . . 557
8.33.5.18 rgbMode . . . 557
8.33.5.19 selectAll . . . 557
8.33.5.20 selectionChanged . . . 558
8.33.5.21 setAnimation . . . 558
8.33.5.22 setAnimationToSelection . . . 559
8.33.5.23 setSelection . . . 559
8.33.5.24 sidelightToggled . . . 560
8.33.6 Field Documentation . . . 560
8.33.6.1 _bindMap . . . 560
8.33.6.2 _colorMap . . . 560
8.33.6.3 _displayColorMap . . . 560
8.33.6.4 _indicators . . . 560
8.33.6.5 _monochrome . . . 560
8.33.6.6 _rgbMode . . . 560
8.33.6.7 animation . . . 560
8.33.6.8 keyMap . . . 561
8.33.6.9 mouseCurrentX . . . 561

8.33.6.10 mouseCurrentY . . . 561
8.33.6.11 mouseDownMode . . . 561
8.33.6.12 mouseDownX . . . 561
8.33.6.13 mouseDownY . . . 561
8.33.6.14 newSelection . . . 561
8.33.6.15 selection . . . 561
8.34 KPerfWidget Class Reference . . . 562
8.34.1 Detailed Description . . . 564
8.34.2 Data Structure Documentation . . . 564
8.34.2.1 struct KPerfWidget::IndicatorUi . . . 564
8.34.3 Member Typedef Documentation . . . 565
8.34.3.1 i_hw . . . 566
8.34.3.2 indicator . . . 566
8.34.4 Member Enumeration Documentation . . . 566
8.34.4.1 HwMode . . . 566
8.34.5 Constructor & Destructor Documentation . . . 566
8.34.5.1 KPerfWidget . . . 566
8.34.5.2 ~KPerfWidget . . . 568
8.34.6 Member Function Documentation . . . 568
8.34.6.1 mode2Raw . . . 568
8.34.6.2 on_copyButton_clicked . . . 568
8.34.6.3 on_intensityBox_valueChanged . . . 569
8.34.6.4 raw2Mode . . . 569
8.34.6.5 setPerf . . . 570
8.34.6.6 uiUpdated . . . 571
8.34.7 Field Documentation . . . 572
8.34.7.1 HW_I_COUNT . . . 572
8.34.7.2 I_COUNT . . . 573
8.34.7.3 indicators . . . 573
8.34.7.4 k95Widgets . . . 573
8.34.7.5 perf . . . 573
8.34.7.6 profile . . . 573
8.34.7.7 ui . . . 573
8.34.7.8 updateMapper . . . 573
8.35 LayoutDialog Class Reference . . . 573
8.35.1 Detailed Description . . . 575
8.35.2 Constructor & Destructor Documentation . . . 575
8.35.2.1 LayoutDialog . . . 575
8.35.2.2 ~LayoutDialog . . . 575
8.35.3 Member Function Documentation . . . 575

8.35.3.1 selected . . . 575
8.35.4 Field Documentation . . . 576
8.35.4.1 ui . . . 576
8.36 MacroReader Class Reference . . . 576
8.36.1 Detailed Description . . . 577
8.36.2 Constructor & Destructor Documentation . . . 578
8.36.2.1 MacroReader . . . 578
8.36.2.2 MacroReader . . . 578
8.36.2.3 ~MacroReader . . . 578
8.36.3 Member Function Documentation . . . 578
8.36.3.1 startWorkInAThread . . . 578
8.36.4 Field Documentation . . . 579
8.36.4.1 macroReaderThread . . . 579
8.37 MacroReaderThread Class Reference . . . 579
8.37.1 Detailed Description . . . 581
8.37.2 Constructor & Destructor Documentation . . . 581
8.37.2.1 MacroReaderThread . . . 581
8.37.3 Member Function Documentation . . . 582
8.37.3.1 readMacro . . . 582
8.37.3.2 run . . . 582
8.37.4 Field Documentation . . . 582
8.37.4.1 macroBox . . . 583
8.37.4.2 macroNumber . . . 583
8.37.4.3 macroPath . . . 583
8.37.4.4 macroText . . . 583
8.38 MainWindow Class Reference . . . 583
8.38.1 Detailed Description . . . 586
8.38.2 Constructor & Destructor Documentation . . . 586
8.38.2.1 MainWindow . . . 586
8.38.2.2 ~MainWindow . . . 588
8.38.3 Member Function Documentation . . . 588
8.38.3.1 addDevice . . . 588
8.38.3.2 checkFwUpdates . . . 589
8.38.3.3 cleanup . . . 590
8.38.3.4 closeEvent . . . 590
8.38.3.5 iconClicked . . . 591
8.38.3.6 quitApp . . . 591
8.38.3.7 removeDevice . . . 592
8.38.3.8 showFwUpdateNotification . . . 593
8.38.3.9 showWindow . . . 593

8.38.3.10 stateChange . . . 594
8.38.3.11 timerTick . . . 595
8.38.3.12 toggleTrayIcon . . . 596
8.38.3.13 updateVersion . . . 596
8.38.4 Field Documentation . . . 597
8.38.4.1 closeAction . . . 597
8.38.4.2 kbWidgets . . . 597
8.38.4.3 mainWindow . . . 597
8.38.4.4 restoreAction . . . 598
8.38.4.5 settingsWidget . . . 598
8.38.4.6 trayIcon . . . 598
8.38.4.7 trayIconMenu . . . 598
8.38.4.8 ui . . . 598
8.39 ModeSelectDialog Class Reference . . . 598
8.39.1 Detailed Description . . . 600
8.39.2 Constructor & Destructor Documentation . . . 600
8.39.2.1 ModeSelectDialog . . . 600
8.39.2.2 ∼ModeSelectDialog . . . 601
8.39.3 Member Function Documentation . . . 601
8.39.3.1 on_selAllButton_clicked . . . 601
8.39.3.2 on_selNoneButton_clicked . . . 601
8.39.3.3 selection . . . 601
8.39.4 Field Documentation . . . 602
8.39.4.1 _modeList . . . 602
8.39.4.2 ui . . . 602
8.40 MPerfWidget Class Reference . . . 602
8.40.1 Detailed Description . . . 605
8.40.2 Data Structure Documentation . . . 605
8.40.2.1 struct MPerfWidget::DpiUi . . . 605
8.40.3 Constructor & Destructor Documentation . . . 606
8.40.3.1 MPerfWidget . . . 606
8.40.3.2 ∼MPerfWidget . . . 608
8.40.4 Member Function Documentation . . . 608
8.40.4.1 boxXChanged . . . 608
8.40.4.2 boxYChanged . . . 609
8.40.4.3 colorChanged . . . 610
8.40.4.4 colorClicked . . . 611
8.40.4.5 enableChanged . . . 611
8.40.4.6 on_aSnapBox_clicked . . . 612
8.40.4.7 on_copyButton_clicked . . . 612

8.40.4.8 on_indicBox_clicked . . . 613
8.40.4.9 on_lHeightBox_activated . . . 613
8.40.4.10 on_spinBox_valueChanged . . . 614
8.40.4.11 on_xyBox_clicked . . . 614
8.40.4.12 setPerf . . . 615
8.40.4.13 sliderXMoved . . . 616
8.40.4.14 sliderYMoved . . . 617
8.40.5 Field Documentation . . . 618
8.40.5.1 _xyLink . . . 618
8.40.5.2 boxXMapper . . . 618
8.40.5.3 boxYMapper . . . 618
8.40.5.4 buttonMapper1 . . . 618
8.40.5.5 buttonMapper2 . . . 619
8.40.5.6 colorLink . . . 619
8.40.5.7 DPI_COUNT . . . 619
8.40.5.8 DPI_MAX . . . 619
8.40.5.9 DPI_MIN . . . 619
8.40.5.10 enableMapper . . . 619
8.40.5.11 isSetting . . . 619
8.40.5.12 perf . . . 619
8.40.5.13 profile . . . 619
8.40.5.14 sliderXMapper . . . 619
8.40.5.15 sliderYMapper . . . 620
8.40.5.16 stages . . . 620
8.40.5.17 ui . . . 620
8.41 GradientDialog::Preset Struct Reference . . . 620
8.41.1 Detailed Description . . . 621
8.41.2 Constructor & Destructor Documentation . . . 621
8.41.2.1 Preset . . . 621
8.41.2.2 Preset . . . 621
8.41.2.3 Preset . . . 621
8.41.3 Field Documentation . . . 621
8.41.3.1 builtIn . . . 621
8.41.3.2 gradient . . . 621
8.41.3.3 name . . . 622
8.42 QDialog Class Reference . . . 622
8.43 QIODevice Class Reference . . . 623
8.44 QIODevice_descriptor Struct Reference . . . 624
8.44.1 Detailed Description . . . 624
8.44.2 Constructor & Destructor Documentation . . . 624

8.44.2.1 QIODevice_descriptor . . . 624
8.44.3 Field Documentation . . . 624
8.44.3.1 pos . . . 624
8.45 QListWidget Class Reference . . . 625
8.46 QMainWindow Class Reference . . . 626
8.47 QObject Class Reference . . . 627
8.48 QPushButton Class Reference . . . 628
8.49 QThread Class Reference . . . 629
8.50 QuaAdler32 Class Reference . . . 630
8.50.1 Detailed Description . . . 631
8.50.2 Constructor & Destructor Documentation . . . 632
8.50.2.1 QuaAdler32 . . . 632
8.50.3 Member Function Documentation . . . 632
8.50.3.1 calculate . . . 632
8.50.3.2 reset . . . 632
8.50.3.3 update . . . 633
8.50.3.4 value . . . 633
8.50.4 Field Documentation . . . 633
8.50.4.1 checksum . . . 633
8.51 QuaChecksum32 Class Reference . . . 633
8.51.1 Detailed Description . . . 635
8.51.2 Member Function Documentation . . . 635
8.51.2.1 calculate . . . 635
8.51.2.2 reset . . . 635
8.51.2.3 update . . . 635
8.51.2.4 value . . . 635
8.52 QuaCrc32 Class Reference . . . 636
8.52.1 Detailed Description . . . 637
8.52.2 Constructor & Destructor Documentation . . . 638
8.52.2.1 QuaCrc32 . . . 638
8.52.3 Member Function Documentation . . . 638
8.52.3.1 calculate . . . 638
8.52.3.2 reset . . . 638
8.52.3.3 update . . . 639
8.52.3.4 value . . . 639
8.52.4 Field Documentation . . . 639
8.52.4.1 checksum . . . 639
8.53 QuaGzipFile Class Reference . . . 639
8.53.1 Detailed Description . . . 642
8.53.2 Constructor & Destructor Documentation . . . 642

8.53.2.1 QuaGzipFile . . . 642
8.53.2.2 QuaGzipFile . . . 642
8.53.2.3 QuaGzipFile . . . 643
8.53.2.4 ~QuaGzipFile . . . 643
8.53.2.5 QuaGzipFile . . . 643
8.53.3 Member Function Documentation . . . 643
8.53.3.1 close . . . 643
8.53.3.2 flush . . . 644
8.53.3.3 getFileName . . . 644
8.53.3.4 isSequential . . . 644
8.53.3.5 open . . . 644
8.53.3.6 open . . . 645
8.53.3.7 operator= . . . 645
8.53.3.8 readData . . . 645
8.53.3.9 setFileName . . . 645
8.53.3.10 writeData . . . 646
8.53.4 Field Documentation . . . 646
8.53.4.1 d . . . 646
8.54 QuaZIODevice Class Reference . . . 646
8.54.1 Detailed Description . . . 649
8.54.2 Constructor & Destructor Documentation . . . 649
8.54.2.1 QuaZIODevice . . . 649
8.54.2.2 ~QuaZIODevice . . . 649
8.54.3 Member Function Documentation . . . 649
8.54.3.1 close . . . 649
8.54.3.2 flush . . . 650
8.54.3.3 getIoDevice . . . 651
8.54.3.4 isSequential . . . 651
8.54.3.5 open . . . 651
8.54.3.6 readData . . . 652
8.54.3.7 writeData . . . 653
8.54.4 Field Documentation . . . 653
8.54.4.1 d . . . 653
8.55 QuaZip Class Reference . . . 654
8.55.1 Detailed Description . . . 656
8.55.2 Member Enumeration Documentation . . . 657
8.55.2.1 CaseSensitivity . . . 657
8.55.2.2 Constants . . . 657
8.55.2.3 Mode . . . 657
8.55.3 Constructor & Destructor Documentation . . . 658

8.55.3.1 QuaZip . . . 658
8.55.3.2 QuaZip . . . 658
8.55.3.3 QuaZip . . . 658
8.55.3.4 QuaZip . . . 658
8.55.3.5 ~QuaZip . . . 658
8.55.4 Member Function Documentation . . . 659
8.55.4.1 close . . . 659
8.55.4.2 convertCaseSensitivity . . . 660
8.55.4.3 getComment . . . 661
8.55.4.4 getCommentCodec . . . 661
8.55.4.5 getCurrentFileInfo . . . 662
8.55.4.6 getCurrentFileInfo . . . 663
8.55.4.7 getCurrentFileName . . . 664
8.55.4.8 getEntriesCount . . . 665
8.55.4.9 getFileInfoList . . . 666
8.55.4.10 getFileInfoList64 . . . 667
8.55.4.11 getFileNameCodec . . . 667
8.55.4.12 getFileNameList . . . 668
8.55.4.13 getIoDevice . . . 668
8.55.4.14 getMode . . . 668
8.55.4.15 getUnzFile . . . 669
8.55.4.16 getZipError . . . 670
8.55.4.17 getZipFile . . . 671
8.55.4.18 getZipName . . . 671
8.55.4.19 goToFirstFile . . . 672
8.55.4.20 goToNextFile . . . 673
8.55.4.21 hasCurrentFile . . . 674
8.55.4.22 isAutoClose . . . 674
8.55.4.23 isDataDescriptorWritingEnabled . . . 674
8.55.4.24 isOpen . . . 675
8.55.4.25 isZip64Enabled . . . 675
8.55.4.26 open . . . 676
8.55.4.27 operator= . . . 679
8.55.4.28 setAutoClose . . . 679
8.55.4.29 setComment . . . 680
8.55.4.30 setCommentCodec . . . 680
8.55.4.31 setCommentCodec . . . 680
8.55.4.32 setCurrentFile . . . 681
8.55.4.33 setDataDescriptorWritingEnabled . . . 682
8.55.4.34 setDefaultFileNameCodec . . . 683

8.55.4.35 setDefaultFileNameCodec . . . 684
8.55.4.36 setFileNameCodec . . . 684
8.55.4.37 setFileNameCodec . . . 684
8.55.4.38 setIoDevice . . . 684
8.55.4.39 setZip64Enabled . . . 685
8.55.4.40 setZipName . . . 685
8.55.5 Friends And Related Function Documentation . . . 686
8.55.5.1 QuaZipPrivate . . . 686
8.55.6 Field Documentation . . . 686
8.55.6.1 p . . . 686
8.56 QuaZipDir Class Reference . . . 686
8.56.1 Detailed Description . . . 689
8.56.2 Constructor & Destructor Documentation . . . 689
8.56.2.1 QuaZipDir . . . 689
8.56.2.2 QuaZipDir . . . 689
8.56.2.3 ~QuaZipDir . . . 690
8.56.3 Member Function Documentation . . . 690
8.56.3.1 caseSensitivity . . . 690
8.56.3.2 cd . . . 690
8.56.3.3 cdUp . . . 692
8.56.3.4 count . . . 692
8.56.3.5 dirName . . . 693
8.56.3.6 entryInfoList . . . 693
8.56.3.7 entryInfoList . . . 694
8.56.3.8 entryInfoList64 . . . 694
8.56.3.9 entryInfoList64 . . . 695
8.56.3.10 entryList . . . 695
8.56.3.11 entryList . . . 696
8.56.3.12 exists . . . 696
8.56.3.13 exists . . . 697
8.56.3.14 filePath . . . 698
8.56.3.15 filter . . . 699
8.56.3.16 isRoot . . . 699
8.56.3.17 nameFilters . . . 699
8.56.3.18 operator!= . . . 700
8.56.3.19 operator= . . . 700
8.56.3.20 operator== . . . 700
8.56.3.21 operator[] . . . 700
8.56.3.22 path . . . 701
8.56.3.23 relativeFilePath . . . 701

8.56.3.24 setCaseSensitivity . . . 701
8.56.3.25 setFilter . . . 702
8.56.3.26 setNameFilters . . . 702
8.56.3.27 setPath . . . 702
8.56.3.28 setSorting . . . 703
8.56.3.29 sorting . . . 703
8.56.4 Field Documentation . . . 703
8.56.4.1 d . . . 703
8.57 QuaZipFile Class Reference . . . 704
8.57.1 Detailed Description . . . 706
8.57.2 Sequential or random-access? . . . 707
8.57.3 Constructor & Destructor Documentation . . . 707
8.57.3.1 QuaZipFile . . . 707
8.57.3.2 QuaZipFile . . . 707
8.57.3.3 QuaZipFile . . . 707
8.57.3.4 QuaZipFile . . . 707
8.57.3.5 QuaZipFile . . . 708
8.57.3.6 QuaZipFile . . . 708
8.57.3.7 ∼QuaZipFile . . . 709
8.57.4 Member Function Documentation . . . 709
8.57.4.1 atEnd . . . 709
8.57.4.2 bytesAvailable . . . 710
8.57.4.3 close . . . 710
8.57.4.4 csize . . . 711
8.57.4.5 getActualFileName . . . 712
8.57.4.6 getCaseSensitivity . . . 713
8.57.4.7 getFileInfo . . . 713
8.57.4.8 getFileInfo . . . 714
8.57.4.9 getFileName . . . 715
8.57.4.10 getZip . . . 715
8.57.4.11 getZipError . . . 715
8.57.4.12 getZipName . . . 716
8.57.4.13 isRaw . . . 716
8.57.4.14 isSequential . . . 717
8.57.4.15 open . . . 717
8.57.4.16 open . . . 718
8.57.4.17 open . . . 718
8.57.4.18 open . . . 719
8.57.4.19 operator= . . . 721
8.57.4.20 pos . . . 721

8.57.4.21 readData . . . 722
8.57.4.22 setFileName . . . 723
8.57.4.23 setZip . . . 723
8.57.4.24 setZipName . . . 724
8.57.4.25 size . . . 724
8.57.4.26 usize . . . 725
8.57.4.27 writeData . . . 726
8.57.5 Friends And Related Function Documentation . . . 726
8.57.5.1 QuaZipFilePrivate . . . 727
8.57.6 Field Documentation . . . 727
8.57.6.1 p . . . 727
8.58 QuaZipFileInfo Struct Reference . . . 727
8.58.1 Detailed Description . . . 728
8.58.2 Member Function Documentation . . . 728
8.58.2.1 getPermissions . . . 728
8.58.3 Field Documentation . . . 729
8.58.3.1 comment . . . 729
8.58.3.2 compressedSize . . . 729
8.58.3.3 crc . . . 729
8.58.3.4 dateTime . . . 729
8.58.3.5 diskNumberStart . . . 729
8.58.3.6 externalAttr . . . 729
8.58.3.7 extra . . . 729
8.58.3.8 flags . . . 730
8.58.3.9 internalAttr . . . 730
8.58.3.10 method . . . 730
8.58.3.11 name . . . 730
8.58.3.12 uncompressedSize . . . 730
8.58.3.13 versionCreated . . . 730
8.58.3.14 versionNeeded . . . 730
8.59 QuaZipFileInfo64 Struct Reference . . . 730
8.59.1 Detailed Description . . . 732
8.59.2 Member Function Documentation . . . 732
8.59.2.1 getNTFSaTime . . . 732
8.59.2.2 getNTFScTime . . . 733
8.59.2.3 getNTFSmTime . . . 733
8.59.2.4 getPermissions . . . 734
8.59.2.5 isEncrypted . . . 735
8.59.2.6 toQuaZipFileInfo . . . 735
8.59.3 Field Documentation . . . 736

8.59.3.1 comment . . . 736
8.59.3.2 compressedSize . . . 736
8.59.3.3 crc . . . 736
8.59.3.4 dateTime . . . 736
8.59.3.5 diskNumberStart . . . 736
8.59.3.6 externalAttr . . . 737
8.59.3.7 extra . . . 737
8.59.3.8 flags . . . 737
8.59.3.9 internalAttr . . . 737
8.59.3.10 method . . . 737
8.59.3.11 name . . . 737
8.59.3.12 uncompressedSize . . . 737
8.59.3.13 versionCreated . . . 737
8.59.3.14 versionNeeded . . . 737
8.60 QuaZipFilePrivate Class Reference . . . 738
8.60.1 Detailed Description . . . 739
8.60.2 Constructor & Destructor Documentation . . . 739
8.60.2.1 QuaZipFilePrivate . . . 739
8.60.2.2 QuaZipFilePrivate . . . 739
8.60.2.3 QuaZipFilePrivate . . . 739
8.60.2.4 QuaZipFilePrivate . . . 740
8.60.2.5 ~QuaZipFilePrivate . . . 740
8.60.3 Member Function Documentation . . . 740
8.60.3.1 resetZipError . . . 740
8.60.3.2 setZipError . . . 740
8.60.4 Friends And Related Function Documentation . . . 741
8.60.4.1 QuaZipFile . . . 741
8.60.5 Field Documentation . . . 741
8.60.5.1 caseSensitivity . . . 741
8.60.5.2 crc . . . 741
8.60.5.3 fileName . . . 741
8.60.5.4 internal . . . 742
8.60.5.5 q . . . 742
8.60.5.6 raw . . . 742
8.60.5.7 uncompressedSize . . . 742
8.60.5.8 writePos . . . 742
8.60.5.9 zip . . . 742
8.60.5.10 zipError . . . 742
8.61 QuaZipNewInfo Struct Reference . . . 743
8.61.1 Detailed Description . . . 744

8.61.2 Constructor & Destructor Documentation . . . 744
8.61.2.1 QuaZipNewInfo . . . 744
8.61.2.2 QuaZipNewInfo . . . 744
8.61.2.3 QuaZipNewInfo . . . 745
8.61.2.4 QuaZipNewInfo . . . 746
8.61.3 Member Function Documentation . . . 746
8.61.3.1 setFileDateTime . . . 746
8.61.3.2 setFileNTFSaTime . . . 746
8.61.3.3 setFileNTFScTime . . . 747
8.61.3.4 setFileNTFSmTime . . . 748
8.61.3.5 setFileNTFSTimes . . . 749
8.61.3.6 setFilePermissions . . . 749
8.61.3.7 setPermissions . . . 750
8.61.4 Field Documentation . . . 750
8.61.4.1 comment . . . 750
8.61.4.2 dateTime . . . 750
8.61.4.3 externalAttr . . . 751
8.61.4.4 extraGlobal . . . 751
8.61.4.5 extraLocal . . . 751
8.61.4.6 internalAttr . . . 751
8.61.4.7 name . . . 751
8.61.4.8 uncompressedSize . . . 751
8.62 QuaZipPrivate Class Reference . . . 751
8.62.1 Detailed Description . . . 753
8.62.2 Constructor & Destructor Documentation . . . 753
8.62.2.1 QuaZipPrivate . . . 753
8.62.2.2 QuaZipPrivate . . . 754
8.62.2.3 QuaZipPrivate . . . 754
8.62.3 Member Function Documentation . . . 754
8.62.3.1 addCurrentFileToDirectoryMap . . . 754
8.62.3.2 clearDirectoryMap . . . 755
8.62.3.3 getDefaultFileNameCodec . . . 756
8.62.3.4 getFileInfoList . . . 756
8.62.3.5 goToFirstUnmappedFile . . . 757
8.62.4 Friends And Related Function Documentation . . . 758
8.62.4.1 QuaZip . . . 758
8.62.5 Field Documentation . . . 758
8.62.5.1 "@4 . . . 758
8.62.5.2 autoClose . . . 758
8.62.5.3 comment . . . 758

8.62.5.4 commentCodec . . . 758
8.62.5.5 dataDescriptorWritingEnabled . . . 759
8.62.5.6 defaultFileNameCodec . . . 759
8.62.5.7 directoryCaseInsensitive . . . 759
8.62.5.8 directoryCaseSensitive . . . 759
8.62.5.9 fileNameCodec . . . 759
8.62.5.10 hasCurrentFile_f . . . 759
8.62.5.11 ioDevice . . . 759
8.62.5.12 lastMappedDirectoryEntry . . . 759
8.62.5.13 mode . . . 759
8.62.5.14 q . . . 760
8.62.5.15 zip64 . . . 760
8.62.5.16 zipError . . . 760
8.62.5.17 zipName . . . 760
8.63 QWidget Class Reference . . . 760
8.64 RebindWidget Class Reference . . . 761
8.64.1 Detailed Description . . . 765
8.64.2 Constructor & Destructor Documentation . . . 765
8.64.2.1 RebindWidget . . . 765
8.64.2.2 ~RebindWidget . . . 766
8.64.3 Member Function Documentation . . . 766
8.64.3.1 applyChanges . . . 766
8.64.3.2 convertMacroBox . . . 768
8.64.3.3 helpStatus . . . 768
8.64.3.4 on_animBox_currentIndexChanged . . . 769
8.64.3.5 on_animButton_clicked . . . 770
8.64.3.6 on_applyButton_clicked . . . 770
8.64.3.7 on_btnClearMacro_clicked . . . 771
8.64.3.8 on_btnStartMacro_clicked . . . 771
8.64.3.9 on_btnStopMacro_clicked . . . 772
8.64.3.10 on_cancelButton_clicked . . . 772
8.64.3.11 on_dpiBox_currentIndexChanged . . . 773
8.64.3.12 on_dpiButton_clicked . . . 774
8.64.3.13 on_fnBox_currentIndexChanged . . . 774
8.64.3.14 on_fnButton_clicked . . . 774
8.64.3.15 on_lightBox_currentIndexChanged . . . 775
8.64.3.16 on_lightButton_clicked . . . 775
8.64.3.17 on_lockBox_currentIndexChanged . . . 775
8.64.3.18 on_lockButton_clicked . . . 776
8.64.3.19 on_mb2Box_currentIndexChanged . . . 776

8.64.3.20 on_mb2Button_clicked . . . 776
8.64.3.21 on_mbBox_currentIndexChanged . . . 776
8.64.3.22 on_mbButton_clicked . . . 777
8.64.3.23 on_mediaBox_currentIndexChanged . . . 777
8.64.3.24 on_mediaButton_clicked . . . 777
8.64.3.25 on_modBox_currentIndexChanged . . . 778
8.64.3.26 on_modButton_clicked . . . 778
8.64.3.27 on_modeBox_currentIndexChanged . . . 778
8.64.3.28 on_modeButton_clicked . . . 779
8.64.3.29 on_numBox_currentIndexChanged . . . 779
8.64.3.30 on_numButton_clicked . . . 779
8.64.3.31 on_programKpBox_textChanged . . . 779
8.64.3.32 on_programKpButton_clicked . . . 780
8.64.3.33 on_programKpSIBox_clicked . . . 780
8.64.3.34 on_programKrBox_textChanged . . . 780
8.64.3.35 on_programKrButton_clicked . . . 781
8.64.3.36 on_programKrSIBox_clicked . . . 781
8.64.3.37 on_pteMacroBox_textChanged . . . 781
8.64.3.38 on_resetButton_clicked . . . 782
8.64.3.39 on_typingBox_currentIndexChanged . . . 782
8.64.3.40 on_typingButton_clicked . . . 783
8.64.3.41 on_unbindButton_clicked . . . 783
8.64.3.42 on_wheelBox_currentIndexChanged . . . 783
8.64.3.43 on_wheelButton_clicked . . . 784
8.64.3.44 setBind . . . 784
8.64.3.45 setBox . . . 785
8.64.3.46 setSelection . . . 787
8.64.4 Field Documentation . . . 791
8.64.4.1 bind . . . 791
8.64.4.2 fnKeys . . . 792
8.64.4.3 macReader . . . 792
8.64.4.4 mediaKeys . . . 792
8.64.4.5 modKeys . . . 792
8.64.4.6 mouseExtKeys . . . 792
8.64.4.7 mouseKeys . . . 792
8.64.4.8 numKeys . . . 792
8.64.4.9 profile . . . 792
8.64.4.10 selection . . . 792
8.64.4.11 TAB_ANIM . . . 793
8.64.4.12 TAB_KB . . . 793

8.64.4.13 TAB_MACRO . . . 793
8.64.4.14 TAB_MOUSE . . . 793
8.64.4.15 TAB_PROGRAM . . . 793
8.64.4.16 TAB_SPECIAL . . . 793
8.64.4.17 typingKeys . . . 793
8.64.4.18 ui . . . 793
8.64.4.19 wheelKeys . . . 794
8.65 RListWidget Class Reference . . . 794
8.65.1 Detailed Description . . . 795
8.65.2 Constructor & Destructor Documentation . . . 796
8.65.2.1 RListWidget . . . 796
8.65.3 Member Function Documentation . . . 796
8.65.3.1 change . . . 796
8.65.3.2 enter . . . 797
8.65.3.3 orderChanged . . . 797
8.65.3.4 rescanItems . . . 797
8.65.3.5 timerTick . . . 798
8.65.4 Field Documentation . . . 799
8.65.4.1 currentData . . . 799
8.65.4.2 DATA_ROLE . . . 799
8.65.4.3 dragged . . . 799
8.65.4.4 previousItems . . . 799
8.65.4.5 reorderTimer . . . 799
8.66 SettingsWidget Class Reference . . . 799
8.66.1 Detailed Description . . . 802
8.66.2 Constructor & Destructor Documentation . . . 802
8.66.2.1 SettingsWidget . . . 802
8.66.2.2 ~SettingsWidget . . . 804
8.66.3 Member Function Documentation . . . 804
8.66.3.1 on_altBox_activated . . . 804
8.66.3.2 on_autoFWBox_clicked . . . 805
8.66.3.3 on_capsBox_activated . . . 805
8.66.3.4 on_ctrlBox_activated . . . 805
8.66.3.5 on_extraButton_clicked . . . 806
8.66.3.6 on_layoutBox_activated . . . 806
8.66.3.7 on_loginItemBox_clicked . . . 806
8.66.3.8 on_pushButton_clicked . . . 807
8.66.3.9 on_shiftBox_activated . . . 807
8.66.3.10 on_winBox_activated . . . 807
8.66.3.11 pollUpdates . . . 808

8.66.3.12 setStatus . . . 808
8.66.3.13 setVersion . . . 808
8.66.3.14 showLayoutDialog . . . 809
8.66.3.15 updateModifiers . . . 810
8.66.4 Friends And Related Function Documentation . . . 811
8.66.4.1 MainWindow . . . 811
8.66.5 Field Documentation . . . 811
8.66.5.1 extra . . . 811
8.66.5.2 ui . . . 811
8.67 SGroup Class Reference . . . 811
8.67.1 Detailed Description . . . 813
8.67.2 Constructor & Destructor Documentation . . . 813
8.67.2.1 SGroup . . . 813
8.67.2.2 ∼SGroup . . . 813
8.67.3 Field Documentation . . . 813
8.67.3.1 _settings . . . 813
8.68 UsbId Struct Reference . . . 814
8.68.1 Detailed Description . . . 815
8.68.2 Constructor & Destructor Documentation . . . 815
8.68.2.1 UsbId . . . 815
8.68.2.2 UsbId . . . 815
8.68.2.3 UsbId . . . 815
8.68.3 Member Function Documentation . . . 815
8.68.3.1 guidString . . . 815
8.68.3.2 guidString . . . 815
8.68.3.3 hwModifiedString . . . 816
8.68.3.4 hwModifiedString . . . 816
8.68.3.5 modifiedString . . . 816
8.68.3.6 modifiedString . . . 817
8.68.3.7 newGuid . . . 817
8.68.3.8 newModified . . . 817
8.68.4 Field Documentation . . . 818
8.68.4.1 guid . . . 818
8.68.4.2 hwModified . . . 818
8.68.4.3 modified . . . 818

9 File Documentation **819**
9.1 src/ckb/animadddialog.cpp File Reference . . . 819
9.2 src/ckb/animadddialog.h File Reference . . . 819
9.3 src/ckb/animscript.cpp File Reference . . . 820

9.3.1 Function Documentation . . . 821
9.3.1.1 urlParam . . . 821
9.3.2 Variable Documentation . . . 821
9.3.2.1 ONE_DAY . . . 821
9.4 src/ckb/animscript.h File Reference . . . 822
9.4.1 Data Structure Documentation . . . 822
9.4.1.1 struct AnimScript::Param . . . 822
9.5 src/ckb/animsettingdialog.cpp File Reference . . . 823
9.5.1 Function Documentation . . . 824
9.5.1.1 angleFlip . . . 824
9.6 src/ckb/animsettingdialog.h File Reference . . . 825
9.7 src/ckb/autorun.cpp File Reference . . . 826
9.7.1 Variable Documentation . . . 826
9.7.1.1 settingPath . . . 827
9.8 src/ckb/autorun.h File Reference . . . 827
9.9 src/ckb/ckb-anim.h File Reference . . . 827
9.9.1 Data Structure Documentation . . . 830
9.9.1.1 struct ckb_key . . . 830
9.9.1.2 struct ckb_runctx . . . 830
9.9.1.3 struct ckb_gradient . . . 831
9.9.2 Macro Definition Documentation . . . 832
9.9.2.1 CKB_CONTAINER . . . 832
9.9.2.2 CKB_COPYRIGHT . . . 832
9.9.2.3 CKB_DESCRIPTION . . . 832
9.9.2.4 CKB_DOWN . . . 832
9.9.2.5 CKB_GRAD_MAX . . . 833
9.9.2.6 CKB_GUID . . . 833
9.9.2.7 CKB_KEYCLEAR . . . 833
9.9.2.8 CKB_KEYNAME_MAX . . . 833
9.9.2.9 CKB_KP_NAME . . . 833
9.9.2.10 CKB_KP_NONE . . . 833
9.9.2.11 CKB_KP_POSITION . . . 833
9.9.2.12 CKB_KPMODE . . . 833
9.9.2.13 CKB_LICENSE . . . 833
9.9.2.14 CKB_LIVEPARAMS . . . 833
9.9.2.15 CKB_MAX_WORD . . . 833
9.9.2.16 CKB_NAME . . . 834
9.9.2.17 CKB_PARAM . . . 834
9.9.2.18 CKB_PARAM_AGRADIENT . . . 834
9.9.2.19 CKB_PARAM_ANGLE . . . 834

9.9.2.20 CKB_PARAM_ARGB . . . 834
9.9.2.21 CKB_PARAM_BOOL . . . 834
9.9.2.22 CKB_PARAM_DOUBLE . . . 834
9.9.2.23 CKB_PARAM_GRADIENT . . . 834
9.9.2.24 CKB_PARAM_LABEL . . . 834
9.9.2.25 CKB_PARAM_LONG . . . 834
9.9.2.26 CKB_PARAM_RGB . . . 834
9.9.2.27 CKB_PARAM_STRING . . . 835
9.9.2.28 CKB_PARSE_AGRADIENT . . . 835
9.9.2.29 CKB_PARSE_ANGLE . . . 835
9.9.2.30 CKB_PARSE_ARGB . . . 835
9.9.2.31 CKB_PARSE_BOOL . . . 835
9.9.2.32 CKB_PARSE_DOUBLE . . . 835
9.9.2.33 CKB_PARSE_GRADIENT . . . 835
9.9.2.34 CKB_PARSE_LONG . . . 835
9.9.2.35 CKB_PARSE_RGB . . . 835
9.9.2.36 CKB_PARSE_STRING . . . 835
9.9.2.37 CKB_PREEMPT . . . 836
9.9.2.38 CKB_PRESET_END . . . 836
9.9.2.39 CKB_PRESET_PARAM . . . 836
9.9.2.40 CKB_PRESET_START . . . 836
9.9.2.41 CKB_REAL_ANGLE . . . 836
9.9.2.42 CKB_REPEAT . . . 836
9.9.2.43 CKB_TIME_ABSOLUTE . . . 836
9.9.2.44 CKB_TIME_DURATION . . . 836
9.9.2.45 CKB_TIMEMODE . . . 836
9.9.2.46 CKB_UP . . . 836
9.9.2.47 CKB_VERSION . . . 836
9.9.2.48 FALSE . . . 836
9.9.2.49 M_PI . . . 837
9.9.2.50 TRUE . . . 837
9.9.3 Function Documentation . . . 837
9.9.3.1 ckb_alpha_blend . . . 837
9.9.3.2 ckb_frame . . . 837
9.9.3.3 ckb_getline . . . 837
9.9.3.4 ckb_grad_color . . . 838
9.9.3.5 ckb_info . . . 839
9.9.3.6 ckb_init . . . 839
9.9.3.7 ckb_keypress . . . 839
9.9.3.8 ckb_parameter . . . 839

9.9.3.9 ckb_read_params . . . 840
9.9.3.10 ckb_scan_grad . . . 841
9.9.3.11 ckb_start . . . 841
9.9.3.12 ckb_time . . . 841
9.9.3.13 main . . . 842
9.9.3.14 printurl . . . 844
9.9.3.15 urldecode . . . 845
9.10 src/ckb/ckbsettings.cpp File Reference . . . 845
9.10.1 Macro Definition Documentation . . . 846
9.10.1.1 lockMutex . . . 846
9.10.1.2 lockMutexCache . . . 846
9.10.1.3 lockMutexStatic . . . 846
9.10.1.4 lockMutexStatic2 . . . 847
9.10.2 Function Documentation . . . 847
9.10.2.1 cacheWritesInProgress . . . 847
9.10.2.2 globalSettings . . . 847
9.10.2.3 settingsCacheMutex . . . 848
9.10.2.4 settingsMutex . . . 848
9.10.3 Variable Documentation . . . 848
9.10.3.1 _globalSettings . . . 848
9.10.3.2 globalCache . . . 848
9.10.3.3 globalThread . . . 848
9.11 src/ckb/ckbsettings.h File Reference . . . 848
9.12 src/ckb/ckbsettingswriter.cpp File Reference . . . 849
9.12.1 Macro Definition Documentation . . . 850
9.12.1.1 lockMutex . . . 850
9.12.1.2 lockMutexCache . . . 850
9.12.1.3 lockMutexStatic . . . 850
9.12.1.4 lockMutexStatic2 . . . 850
9.12.2 Variable Documentation . . . 851
9.12.2.1 cacheWritesInProgress . . . 851
9.12.2.2 settingsCacheMutex . . . 851
9.12.2.3 settingsMutex . . . 851
9.13 src/ckb/ckbsettingswriter.h File Reference . . . 851
9.14 src/ckb/colorbutton.cpp File Reference . . . 852
9.15 src/ckb/colorbutton.h File Reference . . . 852
9.16 src/ckb/colormap.cpp File Reference . . . 853
9.16.1 Function Documentation . . . 853
9.16.1.1 qs_strcmp . . . 854
9.17 src/ckb/colormap.h File Reference . . . 854

9.17.1 Typedef Documentation . . . 855
9.17.1.1 QColorMap . . . 855
9.17.1.2 QColorMapIterator . . . 855
9.17.1.3 QMutableColorMapIterator . . . 855
9.18 src/ckb/extrasettingswidget.cpp File Reference . . . 855
9.18.1 Variable Documentation . . . 856
9.18.1.1 lastSharedDimming . . . 856
9.19 src/ckb/extrasettingswidget.h File Reference . . . 856
9.20 src/ckb/fwupgradedialog.cpp File Reference . . . 857
9.20.1 Data Structure Documentation . . . 858
9.20.1.1 struct KbId . . . 858
9.20.2 Function Documentation . . . 858
9.20.2.1 DIALOG_HEIGHT_MAX . . . 859
9.20.2.2 verifyFw . . . 859
9.20.3 Variable Documentation . . . 859
9.20.3.1 DIALOG_HEIGHT_MIN . . . 859
9.20.3.2 DIALOG_WIDTH . . . 859
9.20.3.3 ids . . . 860
9.21 src/ckb/fwupgradedialog.h File Reference . . . 860
9.22 src/ckb/gradientbutton.cpp File Reference . . . 861
9.23 src/ckb/gradientbutton.h File Reference . . . 861
9.24 src/ckb/gradientdialog.cpp File Reference . . . 862
9.24.1 Variable Documentation . . . 863
9.24.1.1 prefsPath . . . 863
9.25 src/ckb/gradientdialog.h File Reference . . . 863
9.26 src/ckb/gradientdialogwidget.cpp File Reference . . . 864
9.27 src/ckb/gradientdialogwidget.h File Reference . . . 864
9.28 src/ckb/kb.cpp File Reference . . . 865
9.28.1 Variable Documentation . . . 866
9.28.1.1 activeDevices . . . 866
9.28.1.2 notifyPathMutex . . . 866
9.28.1.3 notifyPaths . . . 866
9.29 src/ckb/kb.h File Reference . . . 866
9.30 src/ckb/kbanim.cpp File Reference . . . 867
9.30.1 Typedef Documentation . . . 868
9.30.1.1 blendFunc . . . 868
9.30.2 Function Documentation . . . 868
9.30.2.1 blendAdd . . . 868
9.30.2.2 blendDivide . . . 868
9.30.2.3 blendMultiply . . . 868

9.30.2.4 blendNormal . . . 869
9.30.2.5 blendSubtract . . . 869
9.30.3 Variable Documentation . . . 869
9.30.3.1 functions . . . 869
9.31 src/ckb/kbanim.h File Reference . . . 869
9.32 src/ckb/kbanimwidget.cpp File Reference . . . 870
9.33 src/ckb/kbanimwidget.h File Reference . . . 870
9.34 src/ckb/kbbind.cpp File Reference . . . 871
9.35 src/ckb/kbbind.h File Reference . . . 872
9.36 src/ckb/kbbindwidget.cpp File Reference . . . 873
9.37 src/ckb/kbbindwidget.h File Reference . . . 873
9.38 src/ckb/kbfirmware.cpp File Reference . . . 874
9.38.1 Function Documentation . . . 875
9.38.1.1 tableName . . . 875
9.38.2 Variable Documentation . . . 875
9.38.2.1 AUTO_CHECK_TIME . . . 875
9.39 src/ckb/kbfirmware.h File Reference . . . 875
9.40 src/ckb/kblight.cpp File Reference . . . 876
9.40.1 Function Documentation . . . 877
9.40.1.1 lToS . . . 877
9.40.1.2 monoRgb . . . 878
9.40.1.3 sToL . . . 878
9.40.2 Variable Documentation . . . 879
9.40.2.1 _shareDimming . . . 879
9.40.2.2 activeLights . . . 879
9.41 src/ckb/kblight.h File Reference . . . 879
9.42 src/ckb/kblightwidget.cpp File Reference . . . 880
9.43 src/ckb/kblightwidget.h File Reference . . . 880
9.44 src/ckb/kbmanager.cpp File Reference . . . 881
9.44.1 Variable Documentation . . . 882
9.44.1.1 devpath . . . 882
9.45 src/ckb/kbmanager.h File Reference . . . 882
9.45.1 Macro Definition Documentation . . . 883
9.45.1.1 DAEMON_UNAVAILABLE_STR . . . 883
9.46 src/ckb/kbmode.cpp File Reference . . . 883
9.47 src/ckb/kbmode.h File Reference . . . 884
9.48 src/ckb/kbperf.cpp File Reference . . . 885
9.48.1 Macro Definition Documentation . . . 885
9.48.1.1 map_last . . . 885
9.49 src/ckb/kbperf.h File Reference . . . 885

9.50 src/ckb/kbprofile.cpp File Reference . . . 886
9.51 src/ckb/kbprofile.h File Reference . . . 887
9.52 src/ckb/kbprofiledialog.cpp File Reference . . . 887
9.53 src/ckb/kbprofiledialog.h File Reference . . . 888
9.54 src/ckb/kbwidget.cpp File Reference . . . 889
9.55 src/ckb/kbwidget.h File Reference . . . 890
9.56 src/ckb/keyaction.cpp File Reference . . . 891
9.56.1 Variable Documentation . . . 891
9.56.1.1 ANIM_KRSTOP . . . 891
9.56.1.2 ANIM_ONCE . . . 892
9.57 src/ckb/keyaction.h File Reference . . . 892
9.58 src/ckb/keymap.cpp File Reference . . . 892
9.58.1 Data Structure Documentation . . . 894
9.58.1.1 struct KeyPatch . . . 894
9.58.2 Macro Definition Documentation . . . 895
9.58.2.1 ANSI_ENTER_H . . . 895
9.58.2.2 ANSI_ENTER_W . . . 895
9.58.2.3 ANSI_ENTER_X . . . 895
9.58.2.4 ANSI_ENTER_Y . . . 895
9.58.2.5 ANSI_LSHIFT_W . . . 896
9.58.2.6 ANSI_LSHIFT_X . . . 896
9.58.2.7 K65_HEIGHT . . . 896
9.58.2.8 K65_TOP_COUNT . . . 896
9.58.2.9 K65_WIDTH . . . 896
9.58.2.10 K70_HEIGHT . . . 896
9.58.2.11 K70_WIDTH . . . 896
9.58.2.12 K70_X_START . . . 896
9.58.2.13 K95_HEIGHT . . . 896
9.58.2.14 K95_WIDTH . . . 896
9.58.2.15 KEYCOUNT_K95 . . . 897
9.58.2.16 KEYCOUNT_M65 . . . 897
9.58.2.17 KEYCOUNT_SABRE . . . 897
9.58.2.18 KEYCOUNT_SCIM . . . 897
9.58.2.19 KSTRAFE_HEIGHT . . . 897
9.58.2.20 KSTRAFE_WIDTH . . . 897
9.58.2.21 KSTRAFE_X_START . . . 897
9.58.2.22 M65_HEIGHT . . . 897
9.58.2.23 M65_WIDTH . . . 897
9.58.2.24 N_LAYOUTS . . . 898
9.58.2.25 N_MODELS . . . 898

9.58.2.26 NS . . . 898
9.58.2.27 patch . . . 898
9.58.2.28 PATCH_COUNT . . . 898
9.58.2.29 SABRE_HEIGHT . . . 898
9.58.2.30 SABRE_WIDTH . . . 898
9.58.2.31 SCIM_HEIGHT . . . 898
9.58.2.32 SCIM_WIDTH . . . 898
9.58.3 Function Documentation . . . 898
9.58.3.1 _patch . . . 898
9.58.3.2 getMap . . . 899
9.58.3.3 patchANSI . . . 901
9.58.3.4 patchISO . . . 902
9.58.4 Variable Documentation . . . 902
9.58.4.1 K65TopRow . . . 902
9.58.4.2 K95BaseMap . . . 902
9.58.4.3 K95Keys . . . 903
9.58.4.4 KStrafeKeys . . . 903
9.58.4.5 M65Keys . . . 904
9.58.4.6 patchDE . . . 904
9.58.4.7 patchDK . . . 904
9.58.4.8 patchDvorak . . . 904
9.58.4.9 patchES . . . 905
9.58.4.10 patchEU . . . 905
9.58.4.11 patchFR . . . 905
9.58.4.12 patchIT . . . 905
9.58.4.13 patchMX . . . 905
9.58.4.14 patchNO . . . 906
9.58.4.15 patchSE . . . 906
9.58.4.16 SabreKeys . . . 906
9.58.4.17 ScimKeys . . . 906
9.58.4.18 standardMaps . . . 907
9.59 src/ckb/keymap.h File Reference . . . 907
9.60 src/ckb/keywidget.cpp File Reference . . . 907
9.60.1 Function Documentation . . . 908
9.60.1.1 monoRgb . . . 908
9.60.2 Variable Documentation . . . 909
9.60.2.1 KEY_SIZE . . . 909
9.60.2.2 m65Overlay . . . 909
9.60.2.3 sabOverlay . . . 909
9.60.2.4 scimOverlay . . . 909

9.61 src/ckb/keywidget.h File Reference . . . 909
9.62 src/ckb/kperfwidget.cpp File Reference . . . 910
9.63 src/ckb/kperfwidget.h File Reference . . . 911
9.63.1 Data Structure Documentation . . . 912
9.63.1.1 struct KPerfWidget::IndicatorUi . . . 912
9.64 src/ckb/layoutdialog.cpp File Reference . . . 913
9.65 src/ckb/layoutdialog.h File Reference . . . 914
9.66 src/ckb/macroreader.cpp File Reference . . . 915
9.67 src/ckb/macroreader.h File Reference . . . 915
9.68 src/ckb/main.cpp File Reference . . . 916
9.68.1 Macro Definition Documentation . . . 917
9.68.1.1 SHMEM_SIZE . . . 917
9.68.1.2 SHMEM_SIZE_V015 . . . 918
9.68.2 Enumeration Type Documentation . . . 918
9.68.2.1 CommandLineParseResults . . . 918
9.68.3 Function Documentation . . . 918
9.68.3.1 appShare . . . 918
9.68.3.2 isRunning . . . 918
9.68.3.3 main . . . 919
9.68.3.4 parseCommandLine . . . 921
9.68.3.5 pidActive . . . 922
9.69 src/ckb/mainwindow.cpp File Reference . . . 923
9.69.1 Variable Documentation . . . 923
9.69.1.1 appShare . . . 923
9.69.1.2 configLabel . . . 923
9.70 src/ckb/mainwindow.h File Reference . . . 923
9.71 src/ckb/media.h File Reference . . . 925
9.71.1 Macro Definition Documentation . . . 925
9.71.1.1 ENUM_C . . . 925
9.71.1.2 ENUM_END_C . . . 925
9.71.1.3 EXTERN_C . . . 925
9.71.2 Enumeration Type Documentation . . . 925
9.71.2.1 muteState . . . 925
9.71.3 Function Documentation . . . 926
9.71.3.1 getMuteState . . . 926
9.72 src/ckb/media_linux.cpp File Reference . . . 926
9.72.1 Function Documentation . . . 927
9.72.1.1 getMuteState . . . 927
9.72.2 Variable Documentation . . . 927
9.72.2.1 muteProcess . . . 928

9.73 src/ckb/media_mac.m File Reference . . . 928
9.73.1 Function Documentation . . . 928
9.73.1.1 disableAppNap . . . 928
9.73.1.2 getMuteState . . . 929
9.74 src/ckb/modeselectdialog.cpp File Reference . . . 929
9.75 src/ckb/modeselectdialog.h File Reference . . . 930
9.76 src/ckb/mperfwidget.cpp File Reference . . . 931
9.76.1 Macro Definition Documentation . . . 932
9.76.1.1 SET_END . . . 932
9.76.1.2 SET_START . . . 932
9.76.2 Function Documentation . . . 932
9.76.2.1 dpiExp . . . 932
9.76.2.2 dpiLog . . . 932
9.76.2.3 dpiRound . . . 933
9.76.3 Variable Documentation . . . 933
9.76.3.1 DPI_BASE . . . 933
9.76.3.2 xyLinkPath . . . 933
9.77 src/ckb/mperfwidget.h File Reference . . . 933
9.77.1 Data Structure Documentation . . . 934
9.77.1.1 struct MPerfWidget::DpiUi . . . 934
9.78 src/ckb/quazip/crypt.h File Reference . . . 935
9.78.1 Macro Definition Documentation . . . 936
9.78.1.1 CRC32 . . . 937
9.78.1.2 zdecode . . . 937
9.78.1.3 zencode . . . 937
9.78.2 Function Documentation . . . 937
9.78.2.1 decrypt_byte . . . 937
9.78.2.2 init_keys . . . 937
9.78.2.3 update_keys . . . 938
9.79 src/ckb/quazip/ioapi.h File Reference . . . 939
9.79.1 Data Structure Documentation . . . 941
9.79.1.1 struct zlib_filefunc_def_s . . . 941
9.79.1.2 struct zlib_filefunc64_def_s . . . 941
9.79.1.3 struct zlib_filefunc64_32_def_s . . . 942
9.79.2 Macro Definition Documentation . . . 942
9.79.2.1 __USE_FILE_OFFSET64 . . . 942
9.79.2.2 __USE_LARGEFILE64 . . . 942
9.79.2.3 _FILE_OFFSET_BIT . . . 943
9.79.2.4 _LARGEFILE64_SOURCE . . . 943
9.79.2.5 OF . . . 943

9.79.2.6 ZCALLBACK . . . 943
9.79.2.7 ZCLOSE64 . . . 943
9.79.2.8 ZERROR64 . . . 943
9.79.2.9 ZFAKECLOSE64 . . . 943
9.79.2.10 ZLIB_FILEFUNC_MODE_CREATE . . . 943
9.79.2.11 ZLIB_FILEFUNC_MODE_EXISTING . . . 943
9.79.2.12 ZLIB_FILEFUNC_MODE_READ . . . 943
9.79.2.13 ZLIB_FILEFUNC_MODE_READWRITEFILTER . . . 944
9.79.2.14 ZLIB_FILEFUNC_MODE_WRITE . . . 944
9.79.2.15 ZLIB_FILEFUNC_SEEK_CUR . . . 944
9.79.2.16 ZLIB_FILEFUNC_SEEK_END . . . 944
9.79.2.17 ZLIB_FILEFUNC_SEEK_SET . . . 944
9.79.2.18 ZOPEN64 . . . 944
9.79.2.19 ZREAD64 . . . 944
9.79.2.20 ZSEEK64 . . . 944
9.79.2.21 ZTELL64 . . . 945
9.79.2.22 ZWRITE64 . . . 945
9.79.3 Typedef Documentation . . . 945
9.79.3.1 _Z_OF . . . 945
9.79.3.2 zlib_filefunc64_32_def . . . 945
9.79.3.3 zlib_filefunc64_def . . . 945
9.79.3.4 zlib_filefunc_def . . . 945
9.79.3.5 ZPOS64_T . . . 945
9.79.4 Function Documentation . . . 945
9.79.4.1 _Z_OF . . . 945
9.79.4.2 _Z_OF . . . 945
9.79.4.3 _Z_OF . . . 945
9.79.4.4 _Z_OF . . . 945
9.79.4.5 _Z_OF . . . 945
9.79.4.6 fill_zlib_filefunc64_32_def_from_filefunc32 . . . 946
9.80 src/ckb/quazip/JlCompress.cpp File Reference . . . 946
9.80.1 Function Documentation . . . 947
9.80.1.1 copyData . . . 947
9.81 src/ckb/quazip/JlCompress.h File Reference . . . 948
9.82 src/ckb/quazip/qioapi.cpp File Reference . . . 948
9.82.1 Macro Definition Documentation . . . 950
9.82.1.1 SEEK_CUR . . . 950
9.82.1.2 SEEK_END . . . 950
9.82.1.3 SEEK_SET . . . 950
9.82.2 Function Documentation . . . 950

9.82.2.1 call_zopen64 . . . 950
9.82.2.2 call_zseek64 . . . 950
9.82.2.3 call_ztell64 . . . 950
9.82.2.4 fill_qiodevice64_filefunc . . . 951
9.82.2.5 fill_qiodevice_filefunc . . . 952
9.82.2.6 fill_zlib_filefunc64_32_def_from_filefunc32 . . . 953
9.82.2.7 qiodevice64_seek_file_func . . . 954
9.82.2.8 qiodevice64_tell_file_func . . . 955
9.82.2.9 qiodevice_close_file_func . . . 955
9.82.2.10 qiodevice_error_file_func . . . 956
9.82.2.11 qiodevice_fakeclose_file_func . . . 956
9.82.2.12 qiodevice_open_file_func . . . 957
9.82.2.13 qiodevice_read_file_func . . . 958
9.82.2.14 qiodevice_seek_file_func . . . 958
9.82.2.15 qiodevice_tell_file_func . . . 959
9.82.2.16 qiodevice_write_file_func . . . 960
9.83 src/ckb/quazip/quaadler32.cpp File Reference . . . 960
9.84 src/ckb/quazip/quaadler32.h File Reference . . . 961
9.85 src/ckb/quazip/quachecksum32.h File Reference . . . 962
9.86 src/ckb/quazip/quacrc32.cpp File Reference . . . 963
9.87 src/ckb/quazip/quacrc32.h File Reference . . . 964
9.88 src/ckb/quazip/quagzipfile.cpp File Reference . . . 965
9.89 src/ckb/quazip/quagzipfile.h File Reference . . . 966
9.90 src/ckb/quazip/quaziodevice.cpp File Reference . . . 967
9.90.1 Macro Definition Documentation . . . 968
9.90.1.1 QUAZIO_INBUFSIZE . . . 968
9.90.1.2 QUAZIO_OUTBUFSIZE . . . 968
9.91 src/ckb/quazip/quaziodevice.h File Reference . . . 968
9.92 src/ckb/quazip/quazip.cpp File Reference . . . 969
9.92.1 Function Documentation . . . 969
9.92.1.1 QuaZip_getFileInfo . . . 969
9.92.1.2 QuaZip_getFileInfo . . . 970
9.92.1.3 QuaZip_getFileInfo . . . 970
9.92.1.4 QuaZip_getFileInfo . . . 971
9.93 src/ckb/quazip/quazip.h File Reference . . . 971
9.93.1 Macro Definition Documentation . . . 972
9.93.1.1 UNZ_OPENERROR . . . 972
9.94 src/ckb/quazip/quazip_global.h File Reference . . . 972
9.94.1 Macro Definition Documentation . . . 973
9.94.1.1 QUAZIP_EXPORT . . . 973

9.94.1.2 QUAZIP_EXTRA_NTFS_MAGIC . . . 973
9.94.1.3 QUAZIP_EXTRA_NTFS_TIME_MAGIC . . . 973
9.94.1.4 UNUSED . . . 973
9.95 src/ckb/quazip/quazipdir.cpp File Reference . . . 973
9.95.1 Function Documentation . . . 974
9.95.1.1 QuaZipDir_convertInfoList . . . 974
9.95.1.2 QuaZipDir_convertInfoList . . . 974
9.95.1.3 QuaZipDir_convertInfoList . . . 974
9.95.1.4 QuaZipDir_getFileInfo . . . 975
9.96 src/ckb/quazip/quazipdir.h File Reference . . . 975
9.97 src/ckb/quazip/quazipfile.cpp File Reference . . . 976
9.98 src/ckb/quazip/quazipfile.h File Reference . . . 977
9.99 src/ckb/quazip/quazipfileinfo.cpp File Reference . . . 978
9.99.1 Function Documentation . . . 979
9.99.1.1 getNTFSTime . . . 979
9.99.1.2 permissionsFromExternalAttr . . . 980
9.100 src/ckb/quazip/quazipfileinfo.h File Reference . . . 980
9.101 src/ckb/quazip/quazipnewinfo.cpp File Reference . . . 981
9.101.1 Function Documentation . . . 982
9.101.1.1 QuaZipNewInfo_init . . . 982
9.101.1.2 QuaZipNewInfo_setPermissions . . . 983
9.101.1.3 setNTFSTime . . . 983
9.102 src/ckb/quazip/quazipnewinfo.h File Reference . . . 985
9.103 src/ckb/quazip/unzip.c File Reference . . . 986
9.103.1 Data Structure Documentation . . . 989
9.103.1.1 struct unz_file_info64_internal_s . . . 989
9.103.1.2 struct file_in_zip64_read_info_s . . . 990
9.103.1.3 struct unz64_s . . . 991
9.103.2 Macro Definition Documentation . . . 992
9.103.2.1 ALLOC . . . 992
9.103.2.2 BUFREADCOMMENT . . . 992
9.103.2.3 CASESENSITIVITYDEFAULT_NO . . . 992
9.103.2.4 CASESENSITIVITYDEFAULTVALUE . . . 992
9.103.2.5 local . . . 992
9.103.2.6 SIZECENTRALDIRITEM . . . 992
9.103.2.7 SIZEZIPLOCALHEADER . . . 992
9.103.2.8 STRCMPCASENOSENTIVEFUNCTION . . . 992
9.103.2.9 TRYFREE . . . 993
9.103.2.10 UNZ_BUFSIZE . . . 993
9.103.2.11 UNZ_MAXFILENAMEINZIP . . . 993

9.103.3 Typedef Documentation . . . 993
9.103.3.1 unz_file_info64_internal . . . 993
9.103.3.2 z_crc_t . . . 993
9.103.4 Function Documentation . . . 993
9.103.4.1 _Z_OF . . . 993
9.103.4.2 _Z_OF . . . 993
9.103.4.3 _Z_OF . . . 993
9.103.4.4 _Z_OF . . . 993
9.103.4.5 _Z_OF . . . 993
9.103.4.6 strcmpcasenosensitive_internal . . . 993
9.103.4.7 unz64local_CheckCurrentFileCoherencyHeader . . . 994
9.103.4.8 unz64local_DosDateToTmuDate . . . 996
9.103.4.9 unz64local_getByte . . . 996
9.103.4.10 unz64local_GetCurrentFileInfoInternal . . . 997
9.103.4.11 unz64local_getLong . . . 1001
9.103.4.12 unz64local_getLong64 . . . 1002
9.103.4.13 unz64local_getShort . . . 1003
9.103.4.14 unz64local_SearchCentralDir . . . 1004
9.103.4.15 unz64local_SearchCentralDir64 . . . 1005
9.103.4.16 unzClearFlags . . . 1007
9.103.4.17 unzClose . . . 1007
9.103.4.18 unzCloseCurrentFile . . . 1008
9.103.4.19 unzeof . . . 1009
9.103.4.20 unzGetCurrentFileInfo . . . 1010
9.103.4.21 unzGetCurrentFileInfo64 . . . 1011
9.103.4.22 unzGetCurrentFileZStreamPos64 . . . 1012
9.103.4.23 unzGetFilePos . . . 1012
9.103.4.24 unzGetFilePos64 . . . 1012
9.103.4.25 unzGetGlobalComment . . . 1013
9.103.4.26 unzGetGlobalInfo . . . 1014
9.103.4.27 unzGetGlobalInfo64 . . . 1014
9.103.4.28 unzGetLocalExtrafield . . . 1014
9.103.4.29 unzGetOffset . . . 1015
9.103.4.30 unzGetOffset64 . . . 1015
9.103.4.31 unzGoToFilePos . . . 1016
9.103.4.32 unzGoToFilePos64 . . . 1016
9.103.4.33 unzGoToFirstFile . . . 1017
9.103.4.34 unzGoToNextFile . . . 1018
9.103.4.35 unzLocateFile . . . 1019
9.103.4.36 unzOpen . . . 1020

9.103.4.37 unzOpen2 . . . 1021
9.103.4.38 unzOpen2_64 . . . 1022
9.103.4.39 unzOpen64 . . . 1023
9.103.4.40 unzOpenCurrentFile . . . 1024
9.103.4.41 unzOpenCurrentFile2 . . . 1024
9.103.4.42 unzOpenCurrentFile3 . . . 1024
9.103.4.43 unzOpenCurrentFilePassword . . . 1027
9.103.4.44 unzOpenInternal . . . 1028
9.103.4.45 unzReadCurrentFile . . . 1031
9.103.4.46 unzSetFlags . . . 1034
9.103.4.47 unzSetOffset . . . 1035
9.103.4.48 unzSetOffset64 . . . 1035
9.103.4.49 unzStringFileNameCompare . . . 1036
9.103.4.50 unztell . . . 1036
9.103.4.51 unztell64 . . . 1037
9.103.5 Variable Documentation . . . 1037
9.103.5.1 unz_copyright . . . 1037
9.104 src/ckb/quazip/unzip.h File Reference . . . 1038
9.104.1 Data Structure Documentation . . . 1040
9.104.1.1 struct tm_unz_s . . . 1040
9.104.1.2 struct unz_global_info64_s . . . 1040
9.104.1.3 struct unz_global_info_s . . . 1041
9.104.1.4 struct unz_file_info64_s . . . 1042
9.104.1.5 struct unz_file_info_s . . . 1043
9.104.1.6 struct unz_file_pos_s . . . 1045
9.104.1.7 struct unz64_file_pos_s . . . 1046
9.104.2 Macro Definition Documentation . . . 1046
9.104.2.1 UNZ_AUTO_CLOSE . . . 1046
9.104.2.2 UNZ_BADZIPFILE . . . 1046
9.104.2.3 UNZ_CRCERROR . . . 1046
9.104.2.4 UNZ_DEFAULT_FLAGS . . . 1047
9.104.2.5 UNZ_END_OF_LIST_OF_FILE . . . 1047
9.104.2.6 UNZ_EOF . . . 1047
9.104.2.7 UNZ_ERRNO . . . 1047
9.104.2.8 UNZ_INTERNALERROR . . . 1047
9.104.2.9 UNZ_OK . . . 1047
9.104.2.10 UNZ_PARAMERROR . . . 1047
9.104.2.11 Z_BZIP2ED . . . 1048
9.104.3 Typedef Documentation . . . 1048
9.104.3.1 tm_unz . . . 1048

9.104.3.2 unz64_file_pos . . . 1048
9.104.3.3 unz_file_info . . . 1048
9.104.3.4 unz_file_info64 . . . 1048
9.104.3.5 unz_file_pos . . . 1048
9.104.3.6 unz_global_info . . . 1048
9.104.3.7 unz_global_info64 . . . 1048
9.104.3.8 unzFile . . . 1048
9.104.4 Function Documentation . . . 1048
9.104.4.1 _Z_OF . . . 1048
9.104.4.2 _Z_OF . . . 1048
9.104.4.3 _Z_OF . . . 1048
9.104.4.4 _Z_OF . . . 1048
9.104.4.5 _Z_OF . . . 1048
9.104.4.6 _Z_OF . . . 1048
9.104.4.7 _Z_OF . . . 1048
9.104.4.8 _Z_OF . . . 1048
9.104.4.9 _Z_OF . . . 1048
9.104.4.10 _Z_OF . . . 1048
9.104.4.11 _Z_OF . . . 1049
9.104.4.12 _Z_OF . . . 1049
9.104.4.13 _Z_OF . . . 1049
9.104.4.14 _Z_OF . . . 1049
9.104.4.15 _Z_OF . . . 1049
9.104.4.16 unzClearFlags . . . 1049
9.104.4.17 unzGetFilePos . . . 1049
9.104.4.18 unzGetFilePos64 . . . 1050
9.104.4.19 unzGetOffset . . . 1050
9.104.4.20 unzGetOffset64 . . . 1051
9.104.4.21 unzGoToFilePos . . . 1051
9.104.4.22 unzGoToFilePos64 . . . 1052
9.104.4.23 unzOpenInternal . . . 1053
9.104.4.24 unzSetFlags . . . 1056
9.104.4.25 unzSetOffset . . . 1057
9.104.4.26 unzSetOffset64 . . . 1057
9.105 src/ckb/quazip/zip.c File Reference . . . 1058
9.105.1 Data Structure Documentation . . . 1061
9.105.1.1 struct linkedlist_datablock_internal_s . . . 1061
9.105.1.2 struct linkedlist_data_s . . . 1062
9.105.1.3 struct curfile64_info . . . 1063
9.105.1.4 struct zip64_internal . . . 1064

9.105.2 Macro Definition Documentation . . . 1064
9.105.2.1 ALLOC . . . 1065
9.105.2.2 BUFREADCOMMENT . . . 1065
9.105.2.3 CENTRALHEADERMAGIC . . . 1065
9.105.2.4 CRC_LOCALHEADER_OFFSET . . . 1065
9.105.2.5 DESCRIPTORHEADERMAGIC . . . 1065
9.105.2.6 ENDHEADERMAGIC . . . 1065
9.105.2.7 FLAG_LOCALHEADER_OFFSET . . . 1065
9.105.2.8 INCLUDECRYPTINGCODE_IFCRYPTALLOWED . . . 1065
9.105.2.9 local . . . 1065
9.105.2.10 LOCALHEADERMAGIC . . . 1065
9.105.2.11 MAKEULONG64 . . . 1066
9.105.2.12 SEEK_CUR . . . 1066
9.105.2.13 SEEK_END . . . 1066
9.105.2.14 SEEK_SET . . . 1066
9.105.2.15 SIZECENTRALHEADER . . . 1066
9.105.2.16 SIZEDATA_INDATABLOCK . . . 1066
9.105.2.17 TRYFREE . . . 1066
9.105.2.18 VERSIONMADEBY . . . 1066
9.105.2.19 Z_BUFSIZE . . . 1066
9.105.2.20 Z_MAXFILENAMEINZIP . . . 1066
9.105.2.21 ZIP64ENDHEADERMAGIC . . . 1067
9.105.2.22 ZIP64ENDLOCHEADERMAGIC . . . 1067
9.105.3 Typedef Documentation . . . 1067
9.105.3.1 linkedlist_data . . . 1067
9.105.3.2 linkedlist_datablock_internal . . . 1067
9.105.3.3 z_crc_t . . . 1067
9.105.4 Function Documentation . . . 1067
9.105.4.1 _Z_OF . . . 1067
9.105.4.2 _Z_OF . . . 1067
9.105.4.3 _Z_OF . . . 1067
9.105.4.4 _Z_OF . . . 1067
9.105.4.5 _Z_OF . . . 1067
9.105.4.6 _Z_OF . . . 1067
9.105.4.7 add_data_in_datablock . . . 1067
9.105.4.8 allocate_new_datablock . . . 1069
9.105.4.9 free_datablock . . . 1069
9.105.4.10 free_linkedlist . . . 1070
9.105.4.11 init_linkedlist . . . 1070
9.105.4.12 LoadCentralDirectoryRecord . . . 1071

9.105.4.13 Write_EndOfCentralDirectoryRecord . . . 1074
9.105.4.14 Write_GlobalComment . . . 1076
9.105.4.15 Write_LocalFileHeader . . . 1076
9.105.4.16 Write_Zip64EndOfCentralDirectoryLocator . . . 1078
9.105.4.17 Write_Zip64EndOfCentralDirectoryRecord . . . 1079
9.105.4.18 zip64FlushWriteBuffer . . . 1081
9.105.4.19 zip64local_getByte . . . 1081
9.105.4.20 zip64local_getLong . . . 1082
9.105.4.21 zip64local_getLong64 . . . 1083
9.105.4.22 zip64local_getShort . . . 1084
9.105.4.23 zip64local_putValue . . . 1085
9.105.4.24 zip64local_putValue_inmemory . . . 1086
9.105.4.25 zip64local_SearchCentralDir . . . 1086
9.105.4.26 zip64local_SearchCentralDir64 . . . 1087
9.105.4.27 zip64local_TmzDateToDosDate . . . 1089
9.105.4.28 zipClearFlags . . . 1090
9.105.4.29 zipClose . . . 1090
9.105.4.30 zipCloseFileInZip . . . 1092
9.105.4.31 zipCloseFileInZipRaw . . . 1093
9.105.4.32 zipCloseFileInZipRaw64 . . . 1094
9.105.4.33 zipOpen . . . 1098
9.105.4.34 zipOpen2 . . . 1098
9.105.4.35 zipOpen2_64 . . . 1099
9.105.4.36 zipOpen3 . . . 1100
9.105.4.37 zipOpen64 . . . 1102
9.105.4.38 zipOpenNewFileInZip . . . 1103
9.105.4.39 zipOpenNewFileInZip2 . . . 1103
9.105.4.40 zipOpenNewFileInZip2_64 . . . 1104
9.105.4.41 zipOpenNewFileInZip3 . . . 1104
9.105.4.42 zipOpenNewFileInZip3_64 . . . 1105
9.105.4.43 zipOpenNewFileInZip4 . . . 1105
9.105.4.44 zipOpenNewFileInZip4_64 . . . 1106
9.105.4.45 zipOpenNewFileInZip64 . . . 1110
9.105.4.46 zipRemoveExtraInfoBlock . . . 1110
9.105.4.47 zipSetFlags . . . 1111
9.105.4.48 zipWriteInFileInZip . . . 1112
9.105.5 Variable Documentation . . . 1114
9.105.5.1 zip_copyright . . . 1114
9.106 src/ckb/quazip/zip.h File Reference . . . 1114
9.106.1 Data Structure Documentation . . . 1116

9.106.1.1 struct tm_zip_s . . . 1116
9.106.1.2 struct zip_fileinfo . . . 1117
9.106.2 Macro Definition Documentation . . . 1118
9.106.2.1 APPEND_STATUS_ADDINZIP . . . 1118
9.106.2.2 APPEND_STATUS_CREATE . . . 1118
9.106.2.3 APPEND_STATUS_CREATEAFTER . . . 1119
9.106.2.4 DEF_MEM_LEVEL . . . 1119
9.106.2.5 Z_BZIP2ED . . . 1119
9.106.2.6 ZIP_AUTO_CLOSE . . . 1119
9.106.2.7 ZIP_BADZIPFILE . . . 1119
9.106.2.8 ZIP_DEFAULT_FLAGS . . . 1119
9.106.2.9 ZIP_EOF . . . 1119
9.106.2.10 ZIP_ERRNO . . . 1119
9.106.2.11 ZIP_INTERNALERROR . . . 1119
9.106.2.12 ZIP_OK . . . 1120
9.106.2.13 ZIP_PARAMERROR . . . 1120
9.106.2.14 ZIP_SEQUENTIAL . . . 1120
9.106.2.15 ZIP_WRITE_DATA_DESCRIPTOR . . . 1120
9.106.3 Typedef Documentation . . . 1120
9.106.3.1 tm_zip . . . 1120
9.106.3.2 zipcharpc . . . 1120
9.106.3.3 zipFile . . . 1120
9.106.4 Function Documentation . . . 1120
9.106.4.1 _Z_OF . . . 1120
9.106.4.2 _Z_OF . . . 1120
9.106.4.3 _Z_OF . . . 1120
9.106.4.4 _Z_OF . . . 1121
9.106.4.5 _Z_OF . . . 1121
9.106.4.6 _Z_OF . . . 1121
9.106.4.7 _Z_OF . . . 1121
9.106.4.8 _Z_OF . . . 1121
9.106.4.9 _Z_OF . . . 1121
9.106.4.10 _Z_OF . . . 1121
9.106.4.11 _Z_OF . . . 1121
9.106.4.12 _Z_OF . . . 1121
9.106.4.13 _Z_OF . . . 1121
9.106.4.14 _Z_OF . . . 1121
9.106.4.15 _Z_OF . . . 1121
9.106.4.16 _Z_OF . . . 1121
9.106.4.17 _Z_OF . . . 1121

9.106.4.18 zipClearFlags . . . 1121
9.106.4.19 zipOpen3 . . . 1122
9.106.4.20 zipSetFlags . . . 1124
9.107 src/ckb/rebindwidget.cpp File Reference . . . 1125
9.107.1 Variable Documentation . . . 1125
9.107.1.1 DPI_CUST_IDX . . . 1125
9.107.1.2 DPI_OFFSET . . . 1125
9.108 src/ckb/rebindwidget.h File Reference . . . 1126
9.109 src/ckb/rlistwidget.cpp File Reference . . . 1126
9.110 src/ckb/rlistwidget.h File Reference . . . 1127
9.111 src/ckb/settingswidget.cpp File Reference . . . 1128
9.111.1 Function Documentation . . . 1129
9.111.1.1 right . . . 1129
9.111.2 Variable Documentation . . . 1129
9.111.2.1 devpath . . . 1129
9.111.2.2 modKeys . . . 1129
9.111.2.3 modNames . . . 1130
9.112 src/ckb/settingswidget.h File Reference . . . 1130

Index **1131**

Chapter 1

Todo List

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.

Chapter 2

Deprecated List

Class QuaZipFileInfo

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

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui 17

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

AutoRun . . . 77
ckb_gradient . . . 827
ckb_key . . . 827
ckb_runctx . . . 827
CkbSettings . . . 81
ColorMap . . . 107
curfile64_info . . . 1058
MPerfWidget::DpiUi . . . 602
file_in_zip64_read_info_s . . . 986
KbFirmware::FW . . . 125
KPerfWidget::IndicatorUi . . . 562
JlCompress . . . 169
KbId . . . 857
Key . . . 484
KeyMap . . . 517
KeyPatch . . . 892
linkedlist_data_s . . . 1058
linkedlist_datablock_internal_s . . . 1058
AnimScript::Param . . . 25
GradientDialog::Preset . . . 620
QDialog . . . 622
- AnimAddDialog . . . 19
- AnimSettingDialog . . . 57
- ExtraSettingsWidget . . . 116
- FwUpgradeDialog . . . 126
- GradientDialog . . . 141
- KbProfileDialog . . . 453
- LayoutDialog . . . 573
- ModeSelectDialog . . . 598

QIODevice . . . 623
- QuaGzipFile . . . 639
- QuaZIODevice . . . 646
- QuaZipFile . . . 704

QIODevice_descriptor . . . 624
QListWidget . . . 625
- RListWidget . . . 794

QMainWindow . . . 626

MainWindow . . . 583
QObject . . . 627
AnimScript . . . 25
CkbSettingsWriter . . . 96
KbAnim . . . 236
KbBind . . . 282
KbFirmware . . . 321
KbLight . . . 332
KbManager . . . 373
KbMode . . . 386
KbPerf . . . 399
KbProfile . . . 435
KeyAction . . . 487
QPushButton . . . 628
ColorButton . . . 98
GradientButton . . . 135
QThread . . . 629
Kb . . . 186
MacroReader . . . 576
MacroReaderThread . . . 579
QuaChecksum32 . . . 633
QuaAdler32 . . . 630
QuaCrc32 . . . 636
QuaZip . . . 654
QuaZipDir . . . 686
QuaZipFileInfo . . . 727
QuaZipFileInfo64 . . . 730
QuaZipFilePrivate . . . 738
QuaZipNewInfo . . . 743
QuaZipPrivate . . . 751
QWidget . . . 760
GradientDialogWidget . . . 154
KbAnimWidget . . . 264
KbBindWidget . . . 312
KbLightWidget . . . 362
KbWidget . . . 462
KeyWidget . . . 541
KPerfWidget . . . 562
MPerfWidget . . . 602
RebindWidget . . . 761
SettingsWidget . . . 799
SGroup . . . 811
tm_unz_s . . . 1038
tm_zip_s . . . 1114
unz64_file_pos_s . . . 1038
unz64_s . . . 986
unz_file_info64_internal_s . . . 986
unz_file_info64_s . . . 1038
unz_file_info_s . . . 1038
unz_file_pos_s . . . 1038
unz_global_info64_s . . . 1038
unz_global_info_s . . . 1038
UsbId . . . 814
zip64_internal . . . 1058
zip_fileinfo . . . 1114
zlib_filefunc64_32_def_s . . . 939
zlib_filefunc64_def_s . . . 939

zlib_filefunc_def_s 939

Chapter 5

Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

AnimAddDialog . . . 19
AnimScript . . . 25
AnimSettingDialog . . . 57
AutoRun . . . 77
CkbSettings . . . 81
CkbSettingsWriter . . . 96
ColorButton . . . 98
ColorMap . . . 107
ExtraSettingsWidget . . . 116
KbFirmware::FW . . . 125
FwUpgradeDialog . . . 126
GradientButton . . . 135
GradientDialog . . . 141
GradientDialogWidget . . . 154
JlCompress
Utility class for typical operations . . . 169
Kb . . . 186
KbAnim . . . 236
KbAnimWidget . . . 264
KbBind . . . 282
KbBindWidget . . . 312
KbFirmware . . . 321
KbLight . . . 332
KbLightWidget . . . 362
KbManager . . . 373
KbMode . . . 386
KbPerf . . . 399
KbProfile . . . 435
KbProfileDialog . . . 453
KbWidget . . . 462
Key . . . 484
KeyAction . . . 487
KeyMap . . . 517
KeyWidget . . . 541
KPerfWidget . . . 562
LayoutDialog . . . 573

MacroReader
Creates a worker thread object. It does a connect do delayed deletion of thread local variables in the case the worker thread terminates . . . 576
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 . . . 579
MainWindow . . . 583
ModeSelectDialog . . . 598
MPerfWidget . . . 602
GradientDialog::Preset . . . 620
QDialog . . . 622
QIODevice . . . 623
QIODevice_descriptor . . . 624
QListWidget . . . 625
QMainWindow . . . 626
QObject . . . 627
QPushButton . . . 628
QThread . . . 629
QuaAdler32
Adler32 checksum . . . 630
QuaChecksum32
Checksum interface . . . 633
QuaCrc32
CRC32 checksum . . . 636
QuaGzipFile
GZIP file . . . 639
QuaZIODevice
A class to compress/decompress QIODevice . . . 646
QuaZip
ZIP archive . . . 654
QuaZipDir
Provides ZIP archive navigation . . . 686
QuaZipFile
A file inside ZIP archive . . . 704
QuaZipFileInfo
Information about a file inside archive . . . 727
QuaZipFileInfo64
Information about a file inside archive (with zip64 support) . . . 730
QuaZipFilePrivate
The implementation class for QuaZip . . . 738
QuaZipNewInfo
Information about a file to be created . . . 743
QuaZipPrivate
All the internal stuff for the QuaZip class . . . 751
QWidget . . . 760
RebindWidget . . . 761
RListWidget . . . 794
SettingsWidget . . . 799
SGroup . . . 811
UsbId . . . 814

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

src/ckb/animadddialog.cpp . . . 819
src/ckb/animadddialog.h . . . 819
src/ckb/animscript.cpp . . . 820
src/ckb/animscript.h . . . 822
src/ckb/animsettingdialog.cpp . . . 823
src/ckb/animsettingdialog.h . . . 825
src/ckb/autorun.cpp . . . 826
src/ckb/autorun.h . . . 827
src/ckb/ckb-anim.h . . . 827
src/ckb/ckbsettings.cpp . . . 845
src/ckb/ckbsettings.h . . . 848
src/ckb/ckbsettingswriter.cpp . . . 849
src/ckb/ckbsettingswriter.h . . . 851
src/ckb/colorbutton.cpp . . . 852
src/ckb/colorbutton.h . . . 852
src/ckb/colormap.cpp . . . 853
src/ckb/colormap.h . . . 854
src/ckb/extrasettingswidget.cpp . . . 855
src/ckb/extrasettingswidget.h . . . 856
src/ckb/fwupgradedialog.cpp . . . 857
src/ckb/fwupgradedialog.h . . . 860
src/ckb/gradientbutton.cpp . . . 861
src/ckb/gradientbutton.h . . . 861
src/ckb/gradientdialog.cpp . . . 862
src/ckb/gradientdialog.h . . . 863
src/ckb/gradientdialogwidget.cpp . . . 864
src/ckb/gradientdialogwidget.h . . . 864
src/ckb/kb.cpp . . . 865
src/ckb/kb.h . . . 866
src/ckb/kbanim.cpp . . . 867
src/ckb/kbanim.h . . . 869
src/ckb/kbanimwidget.cpp . . . 870
src/ckb/kbanimwidget.h . . . 870
src/ckb/kbbind.cpp . . . 871
src/ckb/kbbind.h . . . 872
src/ckb/kbbindwidget.cpp . . . 873
src/ckb/kbbindwidget.h . . . 873
src/ckb/kbfirmware.cpp . . . 874

src/ckb/kbfirmware.h . . . 875
src/ckb/kblight.cpp . . . 876
src/ckb/kblight.h . . . 879
src/ckb/kblightwidget.cpp . . . 880
src/ckb/kblightwidget.h . . . 880
src/ckb/kbmanager.cpp . . . 881
src/ckb/kbmanager.h . . . 882
src/ckb/kbmode.cpp . . . 883
src/ckb/kbmode.h . . . 884
src/ckb/kbperf.cpp . . . 885
src/ckb/kbperf.h . . . 885
src/ckb/kbprofile.cpp . . . 886
src/ckb/kbprofile.h . . . 887
src/ckb/kbprofiledialog.cpp . . . 887
src/ckb/kbprofiledialog.h . . . 888
src/ckb/kbwidget.cpp . . . 889
src/ckb/kbwidget.h . . . 890
src/ckb/keyaction.cpp . . . 891
src/ckb/keyaction.h . . . 892
src/ckb/keymap.cpp . . . 892
src/ckb/keymap.h . . . 907
src/ckb/keywidget.cpp . . . 907
src/ckb/keywidget.h . . . 909
src/ckb/kperfwidget.cpp . . . 910
src/ckb/kperfwidget.h . . . 911
src/ckb/layoutdialog.cpp . . . 913
src/ckb/layoutdialog.h . . . 914
src/ckb/macroreader.cpp . . . 915
src/ckb/macroreader.h . . . 915
src/ckb/main.cpp . . . 916
src/ckb/mainwindow.cpp . . . 923
src/ckb/mainwindow.h . . . 923
src/ckb/media.h . . . 925
src/ckb/media_linux.cpp . . . 926
src/ckb/media_mac.m . . . 928
src/ckb/modeselectdialog.cpp . . . 929
src/ckb/modeselectdialog.h . . . 930
src/ckb/mperfwidget.cpp . . . 931
src/ckb/mperfwidget.h . . . 933
src/ckb/rebindwidget.cpp . . . 1125
src/ckb/rebindwidget.h . . . 1126
src/ckb/rlistwidget.cpp . . . 1126
src/ckb/rlistwidget.h . . . 1127
src/ckb/settingswidget.cpp . . . 1128
src/ckb/settingswidget.h . . . 1130
src/ckb/quazip/crypt.h . . . 935
src/ckb/quazip/ioapi.h . . . 939
src/ckb/quazip/JlCompress.cpp . . . 946
src/ckb/quazip/JlCompress.h . . . 948
src/ckb/quazip/qioapi.cpp . . . 948
src/ckb/quazip/quaadler32.cpp . . . 960
src/ckb/quazip/quaadler32.h . . . 961
src/ckb/quazip/quachecksum32.h . . . 962
src/ckb/quazip/quacrc32.cpp . . . 963
src/ckb/quazip/quacrc32.h . . . 964
src/ckb/quazip/quagzipfile.cpp . . . 965
src/ckb/quazip/quagzipfile.h . . . 966
src/ckb/quazip/quaziodevice.cpp . . . 967

src/ckb/quazip/quaziodevice.h . . . 968
src/ckb/quazip/quazip.cpp . . . 969
src/ckb/quazip/quazip.h . . . 971
src/ckb/quazip/quazip_global.h . . . 972
src/ckb/quazip/quazipdir.cpp . . . 973
src/ckb/quazip/quazipdir.h . . . 975
src/ckb/quazip/quazipfile.cpp . . . 976
src/ckb/quazip/quazipfile.h . . . 977
src/ckb/quazip/quazipfileinfo.cpp . . . 978
src/ckb/quazip/quazipfileinfo.h . . . 980
src/ckb/quazip/quazipnewinfo.cpp . . . 981
src/ckb/quazip/quazipnewinfo.h . . . 985
src/ckb/quazip/unzip.c . . . 986
src/ckb/quazip/unzip.h . . . 1038
src/ckb/quazip/zip.c . . . 1058
src/ckb/quazip/zip.h . . . 1114

Chapter 7

Namespace Documentation

7.1 Ui Namespace Reference

Chapter 8

Data Structure Documentation

8.1 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

8.1.1 Detailed Description

Definition at line 12 of file animadddialog.h.

8.1.2 Constructor & Destructor Documentation

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

Definition at line 4 of file animadddialog.cpp.

References AnimScript::list(), AnimScript::name(), on_animBox_activated(), scripts, 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:

8.1.2.2 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:

8.1.3 Member Function Documentation

8.1.3.1 int AnimAddDialog::chosenPreset ( )

Definition at line 24 of file animadddialog.cpp.

References ui.

Referenced by KbLightWidget::on_animButton_clicked().

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

Here is the caller graph for this function:

8.1.3.2 const AnimScript * AnimAddDialog::chosenScript ( )

Definition at line 20 of file animadddialog.cpp.

References scripts, and ui.

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

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

Here is the caller graph for this function:

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

Definition at line 28 of file animadddialog.cpp.

References chosenScript(), on_presetBox_activated(), and ui.

Referenced by AnimAddDialog().

```
{
    // Update preset list
    ui->presetBox->clear();
    foreach(const QString& preset, chosenScript()->presets())
        ui->presetBox->addItem(preset);
    ui->presetBox->setCurrentIndex(0);
    on_presetBox_activated(0);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.1.3.4 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:

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

Definition at line 43 of file animadddialog.cpp.

References _light, on_presetBox_activated(), 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:

8.1.4 Field Documentation

8.1.4.1 QStringList AnimAddDialog::_keys [private]

Definition at line 33 of file animadddialog.h.

Referenced by on_presetBox_activated().

8.1.4.2 KbLight* AnimAddDialog::_light [private]

Definition at line 32 of file animadddialog.h.

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

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

Definition at line 30 of file animadddialog.h.

Referenced by AnimAddDialog(), and chosenScript().

8.1.4.4 bool AnimAddDialog::showPreview [private]

Definition at line 34 of file animadddialog.h.

Referenced by on_presetBox_activated(), and on_previewBox_clicked().

8.1.4.5 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

8.2 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

8.2.1 Detailed Description

Definition at line 16 of file animscript.h.

8.2.2 Data Structure Documentation

8.2.2.1 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	

8.2.3 Member Typedef Documentation

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

Definition at line 58 of file animscript.h.

8.2.4 Constructor & Destructor Documentation

8.2.4.1 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  }
```

8.2.4.2 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:

8.2.4.3 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 }
```

8.2.5 Member Function Documentation

8.2.5.1 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:

8.2.5.2 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:

8.2.5.3 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:

8.2.5.4 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:

8.2.5.5 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:

8.2.5.6 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:

8.2.5.7 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:

8.2.5.8 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:

8.2.5.9 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();
```

```
        connect(process, SIGNAL(finished(int)), process, SLOT(deleteLater()));
        disconnect(process, SIGNAL(readyRead()), this, SLOT(readProcess()));
        process = 0;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.2.5.10 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:

8.2.5.11 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:

8.2.5.12 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:

8.2.5.13 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:

8.2.5.14 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:

8.2.5.15 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:

8.2.5.16 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());
```

```
        break;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.2.5.17 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:

8.2.5.18 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:

8.2.5.19 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, AnimScript::Param::LABEL, AnimScript::Param::LONG, name(), ONE_DAY, param(), preset(), AnimScript::Param::RGB, stop(), AnimScript::Param::STRING, and urlParam().

Referenced by scan().

```
                    {
    // Run the process to get script info
    QProcess infoProcess;
    infoProcess.start(_path, QStringList("--ckb-info"));
    qDebug() << "Scanning " << _path;
    infoProcess.waitForFinished(1000);
    if(infoProcess.state() == QProcess::Running){
        // Kill the process if it takes more than 1s
        infoProcess.kill();
        return false;
    }
    // Set defaults for performance info
    _info.kpMode = KP_NONE;
    _info.absoluteTime = _info.preempt = _info.liveParams = false;
    _info.repeat = true;
    // Read output
    QString line;
    while((line = infoProcess.readLine()) != ""){
        line = line.trimmed();
        QStringList components = line.split(" ");
        int count = components.count();
        if(count < 2)
            continue;
        QString param = components[0].trimmed();
        if(param == "guid")
            _info.guid = QUuid(urlParam(components[1]));
        else if(param == "name")
            _info.name = urlParam(components[1]);
        else if(param == "version")
            _info.version = urlParam(components[1]);
        else if(param == "year")
            _info.year = urlParam(components[1]);
        else if(param == "author")
            _info.author = urlParam(components[1]);
        else if(param == "license")
            _info.license = urlParam(components[1]);
        else if(param == "description")
            _info.description = urlParam(components[1]);
        else if(param == "kpmode")
            _info.kpMode = (components[1] == "position") ? KP_POSITION : (components[1] ==
"name") ? KP_NAME : KP_NONE;
        else if(param == "time")
            _info.absoluteTime = (components[1] == "absolute");
        else if(param == "repeat")
            _info.repeat = (components[1] == "on");
        else if(param == "preempt")
            _info.preempt = (components[1] == "on");
        else if(param == "parammode")
            _info.liveParams = (components[1] == "live");
        else if(param == "param"){
            // Read parameter
            if(count < 3)
                continue;
            while(components.count() < 8)
                components.append("");
            Param::Type type = Param::INVALID;
            QString sType = components[1].toLower();
            if(sType == "long")
                type = Param::LONG;
            else if(sType == "double")
                type = Param::DOUBLE;
            else if(sType == "bool")
                type = Param::BOOL;
            else if(sType == "rgb")
                type = Param::RGB;
            else if(sType == "argb")
                type = Param::ARGB;
            else if(sType == "gradient")
                type = Param::GRADIENT;
            else if(sType == "agradient")
                type = Param::AGRADIENT;
            else if(sType == "angle")
                type = Param::ANGLE;
            else if(sType == "string")
                type = Param::STRING;
            else if(sType == "label")
                type = Param::LABEL;
            else
                continue;
            // "param <type> <name> <prefix> <postfix> <default>"
            QString name = components[2].toLower();
            // Make sure it's not present already
            if(hasParam(name))
                continue;
            QString prefix = urlParam(components[3]), postfix = urlParam(components[4]);
            QVariant def = urlParam(components[5]), minimum = urlParam(components[6]),
maximum = urlParam(components[7]);
```

```
            // Don't allow predefined params
            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")
                continue;
            Param param = { type, name, prefix, postfix, def, minimum, maximum };
            _info.params.append(param);
        } else if(param == "preset"){
            // Add preset
            QString name = urlParam(components.at(1));
            QMap<QString, QVariant> preset;
            for(int i = 2; i < count; i++){
                // Scan name/value setting pairs
                QString setting = components.at(i);
                QStringList sComponents = setting.split("=");
                if(sComponents.count() != 2)
                    continue;
                QString param = sComponents.first().trimmed();
                QString value = urlParam(sComponents.last());
                preset[param] = value;
            }
            // If the preset contains a duration, set the repeat time to the same value
            if(preset.contains("duration")){
                QVariant duration = preset.value("duration");
                preset["repeat"] = duration;
                preset["kprepeat"] = duration;
            }
            _presets.append(name);
            _presetValues.append(preset);
        }
    }
    // Make sure the required parameters are filled out
    if(_info.guid.isNull() || _info.name == "" || _info.version == "" || _info.year == "" || _info.author == "" || _info.license == "")
        return false;
    // Add timing parameters
    double defaultDuration = -1.;
    if(!_info.absoluteTime){
        defaultDuration = 1.;
        Param duration = { Param::DOUBLE, "duration", "", "", defaultDuration, 0.1, ONE_DAY };
        _info.params.append(duration);
    }
    Param trigger = { Param::BOOL, "trigger", "", "", true, 0, 0 };
    _info.params.append(trigger);
    Param kptrigger = { Param::BOOL, "kptrigger", "", "", false, 0, 0 };
    _info.params.append(kptrigger);
    if(_info.absoluteTime || !_info.repeat)
        _info.preempt = false;
    Param kpmode = { Param::LONG, "kpmode", "", "", 1, 0, 0 };
    if(_info.kpMode)
        kpmode.defaultValue = 0;
    _info.params.append(kpmode);
    Param delay = { Param::DOUBLE, "delay", "", "", 0., 0., ONE_DAY };
    Param kpdelay = { Param::DOUBLE, "kpdelay", "", "", 0., 0., ONE_DAY };
    _info.params.append(delay);
    _info.params.append(kpdelay);
    Param kpmodestop = { Param::BOOL, "kpmodestop", "", "", false, 0, 0 };
    Param kprelease = { Param::BOOL, "kprelease", "", "", false, 0, 0 };
    _info.params.append(kpmodestop);
    _info.params.append(kprelease);
    if(_info.repeat){
        Param repeat = { Param::DOUBLE, "repeat", "", "", defaultDuration, 0.1, ONE_DAY };
        Param kprepeat = { Param::DOUBLE, "kprepeat", "", "", defaultDuration, 0.1, ONE_DAY };
        // When repeats are enabled, stop and kpstop are LONG values (number of repeats)
        Param stop = { Param::LONG, "stop", "", "", -1, 0, 1000 };
        Param kpstop = { Param::LONG, "kpstop", "", "", 0, 0, 1000 };
        _info.params.append(repeat);
        _info.params.append(kprepeat);
        _info.params.append(stop);
        _info.params.append(kpstop);
    } else {
        // When repeats are disabled, stop and kpstop are DOUBLE values (seconds)
        Param stop = { Param::DOUBLE, "stop", "", "", -1., 0.1, ONE_DAY };
        Param kpstop = { Param::DOUBLE, "kpstop", "", "", -1., 0.1, ONE_DAY };
        _info.params.append(stop);
        _info.params.append(kpstop);
    }
    return true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.2.5.20 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:

8.2.5.21 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:

8.2.5.22 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:

8.2.5.23 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:

8.2.5.24 QString AnimScript::path ( ) [static]

Definition at line 29 of file animscript.cpp.

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

```
                        {
#ifdef __APPLE__
    return QDir(QApplication::applicationDirPath() + "/../Resources").absoluteFilePath("ckb-animations");
#else
    return QDir("/usr/lib").absoluteFilePath("ckb-animations");
#endif
}
```

Here is the caller graph for this function:

8.2.5.25 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:

8.2.5.26 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:

8.2.5.27 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:

8.2.5.28 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
```

```
            memcpy(_colors.colors(), _colorBuffer.colors(), sizeof(QRgb) * _colors.count());
            inFrame = false;
            readFrame = readAnyFrame = true;
        }
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.2.5.29 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:

8.2.5.30 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:

8.2.5.31 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:

8.2.5.32 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:

8.2.5.33 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:

8.2.5.34 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:

8.2.6 Field Documentation

8.2.6.1 ColorMap AnimScript::_colorBuffer [private]

Definition at line 143 of file animscript.h.

Referenced by init(), and readProcess().

8.2.6.2 ColorMap AnimScript::_colors [private]

Definition at line 135 of file animscript.h.

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

8.2.6.3 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().

8.2.6.4 QStringList AnimScript::_keys [private]

Definition at line 133 of file animscript.h.

Referenced by begin(), and init().

8.2.6.5 KeyMap AnimScript::_map [private]

Definition at line 130 of file animscript.h.

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

8.2.6.6 PresetValue AnimScript::_paramValues [private]

Definition at line 136 of file animscript.h.

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

8.2.6.7 QString AnimScript::_path [private]

Definition at line 128 of file animscript.h.

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

8.2.6.8 QStringList AnimScript::_presets [private]

Definition at line 125 of file animscript.h.

Referenced by load(), and presets().

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

Definition at line 126 of file animscript.h.

Referenced by load(), and preset().

8.2.6.10 int AnimScript::durationMsec [private]

Definition at line 140 of file animscript.h.

Referenced by advance(), and setDuration().

8.2.6.11 bool AnimScript::firstFrame [private]

Definition at line 141 of file animscript.h.

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

8.2.6.12 bool AnimScript::inFrame [private]

Definition at line 141 of file animscript.h.

Referenced by begin(), and readProcess().

8.2.6.13 bool AnimScript::initialized [private]

Definition at line 141 of file animscript.h.

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

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

Definition at line 124 of file animscript.h.

Referenced by keypress(), and load().

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

Definition at line 124 of file animscript.h.

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

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

Definition at line 124 of file animscript.h.

Referenced by keypress(), and load().

8.2.6.17 quint64 AnimScript::lastFrame [private]

Definition at line 139 of file animscript.h.

Referenced by advance(), and begin().

8.2.6.18 int AnimScript::minX [private]

Definition at line 131 of file animscript.h.

Referenced by begin(), and keypress().

8.2.6.19 int AnimScript::minY [private]

Definition at line 131 of file animscript.h.

Referenced by begin(), and keypress().

8.2.6.20 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().

8.2.6.21 bool AnimScript::readAnyFrame [private]

Definition at line 141 of file animscript.h.

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

8.2.6.22 bool AnimScript::readFrame [private]

Definition at line 141 of file animscript.h.

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

8.2.6.23 int AnimScript::repeatMsec [private]

Definition at line 140 of file animscript.h.

Referenced by retrigger(), and setDuration().

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

Definition at line 152 of file animscript.h.

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

8.2.6.25 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

8.3 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

8.3.1 Detailed Description

Definition at line 15 of file animsettingdialog.h.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 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, AnimScript::Param::ARGB, AnimScript::author(), AnimScript::Param::BOOL, AnimScript::description(), AnimScript::Param::DOUBLE, AnimScript::Param::GRADIENT, AnimScript::hasKeypress(), AnimScript::hasParam(), KbAnim::hasParameter(), hasRepeat, hLine(), 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, AnimScript::Param::RGB, KbAnim::script(), KbAnim::scriptName(), settingWidgets, stopCheck, AnimScript::Param::STRING, 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"
```

```
                || param.name == "stop" || param.name == "kpstop"
                || param.name == "kpmodestop" || param.name == "kprelease")
            continue;
        QVariant value = anim->parameter(param.name);
        // Display prefix label on the left (except for booleans and labels)
        if(param.type != AnimScript::Param::BOOL && param.type != AnimScript::Param::LABEL)
            ui->settingsGrid->addWidget(new QLabel(param.prefix, this), row, 1);
        // Configure and display main widget
        QWidget* widget = 0;
        int colSpan = 1;
        QString postfix = param.postfix;
        switch(param.type){
        case AnimScript::Param::BOOL:
            widget = new QCheckBox(param.prefix, this);
            ((QCheckBox*)widget)->setChecked(value.toBool());
            colSpan = 4;
            connect(widget, SIGNAL(stateChanged(int)), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::LONG:
            widget = new QSpinBox(this);
            ((QSpinBox*)widget)->setMinimum(param.minimum.toInt());
            ((QSpinBox*)widget)->setMaximum(param.maximum.toInt());
            ((QSpinBox*)widget)->setValue(value.toInt());
            if(postfix.length() <= 3){
                ((QSpinBox*)widget)->setSuffix(postfix);
                postfix = "";
            }
            connect(widget, SIGNAL(valueChanged(int)), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::DOUBLE:
            widget = new QDoubleSpinBox(this);
            ((QDoubleSpinBox*)widget)->setDecimals(1);
            ((QDoubleSpinBox*)widget)->setMinimum(param.minimum.toDouble());
            ((QDoubleSpinBox*)widget)->setMaximum(param.maximum.toDouble());
            ((QDoubleSpinBox*)widget)->setValue(value.toDouble());
            if(postfix.length() <= 3){
                ((QDoubleSpinBox*)widget)->setSuffix(postfix);
                postfix = "";
            }
            connect(widget, SIGNAL(valueChanged(double)), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::RGB:
            widget = new ColorButton(this);
            ((ColorButton*)widget)->color(QColor("#" + value.toString()));
            colSpan = 3;
            connect(widget, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::ARGB:{
            widget = new ColorButton(this, true);
            QString val = value.toString();
            QColor color;
            if(val.length() == 8){
                color = "#" + val.right(6);
                color.setAlpha(val.left(2).toInt(0, 16));
            } else
                color = "#" + val;
            ((ColorButton*)widget)->color(color);
            colSpan = 3;
            connect(widget, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
            break;
        }
        case AnimScript::Param::GRADIENT:
            widget = new GradientButton(this);
            ((GradientButton*)widget)->fromString(value.toString());
            colSpan = 3;
            connect(widget, SIGNAL(gradientChanged()), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::AGRADIENT:
            widget = new GradientButton(this, true);
            ((GradientButton*)widget)->fromString(value.toString());
            colSpan = 3;
            connect(widget, SIGNAL(gradientChanged()), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::ANGLE:
            widget = new QDial(this);
            ((QDial*)widget)->setFixedSize(60, 60);
            ((QDial*)widget)->setMinimum(0);
            ((QDial*)widget)->setMaximum(360);
            // this is NOT a typo...     ^
            // Even though 360 shouldn't be valid, the wheel/arrow keys will lose 1 degree at the bottom unless 360 is set as max
            ((QDial*)widget)->setSingleStep(5);
            ((QDial*)widget)->setPageStep(15);
            ((QDial*)widget)->setNotchTarget(6);
            ((QDial*)widget)->setNotchesVisible(true);
            ((QDial*)widget)->setWrapping(true);
```

```
            ((QDial*)widget)->setInvertedAppearance(true);
            ((QDial*)widget)->setValue(angleFlip(value.toInt()));
            angleDialMapper.setMapping(widget, param.name);
            connect(widget, SIGNAL(valueChanged(int)), &angleDialMapper, SLOT(map()));
            break;
        case AnimScript::Param::STRING:
            widget = new QLineEdit(this);
            ((QLineEdit*)widget)->setText(value.toString());
            colSpan = 3;
            connect(widget, SIGNAL(textEdited(const QString&)), &updateMapper, SLOT(map()));
            break;
        case AnimScript::Param::LABEL:
            widget = new QLabel(this);
            ((QLabel*)widget)->setText(param.prefix);
            colSpan = 4;
            break;
        default:
            break;
        }
        if(widget)
            updateMapper.setMapping(widget, param.name);
        if(param.type == AnimScript::Param::BOOL || param.type == AnimScript::Param::LABEL){
            // Boolean values are placed on the left with no prefix or postfix
            settingWidgets[param.name] = widget;
            ui->settingsGrid->addWidget(widget, row, 3, 1, colSpan);
        } else {
            // Display the widget
            if(widget){
                settingWidgets[param.name] = widget;
                ui->settingsGrid->addWidget(widget, row, 3, 1, colSpan);
            }
            // Angles additionally have a spin box
            if(param.type == AnimScript::Param::ANGLE){
                QSpinBox* spinner = new QSpinBox(this);
                spinner->setMinimum(0);
                spinner->setMaximum(359);
                spinner->setWrapping(true);
                spinner->setSuffix("°");
                spinner->setValue(value.toInt());
                angleSpinners[param.name] = spinner;
                angleSpinnerMapper.setMapping(spinner, param.name);
                connect(spinner, SIGNAL(valueChanged(int)), &angleSpinnerMapper, SLOT(map()));
                ui->settingsGrid->addWidget(spinner, row, 4);
                colSpan = 2;
            }
            // Display postfix label on the right
            ui->settingsGrid->addWidget(new QLabel(postfix, this), row, 3 + colSpan, 1, 4 - colSpan);
            if(colSpan < 3 && !rSpacePlaced){
                // Add a spacer to compress short elements to the left
                ui->settingsGrid->addItem(new QSpacerItem(0, 0, QSizePolicy::Minimum), row, 4 + colSpan);
                rSpacePlaced = true;
            }
        }
        row++;
    }
    // Add playback info at bottom
    ui->settingsGrid->addItem(new QSpacerItem(0, 10, QSizePolicy::Fixed, QSizePolicy::Fixed), row++, 6);
    ui->settingsGrid->addWidget(new QLabel("<b>Playback</b>", this), row++, 0, 1, 7);
    ui->settingsGrid->addWidget(hLine(), row++, 0, 1, 7);
    if(script->hasParam("duration")){
        // Show duration spinner (if allowed)
        lastDuration = anim->parameter("duration").toDouble();
        ui->settingsGrid->addWidget(new QLabel("Duration:", this), row, 1);
        QDoubleSpinBox* spinner = new QDoubleSpinBox(this);
        spinner->setDecimals(1);
        spinner->setMinimum(0.1);
        spinner->setValue(lastDuration);
        connect(spinner, SIGNAL(valueChanged(double)), this, SLOT(newDuration(double)));
        settingWidgets["duration"] = spinner;
        ui->settingsGrid->addWidget(spinner, row, 3, 1, 1);
        ui->settingsGrid->addWidget(new QLabel("seconds", this), row, 4, 1, 2);
        row++;
    }
    // Show boxes for start with mode/with keypress
    QCheckBox* check = new QCheckBox("Start with mode", this);
    check->setChecked(anim->parameter("trigger").toBool());
    ui->settingsGrid->addWidget(check, row, 3, 1, 4);
    settingWidgets["trigger"] = check;
    connect(check, SIGNAL(stateChanged(int)), &updateMapper, SLOT(map()));
    updateMapper.setMapping(check, "trigger");
    row++;
    check = new QCheckBox("Start with key press", this);
    check->setChecked(anim->parameter("kptrigger").toBool());
    ui->settingsGrid->addWidget(check, row, 3, 1, 2);
```

```
    settingWidgets["kptrigger"] = check;
    connect(check, SIGNAL(stateChanged(int)), &updateMapper, SLOT(map()));
    updateMapper.setMapping(check, "kptrigger");
    // Add an option allowing the user to select keypress mode
    QComboBox* combo = new QComboBox(this);
    int selected = anim->parameter("kpmode").toInt();
    if(script->hasKeypress()){
        // If the script supports keypresses, show the option to handle them that way (default)
        combo->addItem("on pressed key");
        combo->addItem("on whole keyboard");
        combo->addItem("on keyboard (once)");
    } else {
        selected--;
        // Otherwise, just show the choice of whether to start it every time or just once
        combo->addItem("every time");
        combo->addItem("only once");
    }
    if(selected < 0 || selected > combo->count())
        selected = 0;
    combo->setCurrentIndex(selected);
    ui->settingsGrid->addWidget(combo, row, 5, 1, 2);
    settingWidgets["kpmode"] = combo;
    connect(combo, SIGNAL(activated(int)), &updateMapper, SLOT(map()));
    updateMapper.setMapping(combo, "kpmode");
    row++;

    // Add horizontal spacer to compress content to left
    ui->settingsGrid->addItem(new QSpacerItem(0, 0, QSizePolicy::Expanding), 0, 6);
    // Add vertical spacer to compress content to top
    ui->settingsGrid->addItem(new QSpacerItem(0, 0, QSizePolicy::Minimum, QSizePolicy::Expanding), row, 0
);

    // Add timing fields
    settingWidgets["delay"] = ui->delayBox;
    ui->delayBox->setValue(anim->parameter("delay").toDouble());
    settingWidgets["kpdelay"] = ui->kpDelayBox;
    ui->kpDelayBox->setValue(anim->parameter("kpdelay").toDouble());
    settingWidgets["kpmodestop"] = ui->kpModeStopBox;
    ui->kpModeStopBox->setChecked(anim->parameter("kpmodestop").toBool());
    connect(ui->kpModeStopBox, SIGNAL(clicked(bool)), &updateMapper, SLOT(map()));
    updateMapper.setMapping(ui->kpModeStopBox, "kpmodestop");
    settingWidgets["kprelease"] = ui->kpReleaseBox;
    ui->kpReleaseBox->setChecked(anim->parameter("kprelease").toBool());
    if(anim->hasParameter("repeat")){
        hasRepeat = true;
        settingWidgets["repeat"] = ui->repeatBox;
        ui->repeatBox->setValue(anim->parameter("repeat").toDouble());
        settingWidgets["kprepeat"] = ui->kpRepeatBox;
        ui->kpRepeatBox->setValue(anim->parameter("kprepeat").toDouble());
        // If repeat is enabled, add repeat counts as integer values
        // Mode repeat
        ui->timeGrid->addWidget(new QLabel("Repeat:", this), 4, 1);
        QSpinBox* spinner = new QSpinBox(this);
        spinner->setMinimum(0);
        spinner->setMaximum(1000000);
        spinner->setValue(anim->parameter("stop").toInt());
        settingWidgets["stop"] = spinner;
        connect(spinner, SIGNAL(valueChanged(int)), &updateMapper, SLOT(map()));
        updateMapper.setMapping(spinner, "stop");
        ui->timeGrid->addWidget(spinner, 4, 3);
        ui->timeGrid->addWidget(new QLabel("times", this), 4, 4);
        // KP repeat
        ui->timeGrid->addWidget(new QLabel("Repeat:", this), 12, 1);
        spinner = new QSpinBox(this);
        spinner->setMinimum(0);
        spinner->setMaximum(1000000);
        spinner->setValue(anim->parameter("kpstop").toInt());
        settingWidgets["kpstop"] = spinner;
        connect(spinner, SIGNAL(valueChanged(int)), &updateMapper, SLOT(map()));
        updateMapper.setMapping(spinner, "kpstop");
        ui->timeGrid->addWidget(spinner, 12, 3);
        ui->timeGrid->addWidget(new QLabel("times", this), 12, 4);
        // Infinite repeat toggles
        stopCheck = new QCheckBox("Forever", this);
        stopCheck->setChecked(anim->parameter("stop").toInt() < 0);
        ui->timeGrid->addWidget(stopCheck, 4, 5);
        kpStopCheck = new QCheckBox("Forever", this);
        connect(stopCheck, SIGNAL(clicked()), this, SLOT(updateStops()));
        kpStopCheck->setChecked(anim->parameter("kpstop").toInt() < 0);
        ui->timeGrid->addWidget(kpStopCheck, 12, 5);
        connect(kpStopCheck, SIGNAL(clicked()), this, SLOT(
updateStops()));
    } else {
        hasRepeat = false;
        // If repeat is not enabled, hide repeat-related fields
        ui->repeatBox->setHidden(true);
        ui->repeatLabel->setHidden(true);
```

```
        ui->repeatLabel_2->setHidden(true);
        ui->kpRepeatBox->setHidden(true);
        ui->kpRepeatLabel->setHidden(true);
        ui->kpRepeatLabel_2->setHidden(true);
        // Add stop times as double values
        // Stop time
        QDoubleSpinBox* spinner = new QDoubleSpinBox(this);
        spinner->setDecimals(1);
        spinner->setMinimum(0.1);
        spinner->setMaximum(24. * 60. * 60.);
        double stop = anim->parameter("stop").toDouble();
        if(stop <= 0.)
            spinner->setValue(lastDuration);
        else
            spinner->setValue(stop);
        settingWidgets["stop"] = spinner;
        connect(spinner, SIGNAL(valueChanged(double)), &updateMapper, SLOT(map()));
        updateMapper.setMapping(spinner, "stop");
        ui->timeGrid->addWidget(spinner, 4, 3);
        ui->timeGrid->addWidget(new QLabel("seconds", this), 4, 4);
        // KP stop time
        spinner = new QDoubleSpinBox(this);
        spinner->setDecimals(1);
        spinner->setMinimum(0.1);
        spinner->setMaximum(24. * 60. * 60.);
        double kpstop = anim->parameter("kpstop").toDouble();
        if(kpstop <= 0.)
            spinner->setValue(lastDuration);
        else
            spinner->setValue(kpstop);
        settingWidgets["kpstop"] = spinner;
        connect(spinner, SIGNAL(valueChanged(double)), &updateMapper, SLOT(map()));
        updateMapper.setMapping(spinner, "kpstop");
        ui->timeGrid->addWidget(spinner, 12, 3);
        ui->timeGrid->addWidget(new QLabel("seconds", this), 12, 4);
        // Infinite run toggles
        stopCheck = new QCheckBox("Stop after:", this);
        stopCheck->setChecked(stop > 0.);
        ui->timeGrid->addWidget(stopCheck, 4, 1);
        connect(stopCheck, SIGNAL(clicked()), this, SLOT(updateStops()));
        kpStopCheck = new QCheckBox("Stop after:", this);
        kpStopCheck->setChecked(kpstop > 0.);
        ui->timeGrid->addWidget(kpStopCheck, 12, 1);
        connect(kpStopCheck, SIGNAL(clicked()), this, SLOT(updateStops()));
    }
    updateStops();

    // Set script info UI
    ui->sNameLabel->setText(script->name());
    ui->sVerLabel->setText(script->version());
    ui->sAuthorLabel->setText(script->author());
    ui->sYearLabel->setText(script->year());
    ui->sLicenseLabel->setText(script->license());
    ui->sDescLabel->setText(script->description());

    // Lock dialog size
    setFixedSize(minimumSize());
}
```

Here is the call graph for this function:

8.3.2.2 AnimSettingDialog::~AnimSettingDialog ( )

Definition at line 476 of file animsettingdialog.cpp.

References ui.

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

8.3.3 Member Function Documentation

8.3.3.1 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:

8.3.3.2 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                                                 {
```

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

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 480 of file animsettingdialog.cpp.

Referenced by AnimSettingDialog().

```
                                {
    QFrame* frame = new QFrame(this);
    frame->setFrameShape(QFrame::HLine);
    frame->setFrameShadow(QFrame::Sunken);
    return frame;
}
```

Here is the caller graph for this function:

8.3.3.4 QString AnimSettingDialog::name ( ) const

Definition at line 472 of file animsettingdialog.cpp.

References ui.

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

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

Here is the caller graph for this function:

8.3.3.5 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:

8.3.3.6 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:

8.3.3.7 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:

8.3.3.8 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:

8.3.3.9 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:

8.3.3.10 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:

8.3.3.11 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
```

```
    switch(_anim->script()->param(name).type){
    case AnimScript::Param::BOOL:{
        QCheckBox* widget = (QCheckBox*)settingWidgets[name];
        _anim->parameter(name, (int)widget->isChecked());
        break;
    }
    case AnimScript::Param::LONG:{
        QSpinBox* widget = (QSpinBox*)settingWidgets[name];
        _anim->parameter(name, widget->value());
        break;
    }
    case AnimScript::Param::DOUBLE:{
        QDoubleSpinBox* widget = (QDoubleSpinBox*)settingWidgets[name];
        _anim->parameter(name, widget->value());
        break;
    }
    case AnimScript::Param::RGB:{
        ColorButton* widget = (ColorButton*)settingWidgets[name];
        QColor color = widget->color();
        char hex[7];
        snprintf(hex, sizeof(hex), "%02x%02x%02x", color.red(), color.green(), color.blue());
        _anim->parameter(name, QString(hex));
        break;
    }
    case AnimScript::Param::ARGB:{
        ColorButton* widget = (ColorButton*)settingWidgets[name];
        QColor color = widget->color();
        char hex[9];
        snprintf(hex, sizeof(hex), "%02x%02x%02x%02x", color.alpha(), color.red(), color.green(), color.blue());
        _anim->parameter(name, QString(hex));
        break;
    }
    case AnimScript::Param::GRADIENT:
    case AnimScript::Param::AGRADIENT:{
        GradientButton* widget = (GradientButton*)settingWidgets[name];
        _anim->parameter(name, widget->toString());
        break;
    }
    case AnimScript::Param::ANGLE:{
        QDial* widget = (QDial*)settingWidgets[name];
        _anim->parameter(name, angleFlip(widget->value()));
        break;
    }
    case AnimScript::Param::STRING:{
        QLineEdit* widget = (QLineEdit*)settingWidgets[name];
        _anim->parameter(name, widget->text());
        break;
    }
    default:
        break;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.3.3.12 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:

8.3.4 Field Documentation

8.3.4.1 KbAnim* AnimSettingDialog::_anim [private]

Definition at line 32 of file animsettingdialog.h.

Referenced by updateParam().

8.3.4.2 QSignalMapper AnimSettingDialog::angleDialMapper [private]

Definition at line 38 of file animsettingdialog.h.

Referenced by AnimSettingDialog().

8.3.4.3 QSignalMapper AnimSettingDialog::angleSpinnerMapper [private]

Definition at line 38 of file animsettingdialog.h.

Referenced by AnimSettingDialog().

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

Definition at line 37 of file animsettingdialog.h.

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

8.3.4.5 bool AnimSettingDialog::hasRepeat [private]

Definition at line 30 of file animsettingdialog.h.

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

8.3.4.6 QCheckBox * AnimSettingDialog::kpStopCheck [private]

Definition at line 29 of file animsettingdialog.h.

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

8.3.4.7 double AnimSettingDialog::lastDuration [private]

Definition at line 33 of file animsettingdialog.h.

Referenced by AnimSettingDialog(), and newDuration().

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

Definition at line 34 of file animsettingdialog.h.

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

8.3.4.9 QCheckBox* AnimSettingDialog::stopCheck [private]

Definition at line 29 of file animsettingdialog.h.

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

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

Definition at line 28 of file animsettingdialog.h.

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

8.3.4.11 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

8.4 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 ()

8.4.1 Detailed Description

Definition at line 8 of file autorun.h.

8.4.2 Member Function Documentation

8.4.2.1 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:

8.4.2.2 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:

8.4.2.3 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:

8.4.2.4 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:

8.4.2.5 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

8.5 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

8.5.1 Detailed Description

Definition at line 11 of file ckbsettings.h.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 CkbSettings::CkbSettings ( )

Definition at line 53 of file ckbsettings.cpp.

```
53                          :
54     backing(globalSettings()) {
55 }
```

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

Definition at line 57 of file ckbsettings.cpp.

References beginGroup().

```
57                                                                  :
58     backing(globalSettings()) {
59     if(basePath.isEmpty()){
60         if(eraseExisting)
61             qDebug() << "CkbSettings created with basePath = \"\" and eraseExisting = true. This is a
       mistake.";
62         return;
63     }
64     if(eraseExisting)
65         remove(basePath);
66     beginGroup(basePath);
67 }
```

Here is the call graph for this function:

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

Definition at line 69 of file ckbsettings.cpp.

```
69                                                  :
70      backing(&settings) {
71  }
```

8.5.2.4 CkbSettings::~CkbSettings ( )

Definition at line 137 of file ckbsettings.cpp.

References backing, removeCache, and writeCache.

```
137                                 {
138     if(removeCache.isEmpty() && writeCache.isEmpty())
139         return;
140     // Save the settings from the settings thread.
141     // They have to be saved from that thread specifically to avoid performance issues
142     CkbSettingsWriter* writer = new CkbSettingsWriter(
      backing, removeCache, writeCache);
143     writer->moveToThread(globalThread);
144     QObject::staticMetaObject.invokeMethod(writer, "run", Qt::QueuedConnection);
145 }
```

8.5.3 Member Function Documentation

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

Definition at line 73 of file ckbsettings.cpp.

References groups.

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

```
73                                                  {
74      groups.append(prefix);
75  }
```

Here is the caller graph for this function:

8.5.3.2 QStringList CkbSettings::childGroups ( ) const

Definition at line 81 of file ckbsettings.cpp.

References backing, lockMutex, and pwd().

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

```
81                                          {
82      QString current = pwd();
83      lockMutex;
84      if(!current.isEmpty())
85          backing->beginGroup(current);
86      QStringList res = backing->childGroups();
87      if(!current.isEmpty())
88          backing->endGroup();
89      return res;
90  }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.3.3 QStringList CkbSettings::childKeys ( ) const

Definition at line 92 of file ckbsettings.cpp.

References backing, lockMutex, and pwd().

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

```
92                                          {
93      QString current = pwd();
94      lockMutex;
95      if(!current.isEmpty())
96          backing->beginGroup(current);
97      QStringList res = backing->childKeys();
98      if(!current.isEmpty())
99          backing->endGroup();
100      return res;
101 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.3.4 void CkbSettings::cleanUp ( ) [static]

Definition at line 38 of file ckbsettings.cpp.

References _globalSettings, cacheWritesInProgress(), and globalThread.

Referenced by MainWindow::cleanup().

```
38                          {
39     if(!_globalSettings)
40         return;
41     // Wait for all writers to finish
42     while(cacheWritesInProgress.load() > 0)
43         QThread::yieldCurrentThread();
44     // Stop thread and delete objects
45     globalThread->quit();
46     globalThread->wait();
47     delete globalThread;
48     delete _globalSettings;
49     globalThread = 0;
50     _globalSettings = 0;
51 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 103 of file ckbsettings.cpp.

References backing, lockMutex, and pwd().

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

```
103                                                 {
104     lockMutex;
105     return backing->contains(pwd(key));
106 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 108 of file ckbsettings.cpp.

References childGroups().

Referenced by KbPerf::load().

```
                                                {
    QStringList components = group.split("/");
    if(components.length() > 1){
        // Find sub-group
        SGroup group(*this, components[0]);
        return containsGroup(QStringList(components.mid(1)).join('/'));
    }
    return childGroups().contains(group);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.3.7 void CkbSettings::endGroup ( )

Definition at line 77 of file ckbsettings.cpp.

References groups.

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

```
                        {
    groups.removeLast();
}
```

Here is the caller graph for this function:

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

Definition at line 147 of file ckbsettings.cpp.

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

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

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

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.3.9 bool CkbSettings::isBusy ( ) [static]

Definition at line 34 of file ckbsettings.cpp.

References cacheWritesInProgress().

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

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

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.3.10 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:

8.5.3.11 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:

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

Definition at line 133 of file ckbsettings.cpp.

References pwd(), and removeCache.

Referenced by SettingsWidget::SettingsWidget().

```
133                                         {
134     removeCache.append(pwd(key));
135 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 158 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().

```
{
    {
        lockMutexCache;
        if(globalCache.value(key) == value)
            return;
        globalCache[key] = value;
    }
    lockMutexStatic;
    globalSettings()->setValue(key, value);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 123 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().

```
123                                                                 {
124     // Cache the write values, save them when the object is destroyed
125     QString realKey = pwd(key);
126     writeCache[realKey] = value;
127     // Update global cache if needed
128     lockMutexCache;
129     if(globalCache.contains(realKey))
130         globalCache[realKey] = value;
131 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 118 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().

```
118                                                                 {
119     lockMutex;
120     return backing->value(pwd(key), defaultValue);
121 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.5.4 Field Documentation

8.5.4.1 QSettings* CkbSettings::backing [private]

Definition at line 46 of file ckbsettings.h.

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

8.5.4.2 QStringList CkbSettings::groups [private]

Definition at line 47 of file ckbsettings.h.

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

8.5.4.3 QStringList CkbSettings::removeCache [private]

Definition at line 48 of file ckbsettings.h.

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

8.5.4.4 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

8.6 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

8.6.1 Detailed Description

Definition at line 12 of file ckbsettingswriter.h.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 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 }
```

8.6.2.2 CkbSettingsWriter::~CkbSettingsWriter ( )

Definition at line 17 of file ckbsettingswriter.cpp.

References cacheWritesInProgress.

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

8.6.3 Member Function Documentation

8.6.3.1 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 }
```

8.6.4 Field Documentation

8.6.4.1 QSettings* CkbSettingsWriter::_backing [private]

Definition at line 21 of file ckbsettingswriter.h.

Referenced by run().

8.6.4.2 QStringList CkbSettingsWriter::_removals [private]

Definition at line 22 of file ckbsettingswriter.h.

Referenced by run().

8.6.4.3 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

8.7 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

8.7.1 Detailed Description

Definition at line 6 of file colorbutton.h.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 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:

8.7.3 Member Function Documentation

8.7.3.1 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:

8.7.3.2 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:

8.7.3.3 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:

8.7.3.4 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:

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

Definition at line 13 of file colorbutton.h.

References _color.

Referenced by MPerfWidget::colorChanged(), KPerfWidget::setPerf(), MPerfWidget::setPerf(), KPerfWidget::uiUpdated(), and AnimSettingDialog::updateParam().

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

Here is the caller graph for this function:

8.7.3.6 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:

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

Referenced by pickColor().

Here is the caller graph for this function:

8.7.3.8 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:

8.7.3.9 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:

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

Definition at line 21 of file colorbutton.h.

References _setLabel.

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

8.7.3.11 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:

8.7.4 Field Documentation

8.7.4.1 bool ColorButton::_alpha [private]

Definition at line 34 of file colorbutton.h.

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

8.7.4.2 bool ColorButton::_bigIcons [private]

Definition at line 34 of file colorbutton.h.

Referenced by bigIcons(), and updateImage().

8.7.4.3 QColor ColorButton::_color [private]

Definition at line 33 of file colorbutton.h.

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

8.7.4.4 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

8.8 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)
- 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

8.8.1 Detailed Description

Definition at line 15 of file colormap.h.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 ColorMap::ColorMap ( )

Definition at line 3 of file colormap.cpp.

```
:
    _keyNames(0), _colors(0), _count(0), _mapCount(0)
{
}
```

8.8.2.2 ColorMap::~ColorMap ( )

Definition at line 8 of file colormap.cpp.

References deAlloc().

```
{
    deAlloc();
}
```

Here is the call graph for this function:

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

Definition at line 12 of file colormap.cpp.

```
:
    _keyNames(0), _colors(0), _count(0), _mapCount(0)
{
    *this = rhs;
}
```

8.8.3 Member Function Documentation

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

Definition at line 27 of file colormap.cpp.

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

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

```
27                                  {
28     if(newCount > _mapCount){
29         // ^ map never shrinks, only expands
30         deAlloc();
31         _mapCount = newCount;
32         _keyNames = new const char*[_mapCount];
33         _colors = new QRgb[_mapCount];
34     }
35     _count = newCount;
36     clear();
37 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.8.3.2 void ColorMap::clear ( )

Definition at line 45 of file colormap.cpp.

References _colors, and _count.

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

```
45                          {
46     memset(_colors, 0, _count * sizeof(QRgb));
47 }
```

Here is the caller graph for this function:

8.8.3.3 QRgb ∗ ColorMap::colorForName (const char ∗ *name*)

Definition at line 72 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().

```
72                                                  {
73     if(!_keyNames)
74         return 0;
75     const char** namePtr = (const char**)bsearch(&name, _keyNames,
       _count, sizeof(const char*), qs_strcmp);
76     if(!namePtr)
77         return 0;
78     ptrdiff_t position = namePtr - _keyNames;
79     return _colors + position;
80 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.8.3.4 const QRgb ∗ ColorMap::colorForName (const char ∗ *name*) const

Definition at line 82 of file colormap.cpp.

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

```
82                                                          {
83     if(!_keyNames)
84         return 0;
85     const char** namePtr = (const char**)bsearch(&name, _keyNames,
      _count, sizeof(const char*), qs_strcmp);
86     if(!namePtr)
87         return 0;
88     ptrdiff_t position = namePtr - _keyNames;
89     return _colors + position;
90 }
```

Here is the call graph for this function:

8.8.3.5 QRgb∗ ColorMap::colors ( ) [inline]

Definition at line 32 of file colormap.h.

References _colors.

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

```
32 { return _colors; }
```

Here is the caller graph for this function:

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

Definition at line 33 of file colormap.h.

References _colors.

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

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

Definition at line 30 of file colormap.h.

References _count.

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

```
30 { return _count; }
```

Here is the caller graph for this function:

8.8.3.8 void ColorMap::deAlloc ( ) [private]

Definition at line 39 of file colormap.cpp.

References _colors, _count, _keyNames, and _mapCount.

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

```
                        {
    _count = _mapCount = 0;
    if(_keyNames) delete[] _keyNames;
    if(_colors) delete[] _colors;
}
```

Here is the caller graph for this function:

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

Definition at line 53 of file colormap.cpp.

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

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

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

Here is the call graph for this function:

Here is the caller graph for this function:

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

Definition at line 31 of file colormap.h.

References _keyNames.

Referenced by KbLight::printRGB().

```
31 { return _keyNames; }
```

Here is the caller graph for this function:

8.8.3.11 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:

8.8.4 Field Documentation

8.8.4.1 QRgb* ColorMap::_colors [private]

Definition at line 44 of file colormap.h.

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

8.8.4.2 int ColorMap::_count [private]

Definition at line 45 of file colormap.h.

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

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

Definition at line 43 of file colormap.h.

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

8.8.4.4 int ColorMap::_mapCount [private]

Definition at line 45 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

8.9 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

8.9.1 Detailed Description

Definition at line 10 of file extrasettingswidget.h.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 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 Kb::dither(), Kb::frameRate(), lastSharedDimming, Kb::macroDelay(), KbLight::MAX_DIM, Kb::mouseAccel(), on_animScanButton_clicked(), AnimScript::path(), Kb::scrollSpeed(), KbLight::shareDimming(), 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
68     bool macroDelay = settings.value("MacroDelay").toBool();
69     Kb::macroDelay(macroDelay);
70     ui->delayBox->setChecked(macroDelay);
71 }
```

Here is the call graph for this function:

8.9.2.2 ExtraSettingsWidget::~ExtraSettingsWidget ( )

Definition at line 73 of file extrasettingswidget.cpp.

References ui.

```
73                                                  {
74     delete ui;
75 }
```

8.9.3 Member Function Documentation

8.9.3.1 void ExtraSettingsWidget::on_animScanButton_clicked ( ) [private],[slot]

Definition at line 87 of file extrasettingswidget.cpp.

References 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:

8.9.3.2 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:

8.9.3.3 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:

8.9.3.4 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:

8.9.3.5 void ExtraSettingsWidget::on_fpsBox_valueChanged (int *arg1*) [private], [slot]

Definition at line 98 of file extrasettingswidget.cpp.

References 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:

8.9.3.6 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:

8.9.3.7 void ExtraSettingsWidget::on_sAccelBox_clicked (bool *checked*) [private],[slot]

Definition at line 129 of file extrasettingswidget.cpp.

References Kb::scrollSpeed(), CkbSettings::set(), 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:

8.9.3.8 void ExtraSettingsWidget::on_sSpeedBox_valueChanged (int *arg1*) [private],[slot]

Definition at line 138 of file extrasettingswidget.cpp.

References 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:

8.9.3.9 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:

8.9.3.10 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:

8.9.4 Field Documentation

8.9.4.1 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

8.10 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

8.10.1 Detailed Description

Definition at line 34 of file kbfirmware.h.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 KbFirmware::FW::FW ( )

Definition at line 12 of file kbfirmware.cpp.

```
12 : fwVersion(0.f), ckbVersion(0.f) {}
```

8.10.3 Field Documentation

8.10.3.1 float KbFirmware::FW::ckbVersion

Definition at line 37 of file kbfirmware.h.

Referenced by KbFirmware::_latestForBoard(), and KbFirmware::processDownload().

8.10.3.2 QString KbFirmware::FW::fileName

Definition at line 35 of file kbfirmware.h.

Referenced by KbFirmware::_fileForBoard(), and KbFirmware::processDownload().

8.10.3.3 float KbFirmware::FW::fwVersion

Definition at line 37 of file kbfirmware.h.

Referenced by KbFirmware::_latestForBoard(), and KbFirmware::processDownload().

8.10.3.4 QByteArray KbFirmware::FW::hash

Definition at line 36 of file kbfirmware.h.

Referenced by KbFirmware::_fileForBoard(), KbFirmware::_latestForBoard(), and KbFirmware::processDownload().

8.10.3.5 QString KbFirmware::FW::url

Definition at line 35 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

8.11 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

8.11.1 Detailed Description

Definition at line 13 of file fwupgradedialog.h.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 FwUpgradeDialog::FwUpgradeDialog (QWidget * *parent,* float *newV,* const QByteArray & *fwBlob,* Kb * *device*)

Definition at line 31 of file fwupgradedialog.cpp.

References DIALOG_HEIGHT_MIN, DIALOG_WIDTH, Kb::firmware, fwUpdateFinished(), fwUpdateProgress(), kb, removeDev(), ui, and Kb::usbModel.

```
31                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
:
32     QDialog(parent),
33     ui(new Ui::FwUpgradeDialog),
34     blob(fwBlob), kb(device), evLoop(0), exitSuccess(true)
35 {
36     ui->setupUi(this);
37     ui->curLabel->setText(kb->firmware);
38     ui->newLabel->setText(QString::number(newV, 'f', 2));
39     ui->devLabel->setText(kb->usbModel);
40
41     connect(device, SIGNAL(destroyed()), this, SLOT(removeDev()));
42     connect(device, SIGNAL(fwUpdateProgress(int,int)), this, SLOT(
       fwUpdateProgress(int,int)));
43     connect(device, SIGNAL(fwUpdateFinished(bool)), this, SLOT(
       fwUpdateFinished(bool)));
44
45     setFixedSize(DIALOG_WIDTH, DIALOG_HEIGHT_MIN);
46 }
```

Here is the call graph for this function:

8.11.2.2 FwUpgradeDialog::~FwUpgradeDialog ( )

Definition at line 48 of file fwupgradedialog.cpp.

References cleanBlob(), and ui.

```
48                                  {
49     cleanBlob();
50     delete ui;
51 }
```

Here is the call graph for this function:

8.11.3 Member Function Documentation

8.11.3.1 void FwUpgradeDialog::cleanBlob ( ) [private]

Definition at line 77 of file fwupgradedialog.cpp.

References savePath.

Referenced by fwUpdateFinished(), and ~FwUpgradeDialog().

```
77                                  {
78      if(savePath.isEmpty())
79          return;
80      QFile(savePath).remove();
81      savePath = "";
82  }
```

Here is the caller graph for this function:

8.11.3.2 void FwUpgradeDialog::closeEvent (QCloseEvent * *event*) [private]

Definition at line 53 of file fwupgradedialog.cpp.

```
53                                                  {
54      event->ignore();
55  }
```

8.11.3.3 int FwUpgradeDialog::exec ( )

Definition at line 107 of file fwupgradedialog.cpp.

References blob, KbFirmware::dataForBoard(), DIALOG_HEIGHT_MAX(), DIALOG_WIDTH, evLoop, exitSuccess, Kb::features, kb, saveBlob(), ui, and verifyFw().

Referenced by KbWidget::on_fwUpdButton_clicked().

```
107                             {
108      QString features = kb->features;
109      if(!blob.isEmpty()){
110          // If a blob was already specified, check its version and validity
111          float newV = verifyFw(blob, features);
112          if(newV == 0.f){
113              QMessageBox::warning(parentWidget(), "Error", "<center>Not a valid firmware for this
     device.</center>");
114              return QDialog::Rejected;
115          }
116          ui->newLabel->setText(QString::number(newV, 'f', 2));
117      } else {
118          // Download a new blob file
119          ui->progressBar->show();
120          ui->cancelButton->setEnabled(false);
121          ui->actionButton->setEnabled(false);
122          show();
123          // This can take a while
124          blob = KbFirmware::dataForBoard(features);
125          // Check validity
126          float newV = verifyFw(blob, features);
127          if(newV == 0.f){
128              hide();
129              QMessageBox::warning(parentWidget(), "Error", "<center>There was a problem with the downloaded
     file.<br />Please try again later.</center>");
130              return QDialog::Rejected;
```

```
        }
    }
    // Save temporary file
    if(saveBlob().isEmpty()){
        hide();
        QMessageBox::warning(parentWidget(), "Error", "<center>Unable to save temporary file.</center>");
        return QDialog::Rejected;
    }
    // Set up UI
    ui->progressBar->setValue(0);
    ui->progressBar->setMaximum(1);
    ui->progressBar->setTextVisible(false);
    ui->statusLabel->setText("Ready to install new firmware.<br /><br /><b>Disclaimer:</b> ckb is not
endorsed by Corsair. This is <i>unlikely</i> to brick your device, but I accept no responsibility if it does.
If you're paranoid, update from Windows.");
    ui->cancelButton->setEnabled(true);
    ui->actionButton->setEnabled(true);
    setFixedSize(DIALOG_WIDTH, DIALOG_HEIGHT_MAX);
    show();
    // Run modal event loop
    evLoop = new QEventLoop(this);
    evLoop->exec();
    delete evLoop;
    hide();
    return exitSuccess ? QDialog::Accepted : QDialog::Rejected;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.11.3.4 void FwUpgradeDialog::fwUpdateFinished (bool *succeeded*) [private],[slot]

Definition at line 170 of file fwupgradedialog.cpp.

References cleanBlob(), evLoop, and ui.

Referenced by FwUpgradeDialog(), and removeDev().

```
170                                                 {
171     cleanBlob();
172     if(succeeded)
173         ui->statusLabel->setText("Update successful!");
174     else
```

```
        ui->statusLabel->setText("Update failed.");
    ui->actionButton->setText("OK");
    ui->actionButton->setEnabled(true);
    ui->progressBar->setMaximum(1);
    ui->progressBar->setValue(1);
    // Exit after 10s
    if(evLoop)
        QTimer::singleShot(10000, evLoop, SLOT(quit()));
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.11.3.5 void FwUpgradeDialog::fwUpdateProgress (int *current*, int *total*) [private],[slot]

Definition at line 163 of file fwupgradedialog.cpp.

References ui.

Referenced by FwUpgradeDialog().

```
163                                                     {
164     if(current > 0 && total > 0){
165         ui->progressBar->setMaximum(total);
166         ui->progressBar->setValue(current);
167     }
168 }
```

Here is the caller graph for this function:

8.11.3.6 void FwUpgradeDialog::on_actionButton_clicked ( ) [private],[slot]

Definition at line 191 of file fwupgradedialog.cpp.

References DIALOG_HEIGHT_MIN, DIALOG_WIDTH, evLoop, Kb::fwUpdate(), kb, savePath, and ui.

```
191                                                 {
192     if(!savePath.isEmpty() && kb){
193         // Start upgrade
194         setFixedSize(DIALOG_WIDTH, DIALOG_HEIGHT_MIN);
195         ui->progressBar->show();
196         ui->progressBar->setValue(0);
197         ui->progressBar->setMaximum(0);
198         ui->progressBar->setTextVisible(true);
199         ui->cancelButton->hide();
200         ui->actionButton->setEnabled(false);
201         ui->actionButton->setText("Please wait");
202         ui->statusLabel->setText("Installing firmware...");
203         kb->fwUpdate(savePath);
204     } else {
205         // Finished, close dialog.
206         if(evLoop)
207             evLoop->quit();
208     }
209 }
```

Here is the call graph for this function:

8.11.3.7 void FwUpgradeDialog::on_cancelButton_clicked ( ) [private],[slot]

Definition at line 185 of file fwupgradedialog.cpp.

References evLoop, and exitSuccess.

```
185                                                 {
186     exitSuccess = false;
187     if(evLoop)
188         evLoop->quit();
189 }
```

8.11.3.8 void FwUpgradeDialog::removeDev ( ) [private],[slot]

Definition at line 156 of file fwupgradedialog.cpp.

References fwUpdateFinished(), kb, and savePath.

Referenced by FwUpgradeDialog().

```
156                                 {
157     kb = 0;
158     // Assume success if upgrade in progress
159     if(!savePath.isEmpty())
160         fwUpdateFinished(true);
161 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.11.3.9 const QString & FwUpgradeDialog::saveBlob ( ) [private]

Definition at line 57 of file fwupgradedialog.cpp.

References blob, and savePath.

Referenced by exec().

```
57                                          {
58     if(!savePath.isEmpty())
59         return savePath;
60     QDir tmp = QDir::temp();
61     qint64 pid = QCoreApplication::applicationPid();
62     QString path = tmp.absoluteFilePath(QString("ckb-%1-fwblob.bin").arg(pid));
63     QFile output(path);
64     if(!output.open(QIODevice::WriteOnly)){
65         return savePath;
66     }
67     if(!output.write(blob)){
68         output.close();
69         tmp.remove(path);
70         return savePath;
71     }
72     output.close();
73     savePath = path;
74     return savePath;
75 }
```

Here is the caller graph for this function:

8.11.4 Field Documentation

8.11.4.1 QByteArray FwUpgradeDialog::blob [private]

Definition at line 41 of file fwupgradedialog.h.

Referenced by exec(), and saveBlob().

8.11.4.2 QEventLoop* FwUpgradeDialog::evLoop [private]

Definition at line 45 of file fwupgradedialog.h.

Referenced by exec(), fwUpdateFinished(), on_actionButton_clicked(), and on_cancelButton_clicked().

8.11.4.3 bool FwUpgradeDialog::exitSuccess [private]

Definition at line 46 of file fwupgradedialog.h.

Referenced by exec(), and on_cancelButton_clicked().

8.11.4.4 Kb* FwUpgradeDialog::kb [private]

Definition at line 42 of file fwupgradedialog.h.

Referenced by exec(), FwUpgradeDialog(), on_actionButton_clicked(), and removeDev().

8.11.4.5 QString FwUpgradeDialog::savePath [private]

Definition at line 37 of file fwupgradedialog.h.

Referenced by cleanBlob(), on_actionButton_clicked(), removeDev(), and saveBlob().

8.11.4.6 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

8.12 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

8.12.1 Detailed Description

Definition at line 6 of file gradientbutton.h.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 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:

8.12.3 Member Function Documentation

8.12.3.1 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:

8.12.3.2 void GradientButton::gradientChanged ( ) [signal]

Referenced by pickGradient().

Here is the caller graph for this function:

8.12.3.3 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:

8.12.3.4 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:

8.12.3.5 void GradientButton::updateImage ( ) [private]

Definition at line 17 of file gradientbutton.cpp.

References _alpha, and _stops.

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:

8.12.4 Field Documentation

8.12.4.1 bool GradientButton::_alpha [private]

Definition at line 23 of file gradientbutton.h.

Referenced by fromString(), and updateImage().

8.12.4.2 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

8.13 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

8.13.1 Detailed Description

Definition at line 12 of file gradientdialog.h.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 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, updatePresets(), and CkbSettings::value().

```
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()));
```

```
        }
        presets[pName] = preset;
    }

    updatePresets();
}
```

Here is the call graph for this function:

8.13.2.2 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:

8.13.3 Member Function Documentation

8.13.3.1 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:

8.13.3.2 void GradientDialog::colorChanged (QColor *color*) [private], [slot]

Definition at line 234 of file gradientdialog.cpp.

References setPreset(), and ui.

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:

8.13.3.3 void GradientDialog::currentChanged (QColor *color,* bool *spontaneous,* int *position*) [private],[slot]

Definition at line 128 of file gradientdialog.cpp.

References setPreset(), and ui.

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:

8.13.3.4 QGradientStops GradientDialog::getGradient (const QGradientStops & *prevGradient*)

Definition at line 67 of file gradientdialog.cpp.

References ui.

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 caller graph for this function:

8.13.3.5 QIcon GradientDialog::makeIcon (const Preset & *preset*) [private]

Definition at line 92 of file gradientdialog.cpp.

References GradientDialog::Preset::gradient.

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.);
```

```
    gradient.setStops(preset.gradient);
    painter.setBrush(gradient);
    painter.drawRect(0, 0, w - 1, h - 1);
    return QPixmap::fromImage(image);
}
```

Here is the caller graph for this function:

8.13.3.6 void GradientDialog::on_presetDelete_clicked ( ) [private],[slot]

Definition at line 211 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, currentPreset, GradientDialog::Preset::name, presets, 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:

8.13.3.7 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:

8.13.3.8 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:

8.13.3.9 void GradientDialog::on_presetSave_clicked ( ) [private],[slot]

Definition at line 183 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, currentPreset, GradientDialog::Preset::name, presets, ui, and updatePresets().

```
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;
```

```
    } else if(previous.name != ""){
        // Warn when overwriting an existing preset
        if(QMessageBox::warning(this, "Warning", QString("Preset \"%1\" already exists. Replace?").arg(name
    ), QMessageBox::StandardButtons(QMessageBox::Save | QMessageBox::Cancel)) != QMessageBox::Save){
            setFocus();
            return;
        }
    }
    presets[pName] = Preset(name, ui->widget->stops());
    currentPreset = pName;
    ui->presetDelete->setEnabled(true);
    ui->presetSave->setEnabled(false);
    updatePresets();
    setFocus();
}
```

Here is the call graph for this function:

8.13.3.10 void GradientDialog::on_stopOpacity_valueChanged (int *arg1*) [private],[slot]

Definition at line 240 of file gradientdialog.cpp.

References colorChanged(), setPreset(), and ui.

```
                                                    {
    colorChanged(ui->stopColor->color());
    setPreset("");
}
```

Here is the call graph for this function:

8.13.3.11 void GradientDialog::on_stopPos_valueChanged (int *arg1*) [private],[slot]

Definition at line 227 of file gradientdialog.cpp.

References setPreset(), and ui.

```
                                                {
    int res = ui->widget->moveCurrent(arg1);
    if(res != arg1)
        ui->stopPos->setValue(res);
    setPreset("");
}
```

Here is the call graph for this function:

8.13.3.12 void GradientDialog::setPreset (const QString & *newPreset*) [private]

Definition at line 151 of file gradientdialog.cpp.

References GradientDialog::Preset::builtIn, currentPreset, GradientDialog::Preset::gradient, GradientDialog::-Preset::name, presets, and ui.

Referenced by colorChanged(), currentChanged(), on_presetList_currentItemChanged(), on_presetName_text-Edited(), 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 caller graph for this function:

8.13.3.13 void GradientDialog::updatePresets ( ) [private]

Definition at line 115 of file gradientdialog.cpp.

References currentPreset, makeIcon(), GradientDialog::Preset::name, 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:

8.13.4 Field Documentation

8.13.4.1 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().

8.13.4.2 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().

8.13.4.3 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

8.14 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

8.14.1 Detailed Description

Definition at line 8 of file gradientdialogwidget.h.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 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 }
```

8.14.3 Member Function Documentation

8.14.3.1 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:

8.14.3.2 int GradientDialogWidget::current ( ) [inline]

Definition at line 20 of file gradientdialogwidget.h.

References _current.

```
20 { return _current; }
```

8.14.3.3 void GradientDialogWidget::currentChanged (QColor *color,* bool *spontaneous,* int *position*) [signal]

Referenced by mouseReleaseEvent(), and setStops().

Here is the caller graph for this function:

8.14.3.4 QRect GradientDialogWidget::fillRect ( ) [private]

Definition at line 138 of file gradientdialogwidget.cpp.

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:

8.14.3.5 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:

8.14.3.6 void GradientDialogWidget::mouseMoveEvent (QMouseEvent * *event*) [private]

Definition at line 255 of file gradientdialogwidget.cpp.

References _colors, fillRect(), makeStops(), right(), selected, selectedOffset, and selectedPos.

```
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:

8.14.3.7 void GradientDialogWidget::mousePressEvent (QMouseEvent * *event*) [private]

Definition at line 222 of file gradientdialogwidget.cpp.

References _colors, colorAt(), fillRect(), makeStops(), mouseReleaseEvent(), selected, selectedOffset, selectedPos, selectedStop(), and stopRect().

```
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:

8.14.3.8 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:

8.14.3.9 int GradientDialogWidget::moveCurrent (int *to*)

Definition at line 94 of file gradientdialogwidget.cpp.

References _colors, _current, makeStops(), and selected.

```
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         }
```

```
    } else {
        while(_colors.contains(to)){
            to--;
            if(to < 0)
                return _current;
        }
    }
    _colors[to] = _colors.take(_current);
    _current = to;
    makeStops();
    return to;
}
```

Here is the call graph for this function:

8.14.3.10 void GradientDialogWidget::paintEvent (QPaintEvent *) [private]

Definition at line 172 of file gradientdialogwidget.cpp.

References _colors, _current, _stops, fillRect(), selected, selectedPos, and stopRect().

```
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:

8.14.3.11 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:

8.14.3.12 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:

8.14.3.13 void GradientDialogWidget::setCurrentColor (const QColor & *color*)

Definition at line 117 of file gradientdialogwidget.cpp.

References _colors, _current, makeStops(), and selected.

```
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)
```

```
            continue;
        QColor& value = i.value();
        if(value.rgb() != rgb)
            break;
        int alpha = value.alpha();
        value = color;
        value.setAlpha(alpha);
    }
    makeStops();
}
```

Here is the call graph for this function:

8.14.3.14 void GradientDialogWidget::setStops (const QGradientStops & *stops*)

Definition at line 12 of file gradientdialogwidget.cpp.

References _colors, _current, _stops, currentChanged(), and stops().

```
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:

8.14.3.15 int GradientDialogWidget::stopCount ( ) [inline]

Definition at line 16 of file gradientdialogwidget.h.

References _stops.

```
16 { return _stops.count(); }
```

8.14.3.16 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:

8.14.3.17 QGradientStops GradientDialogWidget::stops ( ) [inline]

Definition at line 15 of file gradientdialogwidget.h.

References _stops.

Referenced by setStops().

```
15 { return _stops; }
```

Here is the caller graph for this function:

8.14.4 Field Documentation

8.14.4.1 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().

8.14.4.2 int GradientDialogWidget::_current [private]

Definition at line 30 of file gradientdialogwidget.h.

Referenced by current(), mouseReleaseEvent(), moveCurrent(), paintEvent(), setCurrentColor(), and setStops().

8.14.4.3 QGradientStops GradientDialogWidget::_stops [private]

Definition at line 32 of file gradientdialogwidget.h.

Referenced by colorAt(), makeStops(), paintEvent(), setStops(), stopCount(), and stops().

8.14.4.4 QColor GradientDialogWidget::selected [private]

Definition at line 28 of file gradientdialogwidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), moveCurrent(), paintEvent(), selectionColors(), and setCurrentColor().

8.14.4.5 int GradientDialogWidget::selectedOffset [private]

Definition at line 29 of file gradientdialogwidget.h.

Referenced by mouseMoveEvent(), and mousePressEvent().

8.14.4.6 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

8.15 JlCompress Class Reference

Utility class for typical operations.

`#include <src/ckb/quazip/JlCompress.h>`

Collaboration diagram for JlCompress:

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.

8.15.1 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.

8.15.2 Member Function Documentation

8.15.2.1 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:

8.15.2.2 bool JlCompress::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 JlCompress.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
```

```
    QFile inFile;
    inFile.setFileName(fileName);
    if(!inFile.open(QIODevice::ReadOnly)) return false;

    // Apro il file risulato
    QuaZipFile outFile(zip);
    if(!outFile.open(QIODevice::WriteOnly, QuaZipNewInfo(fileDest, inFile.fileName()))) return
    false;

    // Copio i dati
    if (!copyData(inFile, outFile) || outFile.getZipError()!=UNZ_OK) {
        return false;
    }

    // Chiudo i file
    outFile.close();
    if (outFile.getZipError()!=UNZ_OK) return false;
    inFile.close();

    return true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.15.2.3 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:

8.15.2.4 bool JICompress::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 JlCompress.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:

8.15.2.5 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 JlCompress.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
```

```
        if(!file.isFile()||file.absoluteFilePath()==zip->getZipName()) continue;

        // Creo il nome relativo da usare all'interno del file compresso
        QString filename = origDirectory.relativeFilePath(file.absoluteFilePath());

        // Comprimo il file
        if (!compressFile(zip,file.absoluteFilePath(),filename)) return false;
    }

    return true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.15.2.6 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 JlCompress.cpp.

References QuaZip::close(), extractFile(), QuaZip::getCurrentFileName(), QuaZip::getZipError(), QuaZip::goToFirstFile(), QuaZip::goToNextFile(), QuaZip::mdUnzip, QuaZip::open(), and removeFile().

```
                                                                    {
    // Apro lo zip
    QuaZip zip(fileCompressed);
    if(!zip.open(QuaZip::mdUnzip)) {
        return QStringList();
    }

    QDir directory(dir);
    QStringList extracted;
    if (!zip.goToFirstFile()) {
        return QStringList();
    }
    do {
        QString name = zip.getCurrentFileName();
        QString absFilePath = directory.absoluteFilePath(name);
        if (!extractFile(&zip, "", absFilePath)) {
            removeFile(extracted);
            return QStringList();
        }
        extracted.append(absFilePath);
    } while (zip.goToNextFile());

    // Chiudo il file zip
    zip.close();
    if(zip.getZipError()!=0) {
        removeFile(extracted);
        return QStringList();
    }

    return extracted;
}
```

Here is the call graph for this function:

8.15.2.7 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) {
```

```
            QFile(fileDest).setPermissions(srcPerm);
        }
        return true;
    }

    // Apro il file risultato
    QFile outFile;
    outFile.setFileName(fileDest);
    if(!outFile.open(QIODevice::WriteOnly)) return false;

    // Copio i dati
    if (!copyData(inFile, outFile) || inFile.getZipError()!=UNZ_OK) {
        outFile.close();
        removeFile(QStringList(fileDest));
        return false;
    }
    outFile.close();

    // Chiudo i file
    inFile.close();
    if (inFile.getZipError()!=UNZ_OK) {
        removeFile(QStringList(fileDest));
        return false;
    }

    if (srcPerm != 0) {
        outFile.setPermissions(srcPerm);
    }
    return true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.15.2.8 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:

8.15.2.9 QStringList JlCompress::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 JlCompress.cpp.

References QuaZip::close(), extractFile(), QuaZip::getZipError(), QuaZip::mdUnzip, QuaZip::open(), and removeFile().

```
                                                                                                                    {
    // Creo lo zip
    QuaZip zip(fileCompressed);
    if(!zip.open(QuaZip::mdUnzip)) {
        return QStringList();
    }

    // Estraggo i file
    QStringList extracted;
    for (int i=0; i<files.count(); i++) {
        QString absPath = QDir(dir).absoluteFilePath(files.at(i));
        if (!extractFile(&zip, files.at(i), absPath)) {
            removeFile(extracted);
            return QStringList();
        }
        extracted.append(absPath);
    }

    // Chiudo il file zip
    zip.close();
    if(zip.getZipError()!=0) {
        removeFile(extracted);
        return QStringList();
    }

    return extracted;
}
```

Here is the call graph for this function:

8.15.2.10 QStringList JICompress::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 JlCompress.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:

8.15.2.11 bool JICompress::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

8.16 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

8.16.1 Detailed Description

Definition at line 11 of file kb.h.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 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, cmd, 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:

8.16.2.2 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, _model, _mouseAccel, _scrollSpeed, activeDevices, cmd, 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;
```

```

    // Find an available notification node (if none is found, take notify1)
    {
        QMutexLocker locker(&notifyPathMutex);
        for(int i = 1; i < 10; i++){
            QString notify = QString(path + "/notify%1").arg(i);
            if(!QFile::exists(notify) && !notifyPaths.contains(notify)){
                notifyNumber = i;
                notifyPath = notify;
                break;
            }
        }
        notifyPaths.insert(notifyPath);
    }
    cmd.write(QString("notifyon %1\n").arg(notifyNumber).toLatin1());
    cmd.flush();

    // Again, find an available notification node for macro definition
    // (if none is found, take notify2)
    {
        QMutexLocker locker(&notifyPathMutex);
        for(int i = 1; i < 10; i++){
            QString notify = QString(path + "/notify%1").arg(i);
            if(!QFile::exists(notify) && !notifyPaths.contains(notify)){
                macroNumber = i;
                macroPath = notify;
                break;
            }
        }
        notifyPaths.insert(notifyPath);
    }
    // Activate device, apply settings, and ask for hardware profile
    cmd.write(QString("fps %1\n").arg(_frameRate).toLatin1());
    cmd.write(QString("dither %1\n").arg(static_cast<int>(_dither)).toLatin1());
    cmd.write(QString("\ndelay %1\n").arg(_delay? "on" : "off").toLatin1());
#ifdef Q_OS_MACX
    // Write ANSI/ISO flag to daemon (OSX only)
    cmd.write("layout ");
    cmd.write(KeyMap::isISO(_layout) ? "iso" : "ansi");
    // Also OSX only: scroll speed and mouse acceleration
    cmd.write(QString("accel %1\n").arg(QString(_mouseAccel ? "on" : "off")).toLatin1());
    cmd.write(QString("scrollspeed %1\n").arg(_scrollSpeed).toLatin1());
#endif
    cmd.write(QString("\nactive\n@%1 get :hwprofileid").arg(notifyNumber).toLatin1());
    hwLoading[0] = true;
    for(int i = 0; i < hwModeCount; i++){
        cmd.write(QString(" mode %1 get :hwid").arg(i + 1).toLatin1());
        hwLoading[i + 1] = true;
    }
    // Ask for current indicator and key state
    cmd.write(" get :i :keys\n");
    cmd.flush();

    emit infoUpdated();
    activeDevices.insert(this);

    // Start a separate thread to read from the notification node
    start();
}
```

Here is the call graph for this function:

8.16.3 Member Function Documentation

8.16.3.1 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:

8.16.3.2 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:

8.16.3.3 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:

8.16.3.4 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:

8.16.3.5 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:

8.16.3.6 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:

8.16.3.7 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:

8.16.3.8 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:

8.16.3.9 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:

8.16.3.10 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:

8.16.3.11 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 }
```

8.16.3.12 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:

8.16.3.13 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:

8.16.3.14 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:

8.16.3.15 void Kb::frameUpdate ( ) [slot]

Definition at line 375 of file kb.cpp.

References _currentMode, _currentProfile, KbPerf::applyIndicators(), KbMode::bind(), KbLight::close(), cmd, 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, changed);
422     cmd.write("\n");
423     cmd.flush();
424 }
```

Here is the call graph for this function:

8.16.3.16 void Kb::fwUpdate (const QString & *path*)

Definition at line 365 of file kb.cpp.

References cmd, 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:

8.16.3.17 void Kb::fwUpdateFinished (bool *succeeded*) [signal]

Referenced by readNotify().

Here is the caller graph for this function:

8.16.3.18 void Kb::fwUpdateProgress (int *current*, int *total*) [signal]

Referenced by readNotify().

Here is the caller graph for this function:

8.16.3.19 KeyMap Kb::getKeyMap ( ) [private]

Definition at line 756 of file kb.cpp.

References _layout, and _model.

Referenced by hwSave(), load(), newMode(), newProfile(), readNotify(), setCurrentProfile(), and updateLayout().

```
{
    return KeyMap(_model, _layout);
}
```

Here is the caller graph for this function:

8.16.3.20 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:

8.16.3.21 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:

8.16.3.22 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:

8.16.3.23 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:

8.16.3.24 void Kb::hwSave ( )

Definition at line 297 of file kb.cpp.

References _currentMode, _currentProfile, _hwProfile, KbProfile::append(), KbLight::base(), KbLight::close(), cmd, 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(), 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, 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:

8.16.3.25 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:

8.16.3.26 void Kb::infoUpdated ( ) [signal]

Referenced by Kb(), load(), and updateLayout().

Here is the caller graph for this function:

8.16.3.27 bool Kb::isKeyboard ( ) const [inline]

Definition at line 23 of file kb.h.

References _model, and 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:

8.16.3.28 bool Kb::isMouse ( ) const [inline]

Definition at line 24 of file kb.h.

References _model, and 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:

8.16.3.29 bool Kb::isOpen ( ) const [inline],[private]

Definition at line 147 of file kb.h.

References cmd.

Referenced by KbManager::scanKeyboards(), and ~Kb().

```
147 { return cmd.isOpen(); }
```

Here is the caller graph for this function:

8.16.3.30 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:

8.16.3.31 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:

8.16.3.32 void Kb::load ( )

Definition at line 241 of file kb.cpp.

References _needsSave, _profiles, getKeyMap(), infoUpdated(), prefsPath, profileAdded(), setCurrentProfile(), and CkbSettings::value().

Referenced by KbManager::scanKeyboards().

```
            {
    if(prefsPath.isEmpty())
        return;
    _needsSave = false;
    CkbSettings settings(prefsPath);
    // Read profiles
    KbProfile* newCurrentProfile = 0;
    QString current = settings.value("CurrentProfile").toString().trimmed().toUpper();
    foreach(QString guid, settings.value("Profiles").toString().split(" ")){
        guid = guid.trimmed().toUpper();
        if(guid != ""){
            KbProfile* profile = new KbProfile(this, getKeyMap(), settings, guid);
            _profiles.append(profile);
            if(guid == current || !newCurrentProfile)
                newCurrentProfile = profile;
        }
    }
    if(newCurrentProfile)
        setCurrentProfile(newCurrentProfile);
    else {
        // If nothing was loaded, load the demo profile
        QSettings demoSettings(":/txt/demoprofile.conf", QSettings::IniFormat, this);
        CkbSettings cSettings(demoSettings);
        KbProfile* demo = new KbProfile(this, getKeyMap(), cSettings, "{BA7FC152-2D51-4C26-A7A6-A036CC93D924}");
        _profiles.append(demo);
        setCurrentProfile(demo);
    }

    emit infoUpdated();
    emit profileAdded();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.16.3.33 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:

8.16.3.34 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 }
```

8.16.3.35 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:

8.16.3.36 void Kb::modeChanged (bool *spontaneous*) [signal]

Referenced by setCurrentMode().

Here is the caller graph for this function:

8.16.3.37 KeyMap::Model Kb::model ( ) const [inline]

Definition at line 22 of file kb.h.

References _model.

```
22 { return _model; }
```

8.16.3.38 void Kb::modeRenamed ( ) [signal]

Referenced by readNotify().

Here is the caller graph for this function:

8.16.3.39 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:

8.16.3.40 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 }
```

8.16.3.41 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:

8.16.3.42 KbMode* Kb::newMode ( ) [inline]

Definition at line 79 of file kb.h.

References getKeyMap().

Referenced by KbWidget::on_modesList_customContextMenuRequested(), and KbWidget::on_modesList_item-Clicked().

```
79 { return new KbMode(this, getKeyMap()); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.16.3.43 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:

8.16.3.44 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:

8.16.3.45 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:

8.16.3.46 void Kb::profileAdded ( ) [signal]

Referenced by load(), and readNotify().

Here is the caller graph for this function:

8.16.3.47 void Kb::profileChanged ( ) [signal]

Referenced by readNotify(), and setCurrentMode().

Here is the caller graph for this function:

8.16.3.48 void Kb::profileRenamed ( ) [signal]

Referenced by readNotify().

Here is the caller graph for this function:

8.16.3.49 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:

8.16.3.50 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; }
```

8.16.3.51 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(), KbMode::bind(), cmd, KbLight::color(), KbPerf::curDpiIdx(), 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(), and KbMode::setNeedsSave().

```
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];
```

```
        if(i.length() < 2)
            return;
        QString iName = i.mid(1);
        bool on = (i[0] == '+');
        if(iName == "num")
            iState[0] = on;
        else if(iName == "caps")
            iState[1] = on;
        else if(iName == "scroll")
            iState[2] = on;
    } else if(components[0] == "hwprofileid"){
        // Hardware profile ID
        if(components.count() < 3)
            return;
        // Find the hardware profile in the list of profiles
        QString guid = components[1];
        QString modified = components[2];
        KbProfile* newProfile = 0;
        foreach(KbProfile* profile, _profiles){
            if(profile->id().guid == guid){
                newProfile = profile;
                break;
            }
        }
        // If it wasn't found, create it
        if(!newProfile){
            newProfile = new KbProfile(this, getKeyMap(), guid, modified);
            hwLoading[0] = true;
            cmd.write(QString("@%1 get :hwprofilename\n").arg(notifyNumber).toLatin1());
            cmd.flush();
        } else {
            // If it's been updated, fetch its name
            if(newProfile->id().hwModifiedString() != modified){
                newProfile->id().modifiedString(modified);
                newProfile->id().hwModifiedString(modified);
                newProfile->setNeedsSave();
                if(hwLoading[0]){
                    cmd.write(QString("@%1 get :hwprofilename\n").arg(notifyNumber).toLatin1());
                    cmd.flush();
                }
            } else {
                hwLoading[0] = false;
            }
        }
        hwProfile(newProfile);
        emit profileAdded();
        if(_hwProfile == _currentProfile)
            emit profileChanged();
    } else if(components[0] == "hwprofilename"){
        // Hardware profile name
        QString name = QUrl::fromPercentEncoding(components[1].toUtf8());
        if(!_hwProfile || !hwLoading[0])
            return;
        QString oldName = _hwProfile->name();
        if(!(oldName.length() >= name.length() && oldName.left(name.length()) == name)){
            // Don't change the name if it's a truncated version of what we already have
            _hwProfile->name(name);
            emit profileRenamed();
        }
    } else if(components[0] == "mode"){
        // Mode-specific data
        if(components.count() < 4)
            return;
        int mode = components[1].toInt() - 1;
        if(components[2] == "hwid"){
            if(components.count() < 5 || mode >= HWMODE_MAX || !_hwProfile)
                return;
            // Hardware mode ID
            QString guid = components[3];
            QString modified = components[4];
            // Look for this mode in the hardware profile
            KbMode* hwMode = 0;
            bool isUpdated = false;
            foreach(KbMode* kbMode, _hwProfile->modes()){
                if(kbMode->id().guid == guid){
                    hwMode = kbMode;
                    if(kbMode->id().hwModifiedString() != modified){
                        // Update modification time
                        hwMode->id().modifiedString(modified);
                        hwMode->id().hwModifiedString(modified);
                        hwMode->setNeedsSave();
                        isUpdated = true;
                    } else {
                        hwLoading[mode + 1] = false;
                    }
```

```
                    break;
                }
            }
            // If it wasn't found, add it
            if(!hwMode){
                isUpdated = true;
                hwMode = new KbMode(this, getKeyMap(), guid, modified);
                _hwProfile->append(hwMode);
                // If the hardware profile now contains enough modes to be added to the list, do so
                if(!_profiles.contains(_hwProfile) && _hwProfile->modeCount() >= hwModeCount){
                    _profiles.append(_hwProfile);
                    _needsSave = true;
                    emit profileAdded();
                    if(!_currentProfile)
                        setCurrentProfile(_hwProfile);
                }
            }
            if(hwLoading[mode + 1] && isUpdated){
                // If the mode isn't in the right place, move it
                int index = _hwProfile->indexOf(hwMode);
                if(mode < _hwProfile->modeCount() && index != mode)
                    _hwProfile->move(index, mode);
                // Fetch the updated data
                cmd.write(QString("@%1 mode %2 get :hwname :hwrgb").arg(notifyNumber).arg(mode + 1).toLatin1());
                if(isMouse())
                    cmd.write(" :hwdpi :hwdpisel :hwlift :hwsnap");
                cmd.write("\n");
                cmd.flush();
            }
        } else if(components[2] == "hwname"){
            // Mode name - update list
            if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >= HWMODE_MAX || !hwLoading[mode + 1])
                return;
            KbMode* hwMode = _hwProfile->modes()[mode];
            QString name = QUrl::fromPercentEncoding(components[3].toUtf8());
            QString oldName = hwMode->name();
            if(!(oldName.length() >= name.length() && oldName.left(name.length()) == name)){
                // Don't change the name if it's a truncated version of what we already have
                hwMode->name(name);
                if(_hwProfile == _currentProfile)
                    emit modeRenamed();
            }
        } else if(components[2] == "hwrgb"){
            // RGB - set mode lighting
            if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >= HWMODE_MAX || !hwLoading[mode + 1])
                return;
            KbMode* kbmode = _hwProfile->modes()[mode];
            KbLight* light = kbmode->light();
            // Scan the input for colors
            QColor lightColor = QColor();
            for(int i = 3; i < components.count(); i++){
                QString comp = components[i];
                if(comp.indexOf(":") < 0){
                    // No ":" - single hex constant
                    bool ok;
                    int rgb = comp.toInt(&ok, 16);
                    if(ok)
                        light->color(QColor::fromRgb((QRgb)rgb));
                } else {
                    // List of keys ("a,b:xxxxxx"). Parse color first
                    QStringList set = comp.split(":");
                    bool ok;
                    int rgb = set[1].toInt(&ok, 16);
                    if(ok){
                        QColor color = QColor::fromRgb((QRgb)rgb);
                        // Parse keys
                        QStringList keys = set[0].split(",");
                        foreach(QString key, keys){
                            if(key == "light")
                                // Extrapolate the Light key to the M-keys and Lock key, since those will be set to black on hwsave
                                lightColor = color;
                            if(key.startsWith("dpi") && key.length() > 3){
                                // DPI levels go to the KbPerf object instead of KbLight
                                bool ok = false;
                                int index = key.mid(3).toInt(&ok);
                                if(ok)
                                    kbmode->perf()->dpiColor(index, color);
                                continue;
                            }
                            light->color(key, color);
                        }
                    }
```

```
                }
            }
            if(lightColor.isValid()){
                light->color("mr", lightColor);
                light->color("m1", lightColor);
                light->color("m2", lightColor);
                light->color("m3", lightColor);
                light->color("lock", lightColor);
            }
        } else if(components[2] == "hwdpi"){
            // DPI settings
            if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >= HWMODE_MAX || !hwLoading[mode + 1])
                return;
            KbPerf* perf = _hwProfile->modes()[mode]->perf();
            // Read the rest of the line as stage:x,y
            foreach(QString comp, components.mid(3)){
                QStringList dpi = comp.split(':');
                if(dpi.length() != 2)
                    continue;
                QStringList xy = dpi[1].split(',');
                int x, y;
                bool off = false;
                if(xy.length() < 2){
                    // If the right side only has one parameter, set both X and Y
                    if(xy[0] == "off")
                        off = true;
                    else
                        x = y = xy[0].toInt();
                } else {
                    x = xy[0].toInt();
                    y = xy[1].toInt();
                }
                // Set DPI for this stage
                int index = dpi[0].toInt();
                if(off){
                    perf->dpiEnabled(index, false);
                    // If all DPIs have been disabled, turn them back on
                    bool allOff = true;
                    for(int i = 1; i < KbPerf::DPI_COUNT; i++){
                        if(perf->dpiEnabled(i)){
                            allOff = false;
                            break;
                        }
                    }
                    if(allOff){
                        for(int i = 1; i < KbPerf::DPI_COUNT; i++)
                            perf->dpiEnabled(i, true);
                    }
                } else {
                    perf->dpiEnabled(index, true);
                    perf->dpi(index, QPoint(x, y));
                }
            }
        } else if(components[2] == "hwdpisel"){
            // Hardware DPI selection (0...5)
            if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >= HWMODE_MAX || !hwLoading[mode + 1])
                return;
            KbPerf* perf = _hwProfile->modes()[mode]->perf();
            int idx = components[3].toInt();
            if(idx < 1)
                idx = 1;
            if(idx >= KbPerf::DPI_COUNT)
                idx = KbPerf::DPI_COUNT - 1;
            perf->curDpiIdx(idx);
        } else if(components[2] == "hwlift"){
            // Mouse lift height (1...5)
            if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >= HWMODE_MAX || !hwLoading[mode + 1])
                return;
            KbPerf* perf = _hwProfile->modes()[mode]->perf();
            perf->liftHeight((KbPerf::height)components[3].toInt());
        } else if(components[3] == "hwsnap"){
            // Mouse angle snapping ("on" or "off")
            if(!_hwProfile || _hwProfile->modeCount() <= mode || mode >= HWMODE_MAX || !hwLoading[mode + 1])
                return;
            KbPerf* perf = _hwProfile->modes()[mode]->perf();
            perf->angleSnap(components[3] == "on");
        }
    } else if(components[0] == "fwupdate"){
        // Firmware update progress
        if(components.count() < 3)
            return;
        // Make sure path is the same
        if(components[1] != fwUpdPath)
```

```
            return;
        QString res = components[2];
        if(res == "invalid" || res == "fail")
            emit fwUpdateFinished(false);
        else if(res == "ok")
            emit fwUpdateFinished(true);
        else {
            // "xx/yy" indicates progress
            if(!res.contains("/"))
                return;
            QStringList numbers = res.split("/");
            emit fwUpdateProgress(numbers[0].toInt(), numbers[1].toInt());
        }
    }
}
```

Here is the call graph for this function:

8.16.3.52 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)
```

```
    QThread::usleep(100000);
    if(!notify.open(QIODevice::ReadOnly)){
        // If it's still not open, try again before giving up (1s at a time, 10s total)
        QThread::usleep(900000);
        for(int i = 1; i < 10; i++){
            if(notify.open(QIODevice::ReadOnly))
                break;
            QThread::sleep(1);
        }
        if(!notify.isOpen())
            return;
    }
    // Read data from notification node
    QByteArray line;
    while(notify.isOpen() && (line = notify.readLine()).length() > 0){
        QString text = QString::fromUtf8(line);
        metaObject()->invokeMethod(this, "readNotify", Qt::QueuedConnection, Q_ARG(QString, text));
    }
    QMutexLocker locker(&notifyPathMutex);
    notifyPaths.remove(notifyPath);
}
```

8.16.3.53 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:

8.16.3.54 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:

8.16.3.55 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());
```

```
        kb->cmd.flush();
    }
#endif
}
```

8.16.3.56 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:

8.16.3.57 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:

8.16.3.58 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:

8.16.3.59 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:

8.16.3.60 void Kb::updateLayout ( ) [private]

Definition at line 186 of file kb.cpp.

References _currentProfile, _hwProfile, _layout, _profiles, KbLight::close(), cmd, 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:

8.16.3.61 void Kb::writeProfileHeader ( ) [private]

Definition at line 354 of file kb.cpp.

References _currentProfile, cmd, 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:

8.16.4 Friends And Related Function Documentation

8.16.4.1 friend class KbManager [friend]

Definition at line 142 of file kb.h.

8.16.5 Field Documentation

8.16.5.1 KbMode* Kb::_currentMode [private]

Definition at line 169 of file kb.h.

Referenced by currentBind(), currentLight(), currentMode(), currentPerf(), frameUpdate(), hwSave(), readNotify(), and setCurrentMode().

8.16.5.2 KbProfile* Kb::_currentProfile [private]

Definition at line 167 of file kb.h.

Referenced by currentProfile(), frameUpdate(), hwSave(), readNotify(), save(), setCurrentMode(), updateLayout(), and writeProfileHeader().

8.16.5.3 bool Kb::_delay = false [static], [private]

Definition at line 198 of file kb.h.

Referenced by Kb(), and macroDelay().

8.16.5.4 bool Kb::_dither = false [static], [private]

Definition at line 165 of file kb.h.

Referenced by dither(), and Kb().

8.16.5.5 int Kb::_frameRate = 30 [static], [private]

Definition at line 164 of file kb.h.

Referenced by frameRate(), and Kb().

8.16.5.6 KbProfile* Kb::_hwProfile [private]

Definition at line 183 of file kb.h.

Referenced by deleteHw(), hwProfile(), hwSave(), readNotify(), and updateLayout().

8.16.5.7 KeyMap::Layout Kb::_layout = KeyMap::NO_LAYOUT [static], [private]

Definition at line 161 of file kb.h.

Referenced by getKeyMap(), Kb(), layout(), and updateLayout().

8.16.5.8 KeyMap::Model Kb::_model [private]

Definition at line 171 of file kb.h.

Referenced by getKeyMap(), isKeyboard(), isMouse(), Kb(), and model().

8.16.5.9 bool Kb::_mouseAccel = true [static], [private]

Definition at line 165 of file kb.h.

Referenced by Kb(), and mouseAccel().

8.16.5.10 bool Kb::_needsSave [private]

Definition at line 201 of file kb.h.

Referenced by appendProfile(), load(), needsSave(), profiles(), readNotify(), save(), and setCurrentMode().

8.16.5.11 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().

8.16.5.12 int Kb::_scrollSpeed = 0 [static], [private]

Definition at line 164 of file kb.h.

Referenced by Kb(), and scrollSpeed().

8.16.5.13 QFile Kb::cmd [private]

Definition at line 191 of file kb.h.

Referenced by dither(), frameRate(), frameUpdate(), fwUpdate(), hwSave(), isOpen(), Kb(), macroDelay(), mouseAccel(), readNotify(), scrollSpeed(), updateLayout(), writeProfileHeader(), and ~Kb().

8.16.5.14 QString Kb::cmdpath [private]

Definition at line 155 of file kb.h.

Referenced by Kb().

8.16.5.15 QString Kb::devpath [private]

Definition at line 155 of file kb.h.

Referenced by matches().

8.16.5.16 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().

8.16.5.17 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().

8.16.5.18 QString Kb::fwUpdPath [private]

Definition at line 181 of file kb.h.

Referenced by fwUpdate(), and readNotify().

8.16.5.19 bool Kb::hwLoading[HWMODE_MAX+1] [private]

Definition at line 205 of file kb.h.

Referenced by Kb(), and readNotify().

8.16.5.20 const int Kb::HWMODE_MAX = 3 [static]

Definition at line 50 of file kb.h.

Referenced by KbPerf::applyIndicators(), and readNotify().

8.16.5.21 int Kb::hwModeCount

Definition at line 49 of file kb.h.

Referenced by hwSave(), Kb(), KbWidget::modeIcon(), KbWidget::on_modesList_customContextMenuRequested(), readNotify(), and setCurrentProfile().

8.16.5.22 bool Kb::iState[KbPerf::HW_I_COUNT] [private]

Definition at line 174 of file kb.h.

Referenced by frameUpdate(), Kb(), and readNotify().

8.16.5.23 quint64 Kb::lastAutoSave [private]

Definition at line 178 of file kb.h.

Referenced by autoSave().

8.16.5.24 int Kb::macroNumber [private]

Definition at line 196 of file kb.h.

Referenced by getMacroNumber(), and Kb().

8.16.5.25 QString Kb::macroPath [private]

Definition at line 155 of file kb.h.

Referenced by getMacroPath(), Kb(), and ~Kb().

8.16.5.26 bool Kb::monochrome

Definition at line 19 of file kb.h.

Referenced by frameUpdate(), hwSave(), Kb(), and KbWidget::KbWidget().

8.16.5.27 int Kb::notifyNumber [private]

Definition at line 194 of file kb.h.

Referenced by frameUpdate(), fwUpdate(), Kb(), readNotify(), and ~Kb().

8.16.5.28 QString Kb::notifyPath [private]

Definition at line 155 of file kb.h.

Referenced by Kb(), and run().

8.16.5.29 QString Kb::pollrate

Definition at line 18 of file kb.h.

Referenced by KbWidget::devUpdate(), and Kb().

8.16.5.30 QString Kb::prefsPath [private]

Definition at line 177 of file kb.h.

Referenced by Kb(), load(), and save().

8.16.5.31 KbMode* Kb::prevMode [private]

Definition at line 186 of file kb.h.

Referenced by deletePrevious(), frameUpdate(), and hwSave().

8.16.5.32 KbProfile* Kb::prevProfile [private]

Definition at line 185 of file kb.h.

Referenced by frameUpdate().

8.16.5.33 QString Kb::usbModel

Definition at line 16 of file kb.h.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), Kb(), KbWidget::name(), and MainWindow::showFwUpdateNotification().

8.16.5.34 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

8.17 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

8.17.1 Detailed Description

Definition at line 11 of file kbanim.h.

8.17.2 Member Enumeration Documentation

8.17.2.1 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      };
```

8.17.3 Constructor & Destructor Documentation

8.17.3.1 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();
```

```
                if(!_parameters.contains(param.name) && param.type != AnimScript::Param::LABEL)
                    _parameters[param.name] = param.defaultValue;
            }
            _scriptName = _script->name();
            reInit();
        }
    }
}
```

Here is the call graph for this function:

8.17.3.2 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.

```
                :
    QObject(parent),
    _script(AnimScript::copy(this, script->guid())),
    _map(map), _keys(keys),
    repeatTime(0), kpRepeatTime(0), stopTime(0), kpStopTime(0), repeatMsec(0), kpRepeatMsec(0),
    _guid(QUuid::createUuid()), _name(name), _opacity(1.),
    _mode(Normal), _isActive(false), _isActiveKp(false), _needsSave(true)
{
    if(_script){
        // Set default parameters
        QListIterator<AnimScript::Param> i = _script->paramIterator();
        while(i.hasNext()){
            AnimScript::Param param = i.next();
            if(param.type != AnimScript::Param::LABEL)
                _parameters[param.name] = param.defaultValue;
        }
        _scriptGuid = script->guid();
        _scriptName = script->name();
        reInit();
    }
}
```

Here is the call graph for this function:

8.17.3.3 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:

8.17.4 Member Function Documentation

8.17.4.1 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);
```

```
    _script->frame(timestamp);

    // Blend the script's map with the current map
    int blendMode = (int)_mode;
    blendFunc blend = functions[blendMode];
    float fOpacity = _opacity / 255.f;  // save some math by pre-dividing the 255 for qAlpha

    const ColorMap& scriptMap = _script->colors();
    int count = animMap.count();
    if(scriptMap.count() != count){
        qDebug() << "Script map didn't match base map (" << count << " vs " << scriptMap.count() << "). This should never happen.";
        return;
    }
    QRgb* background = animMap.colors();
    const QRgb* foreground = scriptMap.colors();
    for(int i = 0; i < count; i++){
        // Mix the colors in with the color map according to blend mode and alpha
        QRgb& bg = background[i];
        QRgb fg = foreground[i];
        int alpha = qAlpha(fg);
        if(alpha == 0)
            continue;
        if(blendMode == 0){
            // Blend: normal
            // This is the most common use case and it requires much less arithmetic
            if(alpha == 255){
                bg = fg;
            } else {
                float r = qRed(bg), g = qGreen(bg), b = qBlue(bg);
                float a = alpha * fOpacity;
                r = r * (1.f - a) + qRed(fg) * a;
                g = g * (1.f - a) + qGreen(fg) * a;
                b = b * (1.f - a) + qBlue(fg) * a;
                bg = qRgb(round(r), round(g), round(b));
            }
        } else {
            // Use blend function
            float r = qRed(bg) / 255.f, g = qGreen(bg) / 255.f, b = qBlue(bg) / 255.f;
            float a = alpha * fOpacity;
            r = r * (1.f - a) + blend(r, qRed(fg) / 255.f) * a;
            g = g * (1.f - a) + blend(g, qGreen(fg) / 255.f) * a;
            b = b * (1.f - a) + blend(b, qBlue(fg) / 255.f) * a;
            bg = qRgb(round(r * 255.f), round(g * 255.f), round(b * 255.f));
        }
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.17.4.2 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:

8.17.4.3 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:

8.17.4.4 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:

8.17.4.5 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:

8.17.4.6 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:

8.17.4.7 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:

8.17.4.8 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:

8.17.4.9 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.)
```

```
                kpStopTime = 0;
            else
                kpStopTime = timestamp + round(stop * 1000.);
        } else {
            // If repeat is allowed, calculate stop time in repetitions
            kpRepeatMsec = repeat;
            if(delay <= 0)
                kpRepeatTime = timestamp + repeat;
            int stop = parameters.value("kpstop").toInt();
            if(stop < 0)
                kpStopTime = 0;
            else
                kpStopTime = timestamp + repeat * (1 + stop);
        }
        repeatKey = key;
    } else {
        // Key released
        _isActiveKp = false;
        _script->keypress(key, pressed, timestamp);
        if(parameters.value("kprelease").toBool())
            // Stop repeating keypress if "Stop on key release" is enabled
            kpStopTime = timestamp;
    }
    _script->frame(timestamp);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.17.4.10 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:

8.17.4.11 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:

8.17.4.12 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:

8.17.4.13 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:

8.17.4.14 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:

8.17.4.15 void KbAnim::mode (Mode *newMode*) [inline]

Definition at line 78 of file kbanim.h.

References _mode, and _needsSave.

```
78 { _needsSave = true; _mode = newMode; }
```

8.17.4.16 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:

8.17.4.17 void KbAnim::name (const QString & *newName*) [inline]

Definition at line 74 of file kbanim.h.

References _name, and _needsSave.

```
74 { _needsSave = true; _name = newName; }
```

8.17.4.18 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:

8.17.4.19 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:

8.17.4.20 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:

8.17.4.21 void KbAnim::opacity (float *newOpacity*) [inline]

Definition at line 76 of file kbanim.h.

References _needsSave, and _opacity.

```
76 { _needsSave = true; _opacity = newOpacity; }
```

8.17.4.22 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:

8.17.4.23 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:

8.17.4.24 void KbAnim::reInit ( )

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().

```
                    {
    if(_script)
        _script->init(_map, _keys, effectiveParams());
    repeatKey = "";
    _isActive = _isActiveKp = false;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.17.4.25 void KbAnim::resetParams ( )

Definition at line 127 of file kbanim.cpp.

References _tempParameters, and updateParams().

Referenced by KbAnimWidget::on_propertyButton_clicked().

```
                    {
    _tempParameters.clear();
    updateParams();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.17.4.26 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:

8.17.4.27 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:

8.17.4.28 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:

8.17.4.29 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:

8.17.4.30 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.
```

```
    _script->frame(timestamp);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.17.4.31 void KbAnim::updateParams ( ) [private]

Definition at line 132 of file kbanim.cpp.

References _script, effectiveParams(), AnimScript::parameters(), and repeatKey.

Referenced by parameter(), and resetParams().

```
                        {
    if(_script)
        _script->parameters(effectiveParams());
    repeatKey = "";
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.17.5 Field Documentation

8.17.5.1 QUuid KbAnim::_guid [private]

Definition at line 110 of file kbanim.h.

Referenced by guid(), KbAnim(), newId(), and save().

8.17.5.2 bool KbAnim::_isActive [private]

Definition at line 114 of file kbanim.h.

Referenced by catchUp(), isActive(), keypress(), reInit(), stop(), and trigger().

8.17.5.3 bool KbAnim::_isActiveKp [private]

Definition at line 114 of file kbanim.h.

Referenced by catchUp(), isActive(), keypress(), reInit(), and stop().

8.17.5.4 QStringList KbAnim::_keys [private]

Definition at line 92 of file kbanim.h.

Referenced by KbAnim(), keys(), reInit(), and save().

8.17.5.5 KeyMap KbAnim::_map [private]

Definition at line 91 of file kbanim.h.

Referenced by KbAnim(), map(), and reInit().

8.17.5.6 Mode KbAnim::_mode [private]

Definition at line 113 of file kbanim.h.

Referenced by blend(), KbAnim(), mode(), and save().

8.17.5.7 QString KbAnim::_name [private]

Definition at line 111 of file kbanim.h.

Referenced by KbAnim(), name(), and save().

8.17.5.8 bool KbAnim::_needsSave [private]

Definition at line 115 of file kbanim.h.

Referenced by commitParams(), mode(), name(), needsSave(), newId(), opacity(), and save().

8.17.5.9 float KbAnim::_opacity [private]

Definition at line 112 of file kbanim.h.

Referenced by blend(), KbAnim(), opacity(), and save().

8.17.5.10 QMap<QString, QVariant> KbAnim::_parameters [private]

Definition at line 94 of file kbanim.h.

Referenced by commitParams(), effectiveParams(), hasParameter(), KbAnim(), parameter(), and save().

8.17.5.11 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().

8.17.5.12 QUuid KbAnim::_scriptGuid [private]

Definition at line 88 of file kbanim.h.

Referenced by KbAnim(), and save().

8.17.5.13 QString KbAnim::_scriptName [private]

Definition at line 89 of file kbanim.h.

Referenced by KbAnim(), save(), and scriptName().

8.17.5.14 QMap<QString, QVariant> KbAnim::_tempParameters [private]

Definition at line 96 of file kbanim.h.

Referenced by commitParams(), effectiveParams(), parameter(), and resetParams().

8.17.5.15 int KbAnim::kpRepeatMsec [private]

Definition at line 106 of file kbanim.h.

Referenced by catchUp(), keypress(), and stop().

8.17.5.16 quint64 KbAnim::kpRepeatTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), and stop().

8.17.5.17 quint64 KbAnim::kpStopTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), and stop().

8.17.5.18 QString KbAnim::repeatKey [private]

Definition at line 104 of file kbanim.h.

Referenced by catchUp(), keypress(), reInit(), stop(), and updateParams().

8.17.5.19 int KbAnim::repeatMsec [private]

Definition at line 106 of file kbanim.h.

Referenced by catchUp(), keypress(), stop(), and trigger().

8.17.5.20 quint64 KbAnim::repeatTime [private]

Definition at line 105 of file kbanim.h.

Referenced by catchUp(), keypress(), stop(), and trigger().

8.17.5.21 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

8.18 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

8.18.1 Detailed Description

Definition at line 15 of file kbanimwidget.h.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 KbAnimWidget::KbAnimWidget (QWidget * *parent* = 0) [explicit]

Definition at line 7 of file kbanimwidget.cpp.

References reorderAnims(), setCurrent(), 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:

8.18.2.2 KbAnimWidget::~KbAnimWidget ( )

Definition at line 17 of file kbanimwidget.cpp.

References ui.

```
17                                  {
18      delete ui;
19  }
```

8.18.3 Member Function Documentation

8.18.3.1 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, KbAnim::guid(), light, KbAnim::name(), noReorder, on_propertyButton_clicked(), setCurrent(), and ui.

```
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
```

```
    setCurrent(animation);
    noReorder = false;

    // Activate settings dialog
     on_propertyButton_clicked();
}
```

Here is the call graph for this function:

8.18.3.2 void KbAnimWidget::animChanged (KbAnim * *selection*) [signal]

Referenced by setCurrent().

Here is the caller graph for this function:

8.18.3.3 void KbAnimWidget::clearSelection ( )

Definition at line 77 of file kbanimwidget.cpp.

References setCurrent(), and ui.

```
77                                  {
78     ui->animList->setCurrentItem(0);
79     setCurrent(0);
80 }
```

Here is the call graph for this function:

8.18.3.4 void KbAnimWidget::didUpdateSelection (QStringList *keys*) [signal]

Referenced by on_keyButton_clicked().

Here is the caller graph for this function:

8.18.3.5 void KbAnimWidget::duplicateAnim (KbAnim * *old*)

Definition at line 103 of file kbanimwidget.cpp.

References KbLight::animList(), KbLight::duplicateAnim(), light, 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
```

```
    setCurrent(animation);
    noReorder = false;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.18.3.6 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:

8.18.3.7 void KbAnimWidget::on_animList_customContextMenuRequested (const QPoint & *pos*) [private],[slot]

Definition at line 169 of file kbanimwidget.cpp.

References animations, 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:

8.18.3.8 void KbAnimWidget::on_animList_itemChanged (QListWidgetItem * *item*) [private],[slot]

Definition at line 158 of file kbanimwidget.cpp.

References animations, current, KbAnim::name(), 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:

8.18.3.9 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:

8.18.3.10 void KbAnimWidget::on_deleteButton_clicked ( ) [private],[slot]

Definition at line 226 of file kbanimwidget.cpp.

References animations, KbLight::animList(), current, KbAnim::guid(), light, 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:

8.18.3.11 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:

8.18.3.12 void KbAnimWidget::on_nameBox_textEdited (const QString & *arg1*) [private],[slot]

Definition at line 191 of file kbanimwidget.cpp.

References 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:

8.18.3.13 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:

8.18.3.14 void KbAnimWidget::on_propertyButton_clicked ( ) [private],[slot]

Definition at line 243 of file kbanimwidget.cpp.

References KbAnim::commitParams(), current, light, AnimSettingDialog::name(), 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:

8.18.3.15 void KbAnimWidget::refreshList ( ) [private],[slot]

Definition at line 32 of file kbanimwidget.cpp.

References animations, KbLight::animList(), KbAnim::guid(), light, KbAnim::name(), 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:

8.18.3.16 void KbAnimWidget::reorderAnims ( ) [private],[slot]

Definition at line 61 of file kbanimwidget.cpp.

References animations, KbLight::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:

8.18.3.17 void KbAnimWidget::setCurrent (KbAnim * *newCurrent*) [private]

Definition at line 118 of file kbanimwidget.cpp.

References animChanged(), AnimScript::copyright(), current, KbAnim::keys(), KbAnim::mode(), AnimScript::name(), KbAnim::name(), KbAnim::opacity(), KbAnim::script(), KbAnim::scriptName(), selectedKeys, 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:

8.18.3.18 void KbAnimWidget::setLight (KbLight * *newLight*)

Definition at line 21 of file kbanimwidget.cpp.

References light, and refreshList().

```
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:

8.18.3.19 void KbAnimWidget::setSelectedKeys (const QStringList & *keys*)

Definition at line 143 of file kbanimwidget.cpp.

References selectedKeys, and ui.

```
143                                                         {
144      selectedKeys = keys;
145      if(keys.count() == 0)
146          ui->keyButton->setVisible(false);
147      else
148          ui->keyButton->setVisible(true);
149  }
```

8.18.4 Field Documentation

8.18.4.1 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().

8.18.4.2 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().

8.18.4.3 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().

8.18.4.4 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().

8.18.4.5 QStringList KbAnimWidget::selectedKeys [private]

Definition at line 57 of file kbanimwidget.h.

Referenced by on_keyButton_clicked(), setCurrent(), and setSelectedKeys().

8.18.4.6 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

8.19 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 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)

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

8.19.1 Detailed Description

Definition at line 20 of file kbbind.h.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 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 }
```

8.19.2.2 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     }
```

```
    newBind.clear();      // here we *must not* delete the KeyActions, because they are referenced by _bind
    now
}
```

Here is the call graph for this function:

8.19.3 Member Function Documentation

8.19.3.1 QString KbBind::action (const QString & *key*)

Definition at line 140 of file kbbind.cpp.

References bindAction(), globalRemap(), and KeyAction::value().

Referenced by friendlyActionName(), KbBindWidget::on_copyButton_clicked(), resetAction(), RebindWidget::setSelection(), and KbBindWidget::updateBind().

```
                                        {
    QString rKey = globalRemap(key);
    return bindAction(rKey)->value();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.2 KeyAction* KbBind::bindAction (const QString & *key*) [inline], [private]

Definition at line 117 of file kbbind.h.

References _bind, KeyAction::defaultAction(), and KeyAction.

Referenced by action(), keyEvent(), and update().

```
117 { if(!_bind.contains(key)) return _bind[key] = new KeyAction(
      KeyAction::defaultAction(key), this); return _bind[key]; }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.3 QString KbBind::defaultAction (const QString & *key*) [static]

Definition at line 145 of file kbbind.cpp.

References KeyAction::defaultAction(), and globalRemap().

Referenced by KeyWidget::paintEvent(), and RebindWidget::setBind().

```
145                                                 {
146     QString rKey = globalRemap(key);
147     return KeyAction::defaultAction(rKey);
148 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.4 Kb* KbBind::devParent ( ) const [inline], [private]

Definition at line 114 of file kbbind.h.

References _devParent.

Referenced by getMacroNumber(), getMacroPath(), and KeyAction::keyEvent().

```
114 { return _devParent; }
```

Here is the caller graph for this function:

8.19.3.5 void KbBind::didLoad ( ) [signal]

Referenced by load().

Here is the caller graph for this function:

8.19.3.6 QString KbBind::friendlyActionName (const QString & *key*)

Definition at line 157 of file kbbind.cpp.

References _map, action(), and KeyAction.

Referenced by KbBindWidget::updateSelDisplay().

```
157                                                 {
158     QString act = action(key);
159     return KeyAction(act).friendlyName(_map);
160 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.7 QString KbBind::friendlyName (const QString & *key*)

Definition at line 150 of file kbbind.cpp.

References _map, Key::friendlyName(), and globalRemap().

```
                                                {
    const Key& pos = _map[globalRemap(key)];
    if(!pos)
        return "(Unknown)";
    return pos.friendlyName();
}
```

Here is the call graph for this function:

8.19.3.8 int KbBind::getMacroNumber ( )

Returns

number of notification channel. Use it in combination with notifyon/off-Statement

Definition at line 257 of file kbbind.cpp.

References devParent(), and Kb::getMacroNumber().

Referenced by handleNotificationChannel(), and RebindWidget::on_btnStartMacro_clicked().

```
                                {
    return devParent()->getMacroNumber();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.9 QString KbBind::getMacroPath ( )

Returns

Filepath of macro notification pipe. If not set, returns initial value ""

Definition at line 265 of file kbbind.cpp.

References devParent(), and Kb::getMacroPath().

Referenced by RebindWidget::on_btnStartMacro_clicked().

```
265                                 {
266     return devParent()->getMacroPath();
267 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.10 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:

8.19.3.11 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 275 of file kbbind.cpp.

References getMacroNumber(), and lastCmd.

Referenced by RebindWidget::on_btnStartMacro_clicked(), and RebindWidget::on_btnStopMacro_clicked().

```
275                                         {
276     if (getMacroNumber() > 0 && lastCmd) {
277         if (start) {
278             lastCmd->write (QString("\nnotifyon %1\n@%1 notify all:on\n").arg(
      getMacroNumber()).toLatin1());
279         } else {
280             lastCmd->write (QString("\n@%1 notify all:off\nnotifyoff %1\n").arg(
      getMacroNumber()).toLatin1());
281         }
282         lastCmd->flush();
283     } else qDebug() << QString("No cmd or valid handle for notification found, macroNumber = %1, lastCmd =
      %2")
284                         .arg(getMacroNumber()).arg(lastCmd? "set" : "unset");
285 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.12 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:

8.19.3.13 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:

8.19.3.14 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:

8.19.3.15 void KbBind::keyEvent (const QString & *key*, bool *down*) [slot]

Definition at line 287 of file kbbind.cpp.

References bindAction(), globalRemap(), and KeyAction::keyEvent().

Referenced by Kb::readNotify().

```
                                                    {
    QString rKey = globalRemap(key);
    KeyAction* act = bindAction(rKey);
    if(act)
        act->keyEvent(this, down);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.16 void KbBind::layoutChanged ( ) [signal]

Referenced by map().

Here is the caller graph for this function:

8.19.3.17 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:

8.19.3.18 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().

```
                                    {
    _needsSave = false;
    SGroup group(settings, "Binding");
    KeyMap currentMap = _map;
    _map = KeyMap::fromName(settings.value("KeyMap").toString());
    // Load key settings
    bool useReal = settings.value("UseRealNames").toBool();
    _bind.clear();
    {
        SGroup group(settings, "Keys");
        foreach(QString key, settings.childKeys()){
            QString name = key.toLower();
            if(!useReal)
                name = _map.fromStorage(name);
            QString bind = settings.value(key).toString();
            _bind[name] = new KeyAction(bind, this);
        }
    }
    _winLock = settings.value("WinLock").toBool();
    emit didLoad();
    map(currentMap);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.19 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:

8.19.3.20 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:

8.19.3.21 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:

8.19.3.22 KbMode* KbBind::modeParent ( ) const [inline],[private]

Definition at line 115 of file kbbind.h.

Referenced by light(), and perf().

```
115 { return (KbMode*)parent(); }
```

Here is the caller graph for this function:

8.19.3.23 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:

8.19.3.24 void KbBind::noAction (const QString & *key*)

Definition at line 172 of file kbbind.cpp.

References _bind, _map, globalRemap(), KeyMap::key(), KeyAction, and resetAction().

Referenced by RebindWidget::applyChanges(), and RebindWidget::on_unbindButton_clicked().

```
172                                     {
173     resetAction(key);
174     QString rKey = globalRemap(key);
175     if(!_map.key(rKey))
176         return;
177     _bind[rKey] = new KeyAction(this);
178 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.25 void KbBind::noAction (const QStringList & *keys*) [inline]

Definition at line 75 of file kbbind.h.

References noAction().

Referenced by noAction().

```
75 { foreach(const QString& key, keys) noAction(key); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.26 KbPerf * KbBind::perf ( )

Definition at line 50 of file kbbind.cpp.

References modeParent(), and KbMode::perf().

Referenced by KeyAction::keyEvent(), and RebindWidget::setBind().

```
                    {
    return modeParent()->perf();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.27 void KbBind::resetAction (const QString & *key*)

Definition at line 162 of file kbbind.cpp.

References _bind, _needsSave, _needsUpdate, action(), and globalRemap().

Referenced by noAction(), KbBindWidget::on_resetButton_clicked(), RebindWidget::on_resetButton_clicked(), and setAction().

```
                                        {
    QString rKey = globalRemap(key);
    // Clean up existing action (if any)
    KeyAction* action = _bind.value(rKey);
    delete action;
    _bind.remove(rKey);
    _needsUpdate = true;
    _needsSave = true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.28 void KbBind::resetAction (const QStringList & *keys*) [inline]

Definition at line 72 of file kbbind.h.

References resetAction().

Referenced by resetAction().

```
72 { foreach(const QString& key, keys) resetAction(key); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.29 void KbBind::save (CkbSettings & *settings*)

Definition at line 81 of file kbbind.cpp.

References _bind, _map, _needsSave, _winLock, KeyAction::defaultAction(), KeyMap::name(), CkbSettings::setValue(), and KeyAction::value().

Referenced by KbMode::save().

```
{
    _needsSave = false;
    SGroup group(settings, "Binding");
    settings.setValue("KeyMap", _map.name());
    // Save key settings
    settings.setValue("UseRealNames", true);
    {
        SGroup group(settings, "Keys");
        foreach(QString key, _bind.keys()){
            KeyAction* act = _bind.value(key);
            if(act && act->value() != KeyAction::defaultAction(key))
                settings.setValue(key, act->value());
        }
    }
    settings.setValue("WinLock", _winLock);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.30 void KbBind::saveGlobalRemap ( ) [static]

Definition at line 124 of file kbbind.cpp.

References _globalRemap, and CkbSettings::setValue().

Referenced by SettingsWidget::updateModifiers().

```
{
    CkbSettings settings("Program/GlobalRemap", true);
    QHashIterator<QString, QString> i(_globalRemap);
    while(i.hasNext()){
        i.next();
        settings.setValue(i.key(), i.value());
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.31 void KbBind::setAction (const QString & *key,* const QString & *action*)

Definition at line 180 of file kbbind.cpp.

References _bind, _map, globalRemap(), KeyMap::key(), KeyAction, and resetAction().

Referenced by RebindWidget::applyChanges(), and KbBindWidget::on_copyButton_clicked().

```
180                                                                 {
181     resetAction(key);
182     QString rKey = globalRemap(key);
183     if(!_map.key(rKey))
184         return;
185     _bind[rKey] = new KeyAction(action, this);
186 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.32 void KbBind::setAction (const QStringList & *keys*, const QString & *action*) [inline]

Definition at line 69 of file kbbind.h.

References setAction().

Referenced by setAction().

```
69 { foreach(const QString& key, keys) setAction(key, action); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.33 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().

```
                                                                    {
    _globalRemap.clear();
    // Ignore any keys with the standard binding
    QHashIterator<QString, QString> i(keyToActual);
    while(i.hasNext()){
        i.next();
        if(i.key() != i.value())
            _globalRemap[i.key()] = i.value();
    }
    globalRemapTime = QDateTime::currentMSecsSinceEpoch();
}
```

Here is the caller graph for this function:

8.19.3.34 void KbBind::setNeedsUpdate ( ) [inline]

Definition at line 84 of file kbbind.h.

References _needsUpdate.

Referenced by KbMode::setNeedsUpdate().

```
84 { _needsUpdate = true; }
```

Here is the caller graph for this function:

8.19.3.35 void KbBind::update (QFile & *cmd*, bool *force* = false)

Definition at line 188 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().

```
                                        {
    if(!force && !_needsUpdate && lastGlobalRemapTime == globalRemapTime)
        return;
    lastGlobalRemapTime = globalRemapTime;
    emit updated();
    _needsUpdate = false;
    // Reset all keys and enable notifications for all
    cmd.write("rebind all notify all");
    // Make sure modifier keys are included as they may be remapped globally
    QHash<QString, KeyAction*> bind(_bind);
    if(!_bind.contains("caps")) bind["caps"] = 0;
    if(!_bind.contains("lshift")) bind["lshift"] = 0;
    if(!_bind.contains("rshift")) bind["rshift"] = 0;
    if(!_bind.contains("lctrl")) bind["lctrl"] = 0;
    if(!_bind.contains("rctrl")) bind["rctrl"] = 0;
    if(!_bind.contains("lwin")) bind["lwin"] = 0;
    if(!_bind.contains("rwin")) bind["rwin"] = 0;
    if(!_bind.contains("lalt")) bind["lalt"] = 0;
    if(!_bind.contains("ralt")) bind["ralt"] = 0;
    if(!_bind.contains("fn")) bind["fn"] = 0;
    QHashIterator<QString, KeyAction*> i(bind);

    // Initialize String buffer for macro Key definitions (G-keys)
    // "macro clear" is neccessary, if an older definition is unbound.
    QString macros = "\nmacro clear\n";

    // Write out rebound keys and collect infos for macro definitions
    while(i.hasNext()){
        i.next();
        QString key = i.key();
        KeyAction* act = i.value();
        if(_globalRemap.contains(key))
            act = bindAction(_globalRemap.value(key));
        if(!act)
            continue;
        QString value = act->driverName();
        if(value.isEmpty()){
            // If the key is unbound or is a special action, unbind it
            cmd.write(" unbind ");
            cmd.write(key.toLatin1());
            // if a macro definiton for the key is given,
            // add the converted string to key-buffer "macro"
            if (act->isValidMacro()) {
                if (act->macroContent().length() > 0) {
                    macros.append("macro " + key.toLatin1() + ":" + act->macroContent().toLatin1() + "\n");
                }
            }
        } else {
            // Otherwise, write the binding
            cmd.write(" bind ");
            cmd.write(key.toLatin1());
            cmd.write(":");
            cmd.write(value.toLatin1());
        }
    }
    // If win lock is enabled, unbind windows keys
    if(_winLock)
        cmd.write(" unbind lwin rwin");

    // At last, send Macro definitions if avalilable.
    // If no definitions are made, clear macro will be sent only to reset all macros,
    cmd.write(macros.toLatin1());
    lastCmd = &cmd;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.19.3.36 void KbBind::updated ( ) [signal]

Referenced by update().

Here is the caller graph for this function:

8.19.3.37 bool KbBind::winLock ( ) [inline]

Definition at line 78 of file kbbind.h.

References _winLock.

Referenced by KeyAction::keyEvent().

```
78 { return _winLock; }
```

Here is the caller graph for this function:

8.19.3.38 void KbBind::winLock (bool *newWinLock*) [inline]

Definition at line 79 of file kbbind.h.

References _needsUpdate, and _winLock.

```
79 { _winLock = newWinLock; _needsUpdate = true; }
```

8.19.4 Friends And Related Function Documentation

8.19.4.1 friend class KeyAction [friend]

Definition at line 136 of file kbbind.h.

Referenced by bindAction(), friendlyActionName(), KbBind(), load(), noAction(), and setAction().

8.19.5 Field Documentation

8.19.5.1 QHash<QString, KeyAction*> KbBind::_bind [private]

Definition at line 131 of file kbbind.h.

Referenced by bindAction(), KbBind(), load(), noAction(), resetAction(), save(), setAction(), and update().

8.19.5.2 Kb* KbBind::_devParent [private]

Definition at line 113 of file kbbind.h.

Referenced by devParent().

8.19.5.3 QHash< QString, QString > KbBind::_globalRemap [static], [private]

Definition at line 119 of file kbbind.h.

Referenced by globalRemap(), loadGlobalRemap(), saveGlobalRemap(), setGlobalRemap(), and update().

8.19.5.4 KeyMap KbBind::_map [private]

Definition at line 129 of file kbbind.h.

Referenced by friendlyActionName(), friendlyName(), isISO(), isKeyboard(), isMouse(), load(), map(), noAction(), save(), and setAction().

8.19.5.5 bool KbBind::_needsSave [private]

Definition at line 135 of file kbbind.h.

Referenced by load(), map(), needsSave(), resetAction(), and save().

8.19.5.6 bool KbBind::_needsUpdate [private]

Definition at line 134 of file kbbind.h.

Referenced by map(), resetAction(), setNeedsUpdate(), update(), and winLock().

8.19.5.7 bool KbBind::_winLock [private]

Definition at line 133 of file kbbind.h.

Referenced by load(), save(), update(), and winLock().

8.19.5.8 quint64 KbBind::globalRemapTime = 0 [static], [private]

Definition at line 120 of file kbbind.h.

Referenced by loadGlobalRemap(), setGlobalRemap(), and update().

8.19.5.9 QFile* KbBind::lastCmd [private]

Definition at line 127 of file kbbind.h.

Referenced by handleNotificationChannel(), and update().

8.19.5.10 quint64 KbBind::lastGlobalRemapTime [private]

Definition at line 121 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

8.20 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

8.20.1 Detailed Description

Definition at line 12 of file kbbindwidget.h.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 KbBindWidget::KbBindWidget (QWidget * *parent* = 0) [explicit]

Definition at line 7 of file kbbindwidget.cpp.

References newSelection(), 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:

8.20.2.2 KbBindWidget::~KbBindWidget ( )

Definition at line 16 of file kbbindwidget.cpp.

References ui.

```
16                                 {
17     delete ui;
18 }
```

8.20.3 Member Function Documentation

8.20.3.1 void KbBindWidget::newLayout ( ) [private],[slot]

Definition at line 50 of file kbbindwidget.cpp.

References bind, KbBind::map(), profile, 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:

8.20.3.2 void KbBindWidget::newSelection (QStringList *selection*) [private],[slot]

Definition at line 56 of file kbbindwidget.cpp.

References currentSelection, 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:

8.20.3.3 void KbBindWidget::on_copyButton_clicked ( ) [private],[slot]

Definition at line 108 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().

```
{
    QStringList selection = currentSelection;
    const KeyMap& map = bind->map();
    int count = selection.count();
    QString type = bind->isMouse() ? "button" : "key";
    QString text = tr("%1 %2").arg(count).arg(type) + (count == 1 ? "" : "s");
    if(count == 0){
        // Copy all keys if none selected
        selection = map.keys();
        text = tr("all %1s").arg(type);
    }
    text = "Copy binding for " + text + " to:";
    // Display popup
    ModeSelectDialog dialog(this, profile->currentMode(), profile->modes(), text);
    if(dialog.exec() != QDialog::Accepted)
        return;
    // Copy selected keys to selected modes
    QList<KbMode*> selectedModes = dialog.selection();
    foreach(KbMode* mode, selectedModes){
        KbBind* modeBind = mode->bind();
        foreach(const QString& key, selection){
            modeBind->setAction(key, bind->action(key));
        }
    }
}
```

Here is the call graph for this function:

8.20.3.4 void KbBindWidget::on_resetButton_clicked ( ) [private],[slot]

Definition at line 87 of file kbbindwidget.cpp.

References bind, KeyMap::count(), currentSelection, KbBind::isMouse(), KeyMap::keys(), KbBind::map(), KbBind::resetAction(), KeyAction::type(), and updateBind().

```
{
    QStringList selection = currentSelection;
    const KeyMap& map = bind->map();
    if(selection.isEmpty())
        // Reset all keys if none selected
        selection = map.keys();
    uint count = selection.count();
    QString type = bind->isMouse() ? "button" : "key";
    QString text;
    if(count == map.count())
        text = "<center>Reset all %1s to default?</center>";
    else if(count == 1)
        text = "<center>Reset this %1 to default?</center>";
    else
        text = tr("<center>Reset %1 %2s to default?</center>").arg(count);
    if(QMessageBox(QMessageBox::NoIcon, "Confirm action", text.arg(type), QMessageBox::Yes | QMessageBox::No, this).exec() != QMessageBox::Yes)
        return;
    bind->resetAction(selection);
```

```
    updateBind();
}
```

Here is the call graph for this function:

8.20.3.5 void KbBindWidget::setBind (KbBind * *newBind*, KbProfile * *newProfile*)

Definition at line 20 of file kbbindwidget.cpp.

References bind, newLayout(), newSelection(), profile, ui, and updateBind().

```
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:

8.20.3.6 void KbBindWidget::updateBind ( ) [private], [slot]

Definition at line 39 of file kbbindwidget.cpp.

References KbBind::action(), bind, currentSelection, KeyMap::keys(), KbBind::map(), 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
```

```
    QHash<QString, QString> actions;
    foreach(const QString& key, map.keys())
        actions[key] = bind->action(key);
    ui->keyWidget->bindMap(actions);
    ui->rbWidget->setSelection(currentSelection);
    updateSelDisplay();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.20.3.7 void KbBindWidget::updateSelDisplay ( ) [private],[slot]

unprintable code, please refer to html view

Definition at line 62 of file kbbindwidget.cpp.

References bind, currentSelection, KbBind::friendlyActionName(), Key::friendlyName(), KbBind::isMouse(), KbBind::map(), 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
77             ui->selectLabel->setText(pos.friendlyName(false).split("\n")[0] + " → " +
       bind->friendlyActionName(key).split("\n")[0]);
78         return;
83     }
84     ui->selectLabel->setText(QString("%1 %2 selected").arg(count).arg(bind->
       isMouse() ? "buttons" : "keys"));
85 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.20.4 Field Documentation

8.20.4.1 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().

8.20.4.2 QStringList KbBindWidget::currentSelection [private]

Definition at line 36 of file kbbindwidget.h.

Referenced by newSelection(), on_copyButton_clicked(), on_resetButton_clicked(), updateBind(), and updateSelDisplay().

8.20.4.3 KbProfile* KbBindWidget::profile [private]

Definition at line 35 of file kbbindwidget.h.

Referenced by newLayout(), on_copyButton_clicked(), and setBind().

8.20.4.4 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

8.21 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)

Private Types

- enum { UNKNOWN = -1, NO, YES }

Private Slots

- void processDownload (QNetworkReply ∗reply)
- void downloadFinished ()

Private Member Functions

- KbFirmware ()
- ∼KbFirmware ()
- void initManager ()
- 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
- QNetworkAccessManager * networkManager
- QNetworkReply * tableDownload
- enum KbFirmware:: { ... } hasGPG

Static Private Attributes

- static KbFirmware instance

8.21.1 Detailed Description

Definition at line 10 of file kbfirmware.h.

8.21.2 Member Enumeration Documentation

8.21.2.1 anonymous enum [private]

Enumerator

UNKNOWN

NO

YES

Definition at line 53 of file kbfirmware.h.

```
53 { UNKNOWN = -1, NO, YES } hasGPG :2;
```

8.21.3 Constructor & Destructor Documentation

8.21.3.1 KbFirmware::KbFirmware ( ) [private]

Definition at line 14 of file kbfirmware.cpp.

```
14                          :
15     lastCheck(0), lastFinished(0), networkManager(0),
       tableDownload(0), hasGPG(UNKNOWN)
16 {
17 }
```

8.21.3.2 KbFirmware::~KbFirmware ( ) [private]

Definition at line 19 of file kbfirmware.cpp.

```
19                          {
20     //delete networkManager;
21     // ^ This can cause a crash (QT bug?)
22 }
```

8.21.4 Member Function Documentation

8.21.4.1 bool KbFirmware::_checkUpdates ( ) [private]

Definition at line 30 of file kbfirmware.cpp.

References AUTO_CHECK_TIME, downloadFinished(), initManager(), lastCheck, networkManager, and tableDownload.

Referenced by checkUpdates().

```
30                                  {
31     initManager();
32     quint64 now = QDateTime::currentMSecsSinceEpoch();
33     if(now < lastCheck + AUTO_CHECK_TIME)
34         return false;
35     // First location is for debugging only.
36     // tableDownload =
       networkManager->get(QNetworkRequest(QUrl("https://raw.githubusercontent.com/frickler24/ckb-next/issues-26-Firmware-Inc
37     // This one is the production one.
38     tableDownload = networkManager->get(QNetworkRequest(QUrl("
      https://raw.githubusercontent.com/mattanger/ckb-next/master/FIRMWARE")));
39     connect(tableDownload, SIGNAL(finished()), this, SLOT(
      downloadFinished()));
40     lastCheck = now;
41     return true;
42 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.21.4.2 QByteArray KbFirmware::_fileForBoard (const QString & *features*) [private]

Definition at line 183 of file kbfirmware.cpp.

References QuaZip::csInsensitive, KbFirmware::FW::fileName, fwTable, KbFirmware::FW::hash, QuaZip::mdUnzip, networkManager, QuaZip::open(), QuaZipFile::open(), QuaZip::setCurrentFile(), tableName(), and KbFirmware::FW::url.

Referenced by dataForBoard().

```
183                                                     {
184     QString name = tableName(features);
185     FW info = fwTable.value(name);
186     if(info.hash.isEmpty())
187         return "";
188     // Download zip from URL. Wait for it to finish.
189     QNetworkReply* reply = networkManager->get(QNetworkRequest(QUrl(info.url)));
190     QEventLoop loop(this);
191     connect(reply, SIGNAL(finished()), &loop, SLOT(quit()));
192     loop.exec();
193     // Download finished, process data
```

```
    if(reply->error() != QNetworkReply::NoError)
        return "";
    QByteArray zipData = reply->readAll();
    QBuffer buffer(&zipData);
    // Open zip archive
    QuaZip zip(&buffer);
    if(!zip.open(QuaZip::mdUnzip))
        return "";
    // Find the desired file
    if(!zip.setCurrentFile(info.fileName, QuaZip::csInsensitive))
        return "";
    QuaZipFile binFile(&zip);
    if(!binFile.open(QIODevice::ReadOnly))
        return "";
    QByteArray binary = binFile.readAll();
    // Check the hash
    if(QCryptographicHash::hash(binary, QCryptographicHash::Sha256) != info.hash)
        return "";
    return binary;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.21.4.3 float KbFirmware::_latestForBoard (const QString & *features*, bool *waitForComplete*) [private]

Definition at line 165 of file kbfirmware.cpp.

References checkUpdates(), KbManager::ckbDaemonVersionF(), KbManager::ckbGuiVersionF(), KbFirmware::FW::ckbVersion, downloaded(), fwTable, KbFirmware::FW::fwVersion, KbFirmware::FW::hash, tableDownload, and tableName().

Referenced by versionForBoard().

```
                                                                                {
    if((tableDownload || checkUpdates()) && waitForComplete){
        // If waiting is desired, enter an event loop and stay here until the download is finished
        QEventLoop loop(this);
        connect(this, SIGNAL(downloaded()), &loop, SLOT(quit()));
        loop.exec();
    }
    // Find this board
    QString name = tableName(features);
    FW info = fwTable.value(name);
    if(info.hash.isEmpty())
        return 0.f;
    // Don't return the new version if the current ckb doesn't support it
    if(info.ckbVersion > KbManager::ckbGuiVersionF() || info.ckbVersion >
KbManager::ckbDaemonVersionF())
        return -1.f;
    return info.fwVersion;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.21.4.4 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:

8.21.4.5 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:

8.21.4.6 void KbFirmware::downloaded ( ) [signal]

Referenced by _latestForBoard(), and downloadFinished().

Here is the caller graph for this function:

8.21.4.7 void KbFirmware::downloadFinished ( ) [private],[slot]

Definition at line 144 of file kbfirmware.cpp.

References downloaded(), processDownload(), and tableDownload.

Referenced by _checkUpdates().

```
144                                 {
145     if(!tableDownload)
146         return;
147     processDownload(tableDownload);
148     tableDownload->deleteLater();
149     tableDownload = 0;
150     emit downloaded();
151 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.21.4.8 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:

8.21.4.9 void KbFirmware::initManager ( ) [private]

Definition at line 24 of file kbfirmware.cpp.

References networkManager.

Referenced by _checkUpdates().

```
24                                  {
25     if(networkManager)
26         return;
27     networkManager = new QNetworkAccessManager();
28 }
```

Here is the caller graph for this function:

8.21.4.10 void KbFirmware::processDownload (QNetworkReply * *reply*) [private],[slot]

Definition at line 44 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().

```
44                                                  {
45     if(reply->error() != QNetworkReply::NoError)
46         return;
47     // Update last check
48     lastCheck = lastFinished = QDateTime::currentMSecsSinceEpoch();
49     QByteArray data = reply->readAll();
50     // Don't do anything if this is the same as the last version downloaded
51     QByteArray hash = QCryptographicHash::hash(data, QCryptographicHash::Sha256);
52     if(hash == fwTableHash)
53         return;
54     fwTableHash = hash;
55     if(hasGPG == UNKNOWN){
56         // Check for a GPG installation
57         QProcess gpg;
58         gpg.start("gpg", QStringList("--version"));
59         gpg.waitForFinished();
60         if(gpg.error() == QProcess::FailedToStart)
61             // No GPG install
62             hasGPG = NO;
63         else {
64             QString output = QString::fromUtf8(gpg.readAll());
65             // Must support RSA keys and SHA256
66             if(output.contains("RSA", Qt::CaseInsensitive) && output.contains("SHA256", Qt::CaseInsensitive
    ))
67                 hasGPG = YES;
68             else
69                 hasGPG = NO;
70         }
71         if(!hasGPG)
72             qDebug() << "No GPG detected, signature verification disabled";
73     }
74     if(hasGPG){
75         // If GPG is available, check the signature on the file before proceeding.
76         QDir tmp = QDir::temp();
77         // Save file to a temporary path. Include PID to avoid conflicts
78         qint64 pid = QCoreApplication::applicationPid();
79         QString fwPath = tmp.absoluteFilePath(QString("ckb-%1-firmware").arg(pid));
80         QFile firmware(fwPath);
81         if(!firmware.open(QIODevice::WriteOnly)
82                 || firmware.write(data) != data.length()){
83             qDebug() << "Failed to write firmware file to temporary location, aborting firmware check";
```

```
            return;
        }
        firmware.close();
        // Write GPG key
        QString keyPath = tmp.absoluteFilePath(QString("ckb-%1-key.gpg").arg(pid));
        if(!QFile::copy(":/bin/ckb-next-key.gpg", keyPath)){
            firmware.remove();
            qDebug() << "Failed to write GPG key to temporary location, aborting firmware check";
            return;
        }
        // Check signature
        QProcess gpg;
        gpg.start("gpg", QStringList("--no-default-keyring") << "--keyring" << keyPath << "--verify" << fwPath);
        gpg.waitForFinished();
        // Clean up temp files
        tmp.remove(fwPath);
        tmp.remove(keyPath);
        if(gpg.error() != QProcess::UnknownError || gpg.exitCode() != 0){
            qDebug() << "GPG couldn't verify firmware signature:";
            qDebug() << gpg.readAllStandardOutput();
            qDebug() << gpg.readAllStandardError();
            return;
        }
        // Signature good, proceed to update database
    }
    fwTable.clear();
    QStringList lines = QString::fromUtf8(data).split("\n");
    bool scan = false;
    foreach(QString line, lines){
        // Collapse whitespace
        line.replace(QRegExp("\\s+"), " ").remove(QRegExp("^\\s")).remove(QRegExp("\\s$"));
        // Skip empty or commented-out lines
        if(line.length() == 0 || line.at(0) == '#')
            continue;
        // Don't read anything until the entries begin and don't read anything after they end
        if(!scan){
            if(line == "!BEGIN FW ENTRIES")
                scan = true;
            else
                continue;
        }
        if(line == "!END FW ENTRIES")
            break;
        QStringList components = line.split(" ");
        if(components.length() != 7)
            continue;
        // "VENDOR-PRODUCT"
        QString device = components[0].toUpper() + "-" + components[1].toUpper();
        FW fw;
        fw.fwVersion = components[2].toFloat();                          // Firmware blob version
        fw.url = QUrl::fromPercentEncoding(components[3].toLatin1());    // URL to zip file
        fw.ckbVersion = KbManager::parseVersionString(components[4]);    // Minimum ckb version
        fw.fileName = QUrl::fromPercentEncoding(components[5].toLatin1()); // Name of file inside zip
        fw.hash = QByteArray::fromHex(components[6].toLatin1());         // SHA256 of file inside zip
        // Update entry
        fwTable[device] = fw;
    }
    qDebug() << "Downloaded new firmware list." << fwTable.count() << "entries found.";
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.21.4.11 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:

8.21.5 Field Documentation

8.21.5.1 QMap<QString, FW> KbFirmware::fwTable `[private]`

Definition at line 41 of file kbfirmware.h.

Referenced by _fileForBoard(), _latestForBoard(), and processDownload().

8.21.5.2 QByteArray KbFirmware::fwTableHash `[private]`

Definition at line 43 of file kbfirmware.h.

Referenced by processDownload().

8.21.5.3 enum { ... } KbFirmware::hasGPG `[private]`

Referenced by processDownload().

8.21.5.4 KbFirmware KbFirmware::instance [static], [private]

Definition at line 59 of file kbfirmware.h.

Referenced by checkUpdates(), dataForBoard(), hasDownloaded(), and versionForBoard().

8.21.5.5 quint64 KbFirmware::lastCheck [private]

Definition at line 32 of file kbfirmware.h.

Referenced by _checkUpdates(), and processDownload().

8.21.5.6 quint64 KbFirmware::lastFinished [private]

Definition at line 32 of file kbfirmware.h.

Referenced by hasDownloaded(), and processDownload().

8.21.5.7 QNetworkAccessManager* KbFirmware::networkManager [private]

Definition at line 47 of file kbfirmware.h.

Referenced by _checkUpdates(), _fileForBoard(), and initManager().

8.21.5.8 QNetworkReply* KbFirmware::tableDownload [private]

Definition at line 50 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

8.22 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 _indicatorMap
- QSet< QString > _indicatorList
- quint64 lastFrameSignal
- int _dimming
- bool _start
- bool _needsSave
- bool _needsMapRefresh

8.22.1 Detailed Description

Definition at line 17 of file kblight.h.

8.22.2 Member Typedef Documentation

8.22.2.1 typedef QList<KbAnim*> KbLight::AnimList

Definition at line 46 of file kblight.h.

8.22.3 Constructor & Destructor Documentation

8.22.3.1 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), _start(false), _needsSave(true),
      _needsMapRefresh(true)
12 {
13     map(keyMap);
14 }
```

Here is the call graph for this function:

8.22.3.2 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), _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:

8.22.3.3 KbLight::~KbLight ( )

Definition at line 45 of file kblight.cpp.

References activeLights.

```
45                  {
46      activeLights.remove(this);
47  }
```

8.22.4 Member Function Documentation

8.22.4.1 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, 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:

8.22.4.2 void KbLight::animKeypress (const QString & *key*, bool *down*)

Definition at line 189 of file kblight.cpp.

References _animList, _previewAnim, 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:

8.22.4.3 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:

8.22.4.4 void KbLight::animList (const AnimList & *newAnimList*) [inline]

Definition at line 50 of file kblight.h.

References _animList, and _needsSave.

```
50 { _needsSave = true; _animList = newAnimList; }
```

8.22.4.5 void KbLight::base (QFile & *cmd*, bool *ignoreDim* = false, bool *monochrome* = false)

Definition at line 371 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().

```
371                                                         {
372     close();
373     if(_dimming == MAX_DIM && !ignoreDim){
374         cmd.write(QString().sprintf("rgb 000000").toLatin1());
375         return;
376     }
377     // Set just the background color, ignoring any animation
378     rebuildBaseMap();
379     _animMap = _colorMap;
380     // If monochrome is active, create grayscale
381     if(monochrome){
382         int count = _animMap.count();
383         QRgb* colors = _animMap.colors();
384         for(int i = 0; i < count; i++){
385             QRgb& rgb = colors[i];
386             rgb = monoRgb(qRed(rgb), qGreen(rgb), qBlue(rgb));
387         }
388     }
389     // Set a few indicators to black as the hardware handles them differently
390     QRgb* mr = _animMap.colorForName("mr"), *m1 = _animMap.
      colorForName("m1"), *m2 = _animMap.colorForName("m2"), *m3 =
      _animMap.colorForName("m3"), *lock = _animMap.
      colorForName("lock");
391     if(mr) *mr = 0;
392     if(m1) *m1 = 0;
393     if(m2) *m2 = 0;
394     if(m3) *m3 = 0;
```

```
    if(lock) *lock = 0;
    // Send to driver
    cmd.write("rgb");
    printRGB(cmd, _animMap);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.6 void KbLight::close ( )

Definition at line 215 of file kblight.cpp.

References _animList, _start, activeLights, KbAnim::stop(), and stopPreview().

Referenced by base(), Kb::frameUpdate(), Kb::hwSave(), and Kb::updateLayout().

```
                    {
    activeLights.remove(this);
    foreach(KbAnim* anim, _animList)
        anim->stop();
    stopPreview();
    _start = false;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.7 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:

8.22.4.8 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:

8.22.4.9 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:

8.22.4.10 void KbLight::didLoad ( ) [signal]

Referenced by load().

Here is the caller graph for this function:

8.22.4.11 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:

8.22.4.12 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:

8.22.4.13 KbAnim * KbLight::duplicateAnim (KbAnim * *oldAnim*)

Definition at line 148 of file kblight.cpp.

References _animList, _map, _needsSave, _start, 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:

8.22.4.14 KbAnim* KbLight::findAnim (const QUuid & *guid*) const [inline]

Definition at line 51 of file kblight.h.

References _animList, 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:

8.22.4.15 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:

8.22.4.16 void KbLight::frameDisplayed (const ColorMap & *animatedColors*, const QSet< QString > & *indicatorList*) [signal]

Referenced by frameUpdate().

Here is the caller graph for this function:

8.22.4.17 void KbLight::frameUpdate (QFile & *cmd*, bool *monochrome* = false)

Definition at line 296 of file kblight.cpp.

References _animList, _animMap, _colorMap, _dimming, _indicatorList, _indicatorMap, _previewAnim, 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     int count = _animMap.count();
307     QRgb* colors = _animMap.colors();
```

```
    // Apply active indicators and/or perform monochrome conversion
    if(monochrome || !_indicatorList.isEmpty()){
        QRgb* indicators = _indicatorMap.colors();
        for(int i = 0; i < count; i++){
            QRgb& rgb = colors[i];
            float r = qRed(rgb);
            float g = qGreen(rgb);
            float b = qBlue(rgb);
            // Apply indicators
            QRgb rgb2 = indicators[i];
            if(qAlpha(rgb2) != 0){
                float r2 = qRed(rgb2);
                float g2 = qGreen(rgb2);
                float b2 = qBlue(rgb2);
                float a2 = qAlpha(rgb2) / 255.f;
                r = round(r2 * a2 + r * (1.f - a2));
                g = round(g2 * a2 + g * (1.f - a2));
                b = round(b2 * a2 + b * (1.f - a2));
            }
            // If monochrome mode is active, average the channels to get a grayscale image
            if(monochrome)
                rgb = monoRgb(r, g, b);
            else
                rgb = qRgb(r, g, b);
        }
    }

    // Emit signals for the animation (only do this every 50ms - it can cause a lot of CPU usage)
    if(timestamp >= lastFrameSignal + 50){
        emit frameDisplayed(_animMap, _indicatorList);
        lastFrameSignal = timestamp;
    }

    // If brightness is at 0%, turn off lighting entirely
    if(_dimming == 3){
        cmd.write("rgb 000000");
        return;
    }

    float light = (3 - _dimming) / 3.f;
    // Apply global dimming
    if(light != 1.f || monochrome){
        for(int i = 0; i < count; i++){
            QRgb& rgb = colors[i];
            // Like the monochrome conversion, this should be done in a linear colorspace
            float r = sToL(qRed(rgb));
            float g = sToL(qGreen(rgb));
            float b = sToL(qBlue(rgb));
            r *= light;
            g *= light;
            b *= light;
            r = round(lToS(r));
            g = round(lToS(g));
            b = round(lToS(b));
            rgb = qRgb(r, g, b);
        }
    }

    // Apply light
    cmd.write("rgb");
    printRGB(cmd, _animMap);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.18 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:

8.22.4.19 void KbLight::load (CkbSettings & *settings*)

Definition at line 401 of file kblight.cpp.

References _animList, _dimming, _map, _needsMapRefresh, _needsSave, _qColorMap, CkbSettings::childKeys(), color(), didLoad(), KeyMap::fromName(), KeyMap::fromStorage(), map(), MAX_DIM, and CkbSettings::value().

Referenced by KbMode::KbMode().

```
401                                         {
402     // Load light settings
403     _needsSave = false;
404     SGroup group(settings, "Lighting");
405     KeyMap currentMap = _map;
406     _map = KeyMap::fromName(settings.value("KeyMap").toString());
407     _dimming = settings.value("Brightness").toUInt();
408     if(_dimming > MAX_DIM)
409         _dimming = MAX_DIM;
410     // Load RGB settings
411     bool useReal = settings.value("UseRealNames").toBool();
412     {
413         SGroup group(settings, "Keys");
414         foreach(QString key, settings.childKeys()){
415             QString name = key.toLower();
416             if(!useReal)
417                 name = _map.fromStorage(name);
418             QColor color = settings.value(key).toString();
419             if(!color.isValid())
420                 color = QColor(255, 255, 255);
421             _qColorMap[name] = color.rgb();
422         }
423         _needsMapRefresh = true;
424     }
425     // Load animations
426     foreach(KbAnim* anim, _animList)
427         anim->deleteLater();
428     _animList.clear();
429     {
430         SGroup group(settings, "Animations");
431         foreach(QString anim, settings.value("List").toStringList()){
432             QUuid id = anim;
433             _animList.append(new KbAnim(this, _map, id, settings));
434         }
435     }
436     emit didLoad();
437     map(currentMap);
438 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.20 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:

8.22.4.21 void KbLight::map (const KeyMap & *map*)

Definition at line 25 of file kblight.cpp.

References _animList, _animMap, _colorMap, _indicatorMap, _map, _needsMapRefresh, _needsSave, _qColorMap, 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:

8.22.4.22 bool KbLight::needsSave ( ) const

Definition at line 467 of file kblight.cpp.

References _animList, _needsSave, and KbAnim::needsSave().

Referenced by KbMode::needsSave().

```
467                                 {
468     if(_needsSave)
469         return true;
470     foreach(KbAnim* anim, _animList){
471         if(anim->needsSave())
472             return true;
473     }
474     return false;
475 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.23 void KbLight::open ( )

Definition at line 200 of file kblight.cpp.

References _animList, _dimming, _previewAnim, _shareDimming, _start, activeLights, dimming(), and KbAnim::trigger().

Referenced by Kb::frameUpdate().

```
                {
    // Apply shared dimming if needed
    if(_shareDimming != -1 && _shareDimming !=
_dimming)
        dimming(_shareDimming);
    activeLights.insert(this);
    if(_start)
        return;
    quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
    foreach(KbAnim* anim, _animList)
        anim->trigger(timestamp);
    if(_previewAnim)
        _previewAnim->trigger(timestamp);
    _start = true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.24 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, KbAnim::commitParams(), KbAnim::parameter(), KbAnim::reInit(), stopPreview(), and KbAnim::trigger().

Referenced by AnimAddDialog::on_presetBox_activated().

```
        {
    if(_previewAnim)
        stopPreview();
    quint64 timestamp = QDateTime::currentMSecsSinceEpoch();
    // Load the new animation and set preset parameters
    KbAnim* anim = new KbAnim(this, _map, "", keys, base);
    QMapIterator<QString, QVariant> i(preset);
    while(i.hasNext()){
        i.next();
        anim->parameter(i.key(), i.value());
```

```
    }
    anim->commitParams();
    anim->reInit();
    // Add the animation and start it
    _previewAnim = anim;
    anim->trigger(timestamp);
    _start = true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.25 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:

8.22.4.26 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:

8.22.4.27 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:

8.22.4.28 void KbLight::restartAnimation ( )

Definition at line 179 of file kblight.cpp.

References _animList, _start, 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();
```

```
    foreach(KbAnim* anim, _animList){
        anim->stop();
        anim->trigger(timestamp);
    }
    stopPreview();
    _start = true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.29 void KbLight::save (CkbSettings & *settings*)

Definition at line 440 of file kblight.cpp.

References _animList, _dimming, _map, _needsSave, _qColorMap, KbAnim::guid(), KeyMap::name(), KbAnim::save(), and CkbSettings::setValue().

Referenced by KbMode::save().

```
440                                         {
441     _needsSave = false;
442     SGroup group(settings, "Lighting");
443     settings.setValue("KeyMap", _map.name());
444     settings.setValue("Brightness", _dimming);
445     settings.setValue("UseRealNames", true);
446     {
447         // Save RGB settings
448         SGroup group(settings, "Keys");
449         QMutableColorMapIterator i(_qColorMap);
450         while(i.hasNext()){
451             i.next();
452             settings.setValue(i.key(), QColor(i.value()).name());
453         }
454     }
455     {
456         // Save animations
457         SGroup group(settings, "Animations");
458         QStringList aList;
459         foreach(KbAnim* anim, _animList){
460             aList << anim->guid().toString().toUpper();
461             anim->save(settings);
```

```
        }
        settings.setValue("List", aList);
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.22.4.30 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:

8.22.4.31 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:

8.22.4.32 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:

8.22.4.33 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:

8.22.4.34 void KbLight::updated ( ) [signal]

Referenced by dimming(), and map().

Here is the caller graph for this function:

8.22.5 Field Documentation

8.22.5.1 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().

8.22.5.2 ColorMap KbLight::_animMap [private]

Definition at line 93 of file kblight.h.

Referenced by base(), frameUpdate(), and map().

8.22.5.3 ColorMap KbLight::_colorMap [private]

Definition at line 93 of file kblight.h.

Referenced by base(), color(), frameUpdate(), map(), and rebuildBaseMap().

8.22.5.4 int KbLight::_dimming [private]

Definition at line 96 of file kblight.h.

Referenced by base(), dimming(), frameUpdate(), load(), open(), and save().

8.22.5.5 QSet<QString> KbLight::_indicatorList [private]

Definition at line 94 of file kblight.h.

Referenced by frameUpdate(), resetIndicators(), and setIndicator().

8.22.5.6 ColorMap KbLight::_indicatorMap [private]

Definition at line 93 of file kblight.h.

Referenced by frameUpdate(), map(), resetIndicators(), and setIndicator().

8.22.5.7 KeyMap KbLight::_map [private]

Definition at line 91 of file kblight.h.

Referenced by addAnim(), duplicateAnim(), load(), map(), previewAnim(), and save().

8.22.5.8 bool KbLight::_needsMapRefresh [private]

Definition at line 98 of file kblight.h.

Referenced by color(), load(), map(), and rebuildBaseMap().

8.22.5.9 bool KbLight::_needsSave [private]

Definition at line 98 of file kblight.h.

Referenced by addAnim(), animList(), color(), dimming(), duplicateAnim(), load(), map(), needsSave(), and save().

8.22.5.10 KbAnim* KbLight::_previewAnim [private]

Definition at line 90 of file kblight.h.

Referenced by animKeypress(), frameUpdate(), open(), previewAnim(), and stopPreview().

8.22.5.11 QColorMap KbLight::_qColorMap [private]

Definition at line 92 of file kblight.h.

Referenced by color(), colorMap(), load(), map(), rebuildBaseMap(), and save().

8.22.5.12 bool KbLight::_start [private]

Definition at line 97 of file kblight.h.

Referenced by addAnim(), close(), duplicateAnim(), isStarted(), open(), previewAnim(), and restartAnimation().

8.22.5.13 quint64 KbLight::lastFrameSignal [private]

Definition at line 95 of file kblight.h.

Referenced by frameUpdate().

8.22.5.14 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

8.23 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

8.23.1 Detailed Description

Definition at line 15 of file kblightwidget.h.

8.23.2 Constructor & Destructor Documentation

8.23.2.1 KbLightWidget::KbLightWidget (QWidget * *parent* = 0) [explicit]

Definition at line 6 of file kblightwidget.cpp.

References changeAnim(), changeAnimKeys(), changeColor(), AnimScript::count(), CkbSettings::get(), newSelection(), 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:

8.23.2.2 KbLightWidget::~KbLightWidget ( )

Definition at line 24 of file kblightwidget.cpp.

References ui.

```
24                                  {
25     delete ui;
26 }
```

8.23.3 Member Function Documentation

8.23.3.1 void KbLightWidget::changeAnim (KbAnim * *newAnim*) [private],[slot]

Definition at line 121 of file kblightwidget.cpp.

References KbAnim::keys(), 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:

8.23.3.2 void KbLightWidget::changeAnimKeys (QStringList *keys*) [private],[slot]

Definition at line 129 of file kblightwidget.cpp.

References ui.

Referenced by KbLightWidget().

```
129                                                 {
130     ui->keyWidget->setAnimation(keys);
131 }
```

Here is the caller graph for this function:

8.23.3.3 void KbLightWidget::changeColor (QColor *newColor*) [private],[slot]

Definition at line 99 of file kblightwidget.cpp.

References KbLight::color(), KbLight::colorMap(), currentSelection, 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:

8.23.3.4 void KbLightWidget::newSelection (QStringList *selection*) [private],[slot]

Definition at line 72 of file kblightwidget.cpp.

References KbLight::colorMap(), currentSelection, KeyMap::isMouse(), light, KbLight::map(), and ui.

Referenced by KbLightWidget().

```
72                                                  {
73     // Determine selected color (invalid color if no selection or if they're not all the same)
```

```
    QColor selectedColor;
    const QColorMap& colorMap = light->colorMap();
    foreach(const QString& key, selection){
        QColor color = colorMap.value(key);
        if(!selectedColor.isValid())
            selectedColor = color;
        else if(color != selectedColor){
            selectedColor = QColor();
            break;
        }
    }

    currentSelection = selection;
    ui->animWidget->setSelectedKeys(selection);
    ui->bgButton->color(selectedColor);
    int count = selection.count();
    if(count == 0){
        ui->selLabel->setText("Click to select");
        return;
    } else if(count == 1)
        ui->selLabel->setText(tr("1 %1 selected").arg(light->map().isMouse() ? "zone" : "key"));
    else
        ui->selLabel->setText(tr("%1 %2 selected").arg(count).arg(light->map().isMouse() ? "zones" : "keys"));
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.23.3.5 void KbLightWidget::on_animButton_clicked ( ) [private],[slot]

Definition at line 143 of file kblightwidget.cpp.

References AnimAddDialog::chosenPreset(), AnimAddDialog::chosenScript(), AnimScript::count(), currentSelection, light, AnimScript::preset(), AnimScript::presets(), KbLight::restartAnimation(), and ui.

```
143                                         {
144     if(AnimScript::count() == 0)
```

```
        return;
    if(currentSelection.isEmpty())
        ui->keyWidget->selectAll();
    AnimAddDialog dialog(this, light, currentSelection);
    dialog.exec();
    if(dialog.result() != QDialog::Accepted)
        return;
    const AnimScript* script = dialog.chosenScript();
    int presetId = dialog.chosenPreset();
    ui->animWidget->addAnim(script, currentSelection, script->
  presets()[presetId], script->preset(presetId));
    light->restartAnimation();
}
```

Here is the call graph for this function:

8.23.3.6 void KbLightWidget::on_bgButton_clicked ( ) [private],[slot]

Definition at line 138 of file kblightwidget.cpp.

References currentSelection, and ui.

```
138                                         {
139     if(currentSelection.isEmpty())
140         ui->keyWidget->selectAll();
141 }
```

8.23.3.7 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:

8.23.3.8 void KbLightWidget::on_showAnimBox_clicked (bool *checked*) [private],[slot]

Definition at line 52 of file kblightwidget.cpp.

References 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:

8.23.3.9 void KbLightWidget::setLight (KbLight * *newLight*)

Definition at line 28 of file kblightwidget.cpp.

References light, on_showAnimBox_clicked(), ui, and updateLight().

```
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();
```

```
    ui->keyWidget->setAnimation(QStringList());
    ui->animWidget->setLight(newLight);
    if(ui->showAnimBox->isChecked())
        on_showAnimBox_clicked(true);
}
```

Here is the call graph for this function:

8.23.3.10 void KbLightWidget::setMonochrome ( )

Definition at line 48 of file kblightwidget.cpp.

References ui.

```
48                                  {
49     ui->keyWidget->monochrome(true);
50 }
```

8.23.3.11 void KbLightWidget::toggleSidelight ( ) [private],[slot]

Definition at line 107 of file kblightwidget.cpp.

References KbLight::color(), KbLight::colorMap(), 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:

8.23.3.12 void KbLightWidget::updateLight ( ) [private],[slot]

Definition at line 66 of file kblightwidget.cpp.

References KbLight::colorMap(), KbLight::dimming(), light, 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:

8.23.4 Field Documentation

8.23.4.1 QStringList KbLightWidget::currentSelection [private]

Definition at line 44 of file kblightwidget.h.

Referenced by changeColor(), newSelection(), on_animButton_clicked(), and on_bgButton_clicked().

8.23.4.2 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().

8.23.4.3 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

8.24 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>"

8.24.1 Detailed Description

Definition at line 14 of file kbmanager.h.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 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(100);
32     connect(_scanTimer, SIGNAL(timeout()), this, SLOT(scanKeyboards()));
33 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.24.3 Member Function Documentation

8.24.3.1 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:

8.24.3.2 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:

8.24.3.3 static QString KbManager::ckbGuiVersion ( ) [inline],[static]

Definition at line 26 of file kbmanager.h.

References _guiVersion.

```
26 { return _guiVersion; }
```

8.24.3.4 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:

8.24.3.5 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*>(); }
```

8.24.3.6 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:

8.24.3.7 void KbManager::fps (int *framerate*) [static]

Definition at line 35 of file kbmanager.cpp.

References eventTimer().

Referenced by Kb::frameRate().

```
35                                  {
36     QTimer* timer = eventTimer();
37     if(!timer)
38         return;
39     if(timer->isActive())
40         timer->setInterval(1000 / framerate);
41     else
42         timer->start(1000 / framerate);
43 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.24.3.8 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:

8.24.3.9 void KbManager::kbConnected (Kb * *device*) [signal]

Referenced by scanKeyboards().

Here is the caller graph for this function:

8.24.3.10 void KbManager::kbDisconnected (Kb * *device*) [signal]

Referenced by scanKeyboards().

Here is the caller graph for this function:

8.24.3.11 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:

8.24.3.12 float KbManager::parseVersionString (QString *version*) [static]

Definition at line 45 of file kbmanager.cpp.

Referenced by ckbDaemonVersionF(), ckbGuiVersionF(), and KbFirmware::processDownload().

```
45                                                  {
46     // Remove extraneous info (anything after a +, anything non-numeric)
47     QStringList dots = version.replace(QRegExp("\\+.+"), "").replace(QRegExp("[^\\d\\.]"), "").split(".");
48     float base = 1.f;
49     float res = 0.f;
50     // A number like "1.2.3" will result in 1.0203
51     // 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.
52     foreach(const QString& dot, dots){
53         res += dot.toFloat() * base;
54         base /= 100.f;
55     }
56     return res;
57 }
```

Here is the caller graph for this function:

8.24.3.13 void KbManager::scanKeyboards ( ) [private],[slot]

Definition at line 59 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().

```
59                                  {
60     QString rootdev = devpath.arg(0);
61     QFile connected(rootdev + "/connected");
62     if(!connected.open(QIODevice::ReadOnly)){
63         // No root controller - remove all keyboards
64         foreach(Kb* kb, _devices){
65             emit kbDisconnected(kb);
66             kb->save();
67             delete kb;
68         }
69         _devices.clear();
70         if(_daemonVersion != DAEMON_UNAVAILABLE_STR){
71             _daemonVersion = DAEMON_UNAVAILABLE_STR;
72             emit versionUpdated();
73         }
74         return;
75     }
76     // Check daemon version
77     QFile version(rootdev + "/version");
78     QString vString;
79     if(version.open(QIODevice::ReadOnly)){
80         vString = QString::fromUtf8(version.readLine()).trimmed();
81         version.close();
82     } else
83         vString = DAEMON_UNAVAILABLE_STR;
84     if(_daemonVersion != vString){
85         _daemonVersion = vString;
86         emit versionUpdated();
87     }
88
89     // Scan connected devices
90     QList<QStringList> lines;
91     while(1){
92         QString line = connected.readLine().trimmed();
93         if(line.isEmpty())
94             break;
```

```
        QStringList components = line.split(" ");
        if(components.length() < 2)             // "<path> <serial> <name>" (we're only interested in the first two)
            continue;
        lines.append(components);
    }
    connected.close();

    // Remove any active devices not in the list
    QMutableSetIterator<Kb*> i(_devices);
    while(i.hasNext()){
        Kb* kb = i.next();
        bool matched = false;
        foreach(const QStringList& line, lines){
            if(kb->matches(line[0], line[1])){
                matched = true;
                break;
            }
        }
        if(matched)
            continue;
        // Device not found, remove
        i.remove();
        emit kbDisconnected(kb);
        kb->save();
        delete kb;
    }

    // Add any new devices found in the list
    foreach(const QStringList& line, lines){
        bool matched = false;
        foreach(Kb* kb, _devices){
            if(kb->matches(line[0], line[1])){
                matched = true;
                break;
            }
        }
        if(matched)
            continue;
        // Device not found, create new
        Kb* kb = new Kb(this, line[0]);
        if(!kb->isOpen()){
            delete kb;
            continue;
        }
        _devices.insert(kb);
        // Load preferences and send signal
        emit kbConnected(kb);
        kb->load();
        connect(_eventTimer, SIGNAL(timeout()), kb, SLOT(frameUpdate()));
        connect(_scanTimer, SIGNAL(timeout()), kb, SLOT(autoSave()));
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.24.3.14 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:

8.24.3.15 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:

8.24.3.16 void KbManager::versionUpdated ( ) [signal]

Referenced by scanKeyboards().

Here is the caller graph for this function:

8.24.4 Field Documentation

8.24.4.1 QString KbManager::_daemonVersion = "<unavailable>" [static], [private]

Definition at line 59 of file kbmanager.h.

Referenced by ckbDaemonVersion(), ckbDaemonVersionF(), and scanKeyboards().

8.24.4.2 QSet<Kb*> KbManager::_devices [private]

Definition at line 63 of file kbmanager.h.

Referenced by devices(), and scanKeyboards().

8.24.4.3 QTimer* KbManager::_eventTimer [private]

Definition at line 64 of file kbmanager.h.

Referenced by eventTimer(), KbManager(), and scanKeyboards().

8.24.4.4 QString KbManager::_guiVersion [static], [private]

Definition at line 59 of file kbmanager.h.

Referenced by ckbGuiVersion(), ckbGuiVersionF(), and init().

8.24.4.5 KbManager * KbManager::_kbManager = 0 [static], [private]

Definition at line 58 of file kbmanager.h.

Referenced by devices(), eventTimer(), init(), kbManager(), scanTimer(), and stop().

8.24.4.6 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

8.25 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

8.25.1 Detailed Description

Definition at line 36 of file kbmode.h.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 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:

8.25.2.2 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:

8.25.2.3 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),
```

```
    _name(other._name), _id(other._id),
    _light(new KbLight(this, keyMap, *other._light)), _bind(new
   KbBind(this, parent, keyMap, *other._bind)), _perf(new KbPerf(this, *other.
   _perf)),
    _needsSave(true)
{
    connect(_light, SIGNAL(updated()), this, SLOT(doUpdate()));
}
```

Here is the call graph for this function:

8.25.2.4 KbMode::~KbMode ( )

Definition at line 89 of file kbmode.cpp.

```
89                  {
90 }
```

8.25.3 Member Function Documentation

8.25.3.1 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:

8.25.3.2 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:

8.25.3.3 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:

8.25.3.4 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:

8.25.3.5 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:

8.25.3.6 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:

8.25.3.7 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:

8.25.3.8 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"; }
```

8.25.3.9 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:

8.25.3.10 void KbMode::newId ( )

Definition at line 52 of file kbmode.cpp.

References _id, _light, _needsSave, 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:

8.25.3.11 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:

8.25.3.12 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:

8.25.3.13 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:

8.25.3.14 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:

8.25.3.15 void KbMode::updated ( ) [signal]

Referenced by doUpdate(), and KbMode().

Here is the caller graph for this function:

8.25.4 Field Documentation

8.25.4.1 KbBind* KbMode::_bind [private]

Definition at line 78 of file kbmode.h.

Referenced by bind(), KbMode(), keyMap(), needsSave(), save(), and setNeedsUpdate().

8.25.4.2 UsbId KbMode::_id [private]

Definition at line 75 of file kbmode.h.

Referenced by id(), KbMode(), newId(), and save().

8.25.4.3 KbLight* KbMode::_light [private]

Definition at line 77 of file kbmode.h.

Referenced by KbMode(), keyMap(), light(), needsSave(), newId(), and save().

8.25.4.4 QString KbMode::_name [private]

Definition at line 74 of file kbmode.h.

Referenced by KbMode(), name(), and save().

8.25.4.5 bool KbMode::_needsSave [private]

Definition at line 81 of file kbmode.h.

Referenced by id(), keyMap(), name(), needsSave(), newId(), save(), and setNeedsSave().

8.25.4.6 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

8.26 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)
- QPoint curDpi () const
- void curDpi (const QPoint &newDpi)
- void curDpi (int newDpi)
- int curDpiIdx () const
- void curDpiIdx (int newIdx)
- void dpiUp ()
- void dpiDown ()
- 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, bool force=false, bool saveCustomDpi=true)
- 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 dpiCurIdx
- QColor dpiClr [DPI_COUNT+1]
- bool dpiOn [DPI_COUNT]
- int dpiLastIdx
- 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

8.26.1 Detailed Description

Definition at line 15 of file kbperf.h.

8.26.2 Member Enumeration Documentation

8.26.2.1 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      };
```

8.26.2.2 enum KbPerf::i_hw

Enumerator

NONE

NORMAL

ON

OFF

Definition at line 98 of file kbperf.h.

```
98              {
99          NONE = -1,  // For non-hardware indicators
100          NORMAL,
101          ON,
102          OFF
103     };
```

8.26.2.3 enum KbPerf::indicator

Enumerator

NUM

CAPS

SCROLL

HW_IMAX

MODE

MACRO

LIGHT

LOCK

MUTE

Definition at line 85 of file kbperf.h.

```
85                  {
86          // Hardware
87          NUM,
88          CAPS,
89          SCROLL, HW_IMAX = SCROLL,
90          // Software
91          MODE,
92          MACRO,
93          LIGHT,
94          LOCK,
95          MUTE
96      };
```

8.26.3 Constructor & Destructor Documentation

8.26.3.1 KbPerf::KbPerf (KbMode * *parent*) [explicit]

Definition at line 13 of file kbperf.cpp.

References CAPS, DPI_COUNT, dpiClr, dpiCurIdx, dpiCurX, dpiCurY, dpiLastIdx, dpiOn, dpiX, dpiY, HW_I_COUNT, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, LOCK, MACRO, MODE, MUTE, muteNAColor, NORMAL, NUM, and SCROLL.

```
                                :
    QObject(parent), runningPushIdx(1),
    _iOpacity(1.f), _dpiIndicator(true), _liftHeight(MEDIUM), _angleSnap(false),
    _needsUpdate(true), _needsSave(true) {
    // Default DPI settings
    dpiX[0] = dpiY[0] = 400;
    dpiX[1] = dpiY[1] = 450;
    dpiX[2] = dpiY[2] = 800;
    dpiX[3] = dpiY[3] = 1500;
    dpiX[4] = dpiY[4] = 3000;
    dpiX[5] = dpiY[5] = 6000;
    dpiClr[0] = QColor(255, 0, 0);
    dpiClr[1] = QColor(255, 192, 0);
    dpiClr[2] = QColor(255, 255, 0);
    dpiClr[3] = QColor(0, 255, 0);
    dpiClr[4] = QColor(0, 255, 255);
    dpiClr[5] = QColor(255, 255, 255);
    dpiClr[6] = QColor(192, 192, 192);
    for(int i = 0; i < DPI_COUNT; i++)
        dpiOn[i] = true;
    dpiLastIdx = dpiCurIdx = 3;
    dpiCurX = dpiX[dpiCurIdx];
    dpiCurY = dpiY[dpiCurIdx];
    // Default indicators
    iColor[NUM][0] = iColor[CAPS][0] = iColor[SCROLL][0] = QColor(0, 255, 0);   // Lock lights: on = green, off = black
    iColor[NUM][1] = iColor[CAPS][1] = iColor[SCROLL][1] = QColor(0, 0, 0);
    iColor[MACRO][0] = QColor(255, 0, 0);                                       // Macro: on = red, off = black
    iColor[LOCK][0] = QColor(255, 255, 255);                                    // Win lock: on = white, off = black
    iColor[MODE][0] = iColor[MUTE][0] = QColor(255, 255, 255, 0);               // Mode, mute: on = transparent, off = black
    iColor[MODE][1] = iColor[MACRO][1] = iColor[LOCK][1] = iColor[MUTE][1] = muteNAColor = QColor(0, 0, 0);
    iColor[LIGHT][0] = QColor(255, 0, 0);       // Brightness: red, yellow, white
    iColor[LIGHT][1] = QColor(255, 255, 0);
    light100Color = QColor(255, 255, 255);
    for(int i = 0; i < HW_I_COUNT; i++){
        // Set all lock lights to HW mode
        iEnable[i] = false;
        hwIType[i] = NORMAL;
    }
    for(int i = HW_I_COUNT; i < I_COUNT; i++){
        // Turn all other indicators on
        iEnable[i] = true;
    }
}
```

8.26.3.2 KbPerf::KbPerf (KbMode * *parent*, const KbPerf & *other*)

Definition at line 57 of file kbperf.cpp.

References curDpi(), DPI_COUNT, dpiClr, dpiOn, dpiX, dpiY, hwIType, I_COUNT, iColor, iEnable, and pushedDpis.

```
                                                :
    QObject(parent), dpiCurX(other.dpiCurX), dpiCurY(other.dpiCurY), dpiCurIdx(other.dpiCurIdx), dpiLastIdx(other.dpiLastIdx), runningPushIdx(1),
    _iOpacity(other._iOpacity), light100Color(other.light100Color), muteNAColor(other.muteNAColor), _dpiIndicator(other._dpiIndicator),
    _liftHeight(other._liftHeight), _angleSnap(other._angleSnap),
    _needsUpdate(true), _needsSave(true) {
    memcpy(dpiX, other.dpiX, sizeof(dpiX));
    memcpy(dpiY, other.dpiY, sizeof(dpiY));
    for(int i = 0; i < DPI_COUNT + 1; i++)
        dpiClr[i] = other.dpiClr[i];
    memcpy(dpiOn, other.dpiOn, sizeof(dpiOn));
    for(int i = 0; i < I_COUNT; i++){
        iColor[i][0] = other.iColor[i][0];
        iColor[i][1] = other.iColor[i][1];
    }
    memcpy(iEnable, other.iEnable, sizeof(iEnable));
    memcpy(hwIType, other.hwIType, sizeof(hwIType));
    // Don't copy pushed DPI states. If the other mode has any, restore the original DPI
    if(!other.pushedDpis.isEmpty())
        curDpi(other.pushedDpis[0]);
}
```

Here is the call graph for this function:

8.26.4 Member Function Documentation

8.26.4.1 void KbPerf::_curDpi (const QPoint & *newDpi*) [private]

Definition at line 262 of file kbperf.cpp.

References _needsSave, _needsUpdate, DPI_COUNT, dpiCurIdx, dpiCurX, dpiCurY, dpiLastIdx, dpiX, and dpiY.

Referenced by curDpi(), popDpi(), and pushDpi().

```
262                                         {
263     dpiCurX = newDpi.x();
264     dpiCurY = newDpi.y();
265     // Set current/last index
266     dpiCurIdx = -1;
267     for(int i = 0; i < DPI_COUNT; i++){
268         if(dpiCurX == dpiX[i] && dpiCurY == dpiY[i]){
269             dpiCurIdx = i;
270             if(i != 0)
271                 dpiLastIdx = i;
272             break;
273         }
274     }
275     _needsUpdate = _needsSave = true;
276 }
```

Here is the caller graph for this function:

8.26.4.2 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:

8.26.4.3 void KbPerf::angleSnap (bool *newAngleSnap*)

Definition at line 368 of file kbperf.cpp.

References _angleSnap, _needsSave, and _needsUpdate.

```
368                                         {
369     _angleSnap = newAngleSnap;
370     _needsUpdate = _needsSave = true;
371 }
```

8.26.4.4 void KbPerf::applyIndicators (int *modeIndex*, const bool *indicatorState[HW_I_COUNT]*)

Definition at line 432 of file kbperf.cpp.

References _dpiIndicator, _iOpacity, bind(), CAPS, curDpiIdx(), 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().

```
432                                                                     {
433     light()->resetIndicators();
434     if(_iOpacity <= 0.f)
435         return;
436     if(_dpiIndicator){
437         // Set DPI indicator according to index
438         int index = curDpiIdx();
439         if(index == -1 || index > OTHER)
440             index = OTHER;
441         lightIndicator("dpi", dpiClr[index].rgba());
442     }
443     // KB indicators
444     if(iEnable[MODE]){
445         for(int i = 0; i < Kb::HWMODE_MAX; i++){
446             char name[4];
447             snprintf(name, sizeof(name), "m%d", i + 1);
448             if(modeIndex == i)
449                 lightIndicator(name, iColor[MODE][0].rgba());
450             else
451                 lightIndicator(name, iColor[MODE][1].rgba());
452         }
453     }
454     if(iEnable[MACRO])
455         lightIndicator("mr", iColor[MUTE][1].rgba());
456     if(iEnable[LIGHT]){
457         switch(light()->dimming()){
```

```
        case 0: // 100%
            lightIndicator("light", light100Color.rgba());
            break;
        case 1: // 67%
            lightIndicator("light", iColor[LIGHT][1].rgba());
            break;
        case 2: // 33%
        case 3: // light off
            lightIndicator("light", iColor[LIGHT][0].rgba());
            break;
        }
    }
    if(iEnable[LOCK]){
        if(bind()->winLock())
            lightIndicator("lock", iColor[LOCK][0].rgba());
        else
            lightIndicator("lock", iColor[LOCK][1].rgba());
    }
    if(iEnable[MUTE]){
        switch(getMuteState()){
        case MUTED:
            lightIndicator("mute", iColor[MUTE][0].rgba());
            break;
        case UNMUTED:
            lightIndicator("mute", iColor[MUTE][1].rgba());
            break;
        default:
            lightIndicator("mute", muteNAColor.rgba());
            break;
        }
    }
    // Lock lights
    if(iEnable[NUM])
        lightIndicator("numlock", indicatorState[0] ? iColor[NUM][0].rgba() :
    iColor[NUM][1].rgba());
    if(iEnable[CAPS])
        lightIndicator("caps", indicatorState[1] ? iColor[CAPS][0].rgba() :
    iColor[CAPS][1].rgba());
    if(iEnable[SCROLL])
        lightIndicator("scroll", indicatorState[2] ? iColor[SCROLL][0].rgba() :
    iColor[SCROLL][1].rgba());
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.5 KbBind * KbPerf::bind ( ) const [private]

Definition at line 100 of file kbperf.cpp.

References KbMode::bind(), and modeParent().

Referenced by applyIndicators().

```
100                                 {
101     return modeParent()->bind();
102 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.6 QPoint KbPerf::curDpi ( ) const [inline]

Definition at line 57 of file kbperf.h.

References dpiCurX, and dpiCurY.

Referenced by curDpiIdx(), KbPerf(), load(), operator=(), and pushDpi().

```
57 { return QPoint(dpiCurX, dpiCurY); }
```

Here is the caller graph for this function:

8.26.4.7 void KbPerf::curDpi (const QPoint & *newDpi*)

Definition at line 278 of file kbperf.cpp.

References _curDpi(), popDpi(), and pushedDpis.

```
278                                     {
279     while(pushedDpis.count() > 0)
280         popDpi(pushedDpis.keys().last());
281     _curDpi(newDpi);
282 }
```

Here is the call graph for this function:

8.26.4.8 void KbPerf::curDpi (int *newDpi*) [inline]

Definition at line 59 of file kbperf.h.

References curDpi().

Referenced by curDpi().

```
59 { curDpi(QPoint(newDpi, newDpi)); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.9 int KbPerf::curDpiIdx ( ) const [inline]

Definition at line 61 of file kbperf.h.

References dpiCurIdx.

Referenced by applyIndicators(), dpiDown(), dpiUp(), and Kb::readNotify().

```
61 { return dpiCurIdx; }
```

Here is the caller graph for this function:

8.26.4.10 void KbPerf::curDpiIdx (int *newIdx*) [inline]

Definition at line 62 of file kbperf.h.

References curDpi(), and dpi().

```
62 { curDpi(dpi(newIdx)); }
```

Here is the call graph for this function:

8.26.4.11 void KbPerf::didLoad ( ) [signal]

Referenced by load().

Here is the caller graph for this function:

8.26.4.12 QPoint KbPerf::dpi (int *index*) const [inline]

Definition at line 48 of file kbperf.h.

References DPI_COUNT, dpiX, and dpiY.

Referenced by MPerfWidget::boxXChanged(), MPerfWidget::boxYChanged(), curDpiIdx(), KeyAction::keyEvent(), Kb::readNotify(), RebindWidget::setBind(), MPerfWidget::setPerf(), MPerfWidget::sliderXMoved(), MPerfWidget::sliderYMoved(), and sniperDpi().

```
48 { if(index < 0 || index >= DPI_COUNT) return QPoint(); return QPoint(
      dpiX[index], dpiY[index]); }
```

Here is the caller graph for this function:

8.26.4.13 void KbPerf::dpi (int *index*, const QPoint & *newValue*)

Definition at line 249 of file kbperf.cpp.

References _needsSave, _needsUpdate, DPI_COUNT, dpiCurIdx, dpiCurX, dpiCurY, dpiX, and dpiY.

```
249                                                 {
250     if(index < 0 || index >= DPI_COUNT)
251         return;
252     dpiX[index] = newValue.x();
253     dpiY[index] = newValue.y();
254     // Update current DPI if needed
255     if(dpiCurIdx == index){
256         dpiCurX = newValue.x();
257         dpiCurY = newValue.y();
258     }
259     _needsUpdate = _needsSave = true;
260 }
```

8.26.4.14 void KbPerf::dpi (int *index*, int *newValue*) [inline]

Definition at line 53 of file kbperf.h.

References dpi().

Referenced by dpi().

```
53 { dpi(index, QPoint(newValue, newValue)); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.15 QColor KbPerf::dpiColor (int *index*) const [inline]

Definition at line 82 of file kbperf.h.

References dpiClr.

Referenced by MPerfWidget::colorChanged(), Kb::readNotify(), MPerfWidget::setPerf(), and update().

```
82 { return dpiClr[index]; }
```

Here is the caller graph for this function:

8.26.4.16 void KbPerf::dpiColor (int *index*, const QColor & *newColor*) [inline]

Definition at line 83 of file kbperf.h.

References _needsSave, _needsUpdate, and dpiClr.

```
83 { dpiClr[index] = newColor; _needsUpdate = _needsSave = true; }
```

8.26.4.17 void KbPerf::dpiDown ( )

Definition at line 319 of file kbperf.cpp.

References curDpiIdx(), dpiOn, and SNIPER.

Referenced by KeyAction::keyEvent().

```
319                     {
320     int idx = curDpiIdx();
321     do {
```

```
        idx--;
        if(idx <= SNIPER)
            return;
    } while(!dpiOn[idx]);
    curDpiIdx(idx);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.18 bool KbPerf::dpiEnabled (int *index*) const [inline]

Definition at line 66 of file kbperf.h.

References dpiOn.

Referenced by MPerfWidget::enableChanged(), Kb::readNotify(), and MPerfWidget::setPerf().

```
66 { return dpiOn[index]; }
```

Here is the caller graph for this function:

8.26.4.19 void KbPerf::dpiEnabled (int *index*, bool *newEnabled*) [inline]

Definition at line 67 of file kbperf.h.

References _needsSave, _needsUpdate, and dpiOn.

```
67 { if(index <= 0) return; dpiOn[index] = newEnabled; _needsUpdate =
     _needsSave = true; }
```

8.26.4.20 bool KbPerf::dpiIndicator ( ) const [inline]

Definition at line 79 of file kbperf.h.

References _dpiIndicator.

Referenced by MPerfWidget::on_indicBox_clicked(), and MPerfWidget::setPerf().

```
79 { return _dpiIndicator; }
```

Here is the caller graph for this function:

8.26.4.21 void KbPerf::dpiIndicator (bool *newDpiIndicator*) [inline]

Definition at line 80 of file kbperf.h.

References _dpiIndicator, and _needsSave.

```
80 { _dpiIndicator = newDpiIndicator; _needsSave = true; }
```

8.26.4.22 void KbPerf::dpiUp ( )

Definition at line 308 of file kbperf.cpp.

References curDpiIdx(), DPI_COUNT, and dpiOn.

Referenced by KeyAction::keyEvent().

```
308                 {
309     // Scroll past disabled DPIs and choose the next one up
310     int idx = curDpiIdx();
311     do {
312         idx++;
313         if(idx >= DPI_COUNT)
314             return;
315     } while(!dpiOn[idx]);
316     curDpiIdx(idx);
317 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.23 void KbPerf::getIndicator (indicator *index*, QColor & *color1*, QColor & *color2*, QColor & *color3*, bool & *software_enable*, i_hw & *hardware_enable*)

Definition at line 329 of file kbperf.cpp.

References HW_IMAX, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, MUTE, muteNAColor, and NONE.

Referenced by KPerfWidget::setPerf().

```
329
                                {
330     if(index < 0 || index >= I_COUNT)
331         return;
332     color1 = iColor[index][0];
333     color2 = iColor[index][1];
334     if(index == LIGHT)
335         color3 = light100Color;
336     else if(index == MUTE)
337         color3 = muteNAColor;
338     else
339         color3 = QColor();
340     software_enable = iEnable[index];
341     hardware_enable = (index <= HW_IMAX ? hwIType[index] : NONE);
342 }
```

Here is the caller graph for this function:

8.26.4.24 float KbPerf::iOpacity ( ) const [inline]

Definition at line 76 of file kbperf.h.

References _iOpacity.

Referenced by KPerfWidget::on_intensityBox_valueChanged(), MPerfWidget::on_spinBox_valueChanged(), KPerfWidget::setPerf(), and MPerfWidget::setPerf().

```
76 { return _iOpacity; }
```

Here is the caller graph for this function:

8.26.4.25 void KbPerf::iOpacity (float *newIOpacity*) [inline]

Definition at line 77 of file kbperf.h.

References _iOpacity, and _needsSave.

```
77 { _iOpacity = newIOpacity; _needsSave = true; }
```

8.26.4.26 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:

8.26.4.27 void KbPerf::liftHeight (height *newHeight*)

Definition at line 361 of file kbperf.cpp.

References _liftHeight, _needsSave, _needsUpdate, and HIGH.

```
361                                     {
362     if(newHeight < LOW || newHeight > HIGH)
363         return;
364     _liftHeight = newHeight;
365     _needsUpdate = _needsSave = true;
366 }
```

8.26.4.28 KbLight * KbPerf::light ( ) const [private]

Definition at line 104 of file kbperf.cpp.

References KbMode::light(), and modeParent().

Referenced by applyIndicators(), and lightIndicator().

```
104                                 {
105     return modeParent()->light();
106 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.29 void KbPerf::lightIndicator (const char * *name*, QRgb *rgba*) [private]

Definition at line 425 of file kbperf.cpp.

References _iOpacity, light(), and KbLight::setIndicator().

Referenced by applyIndicators().

```
425                                                     {
426     int a = round(qAlpha(rgba) * _iOpacity);
427     if(a <= 0)
428         return;
429     light()->setIndicator(name, qRgba(qRed(rgba), qGreen(rgba), qBlue(rgba), a));
430 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.30 void KbPerf::load (CkbSettings & *settings*)

Definition at line 108 of file kbperf.cpp.

References _angleSnap, _dpiIndicator, _iOpacity, _liftHeight, _needsSave, CkbSettings::contains(), CkbSettings::containsGroup(), curDpi(), didLoad(), DPI_COUNT, dpiClr, dpiLastIdx, 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().

```
                                        {
    pushedDpis.clear();
    runningPushIdx = 1;
    _needsSave = false;
    bool readIndicators = true;
    if(!settings.containsGroup("Performance/Indicators")){
        // Read old indicator settings from the lighting group, if present
        // (ckb <= v0.2.0)
        SGroup group(settings, "Lighting");
        if(settings.contains("InactiveIndicators")){
            bool inOk = false;
            int inactive = settings.value("InactiveIndicators").toInt(&inOk);
            if(!inOk || inactive > 2)
                inactive = 2;
            if(inactive == 1)
                _iOpacity = 0.75f;
            else if(inactive == 0)
                _iOpacity = 0.5f;
            else if(inactive < 0){
                // Indicators disabled
                iEnable[MODE] = iEnable[MACRO] = iEnable[
LIGHT] = iEnable[LOCK]  = iEnable[MUTE] = false;
            }
            bool showMute = (settings.value("ShowMute").toInt(&inOk) != 0);
            if(inOk && !showMute)
                iEnable[MUTE] = false;
            readIndicators = false;
        }
    }
    SGroup group(settings, "Performance");
    // Read DPI settings
    {
        SGroup group(settings, "DPI");
        for(int i = 0; i < DPI_COUNT; i++){
            QString iStr = QString::number(i);
            QPoint value = settings.value(iStr).toPoint();
            if(value.isNull())
                continue;
            dpiX[i] = value.x(); dpiY[i] = value.y();
            QColor color = settings.value(iStr + "RGB").toString();
            if(color.isValid())
                dpiClr[i] = color;
            if(i != 0)
                dpiOn[i] = !settings.value(iStr + "Disabled").toBool();
        }
        QColor color = settings.value("6RGB").toString();
        if(color.isValid())
            dpiClr[OTHER] = color;
        if(settings.contains("LastIdx")){
            dpiLastIdx = settings.value("LastIdx").toInt();
            if(dpiLastIdx >= DPI_COUNT || dpiLastIdx < 0)
                dpiLastIdx = 1;
        }
        QPoint value = settings.value("Current").toPoint();
        if(!value.isNull())
            curDpi(value);
    }
    // Read misc. mouse settings
    _liftHeight = (height)settings.value("LiftHeight").toInt();
    if(_liftHeight < LOW || _liftHeight > HIGH)
        _liftHeight = MEDIUM;
    _angleSnap = settings.value("AngleSnap").toBool();
    if(settings.contains("NoIndicator")){
        // ckb <= v0.2.0
        _dpiIndicator = !settings.value("NoIndicator").toBool();
    } else {
        _dpiIndicator = settings.value("Indicators/DPI", true).toBool();
    }
    // Read indicator settings
    if(readIndicators){
        SGroup group(settings, "Indicators");
        _iOpacity = settings.value("Opacity", 100).toInt() / 100.f;
        for(int i = 0; i < I_COUNT; i++){
            SGroup group(settings, QString::number(i));
            QColor color = settings.value("RGB0").toString();
            if(color.isValid())
                iColor[i][0] = color;
            color = settings.value("RGB1").toString();
            if(color.isValid())
                iColor[i][1] = color;
            if(i == LIGHT){
                color = settings.value("RGB2").toString();
                if(color.isValid())
                    light100Color = color;
            } else if(i == MUTE){
                color = settings.value("RGB2").toString();
                if(color.isValid())
```

```
                    muteNAColor = color;
            }
            if(i <= HW_IMAX){
                iEnable[i] = settings.value("Enable", false).toBool();
                hwIType[i] = (i_hw)settings.value("Hardware", (int)
    NORMAL).toInt();
            } else {
                iEnable[i] = settings.value("Enable", true).toBool();
            }
        }
    }
    emit didLoad();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.31 KbMode* KbPerf::modeParent ( ) const [inline],[private]

Definition at line 124 of file kbperf.h.

Referenced by bind(), and light().

```
124 { return (KbMode*)parent(); }
```

Here is the caller graph for this function:

8.26.4.32 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:

8.26.4.33 const KbPerf & KbPerf::operator= (const KbPerf & *rhs*)

Definition at line 78 of file kbperf.cpp.

References _angleSnap, _dpiIndicator, _iOpacity, _liftHeight, _needsSave, _needsUpdate, curDpi(), DPI_COUNT, dpiClr, dpiCurIdx, dpiCurX, dpiCurY, dpiLastIdx, dpiOn, dpiX, dpiY, hwIType, I_COUNT, iColor, iEnable, light100-Color, muteNAColor, pushedDpis, and runningPushIdx.

```
78                                                  {
79     dpiCurX = other.dpiCurX; dpiCurY = other.dpiCurY; dpiCurIdx = other.dpiCurIdx;
      dpiLastIdx = other.dpiLastIdx; runningPushIdx = 1;
80     _iOpacity = other._iOpacity; light100Color = other.light100Color;
      muteNAColor = other.muteNAColor; _dpiIndicator = other._dpiIndicator;
81     _liftHeight = other._liftHeight; _angleSnap = other._angleSnap;
82     _needsUpdate = true; _needsSave = true;
83     memcpy(dpiX, other.dpiX, sizeof(dpiX));
84     memcpy(dpiY, other.dpiY, sizeof(dpiY));
85     for(int i = 0; i < DPI_COUNT + 1; i++)
86         dpiClr[i] = other.dpiClr[i];
87     memcpy(dpiOn, other.dpiOn, sizeof(dpiOn));
88     for(int i = 0; i < I_COUNT; i++){
89         iColor[i][0] = other.iColor[i][0];
90         iColor[i][1] = other.iColor[i][1];
91     }
92     memcpy(iEnable, other.iEnable, sizeof(iEnable));
93     memcpy(hwIType, other.hwIType, sizeof(hwIType));
94     // Don't copy pushed DPI states. If the other mode has any, restore the original DPI
95     if(!other.pushedDpis.isEmpty())
96         curDpi(other.pushedDpis[0]);
97     return other;
98 }
```

Here is the call graph for this function:

8.26.4.34 void KbPerf::popDpi (quint64 *pushIdx*)

Definition at line 296 of file kbperf.cpp.

References _curDpi(), _needsSave, _needsUpdate, map_last, and pushedDpis.

Referenced by curDpi(), and KeyAction::keyEvent().

```
296                                 {
297     if(pushIdx == 0 || !pushedDpis.contains(pushIdx))
298         return;
299     pushedDpis.remove(pushIdx);
300     // Set the DPI to the last-pushed value still on the stack
301     _curDpi(map_last(pushedDpis));
302     // If all values have been popped, remove the original DPI
303     if(pushedDpis.count() == 1)
304         pushedDpis.clear();
305     _needsUpdate = _needsSave = true;
306 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.35 quint64 KbPerf::pushDpi (const QPoint & *newDpi*)

Definition at line 284 of file kbperf.cpp.

References _curDpi(), curDpi(), pushedDpis, and runningPushIdx.

Referenced by KeyAction::keyEvent(), and pushSniper().

```
284                                         {
285     if(pushedDpis.isEmpty())
286         // Push original DPI
287         pushedDpis[0] = curDpi();
288     quint64 index = runningPushIdx++;
289     if(runningPushIdx == 0)
290         runningPushIdx = 1;
291     pushedDpis[index] = newDpi;
292     _curDpi(newDpi);
293     return index;
294 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.36 quint64 KbPerf::pushDpi (int *newDpi*) [inline]

Definition at line 71 of file kbperf.h.

References pushDpi().

Referenced by pushDpi().

```
71 { return pushDpi(QPoint(newDpi, newDpi)); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.37 quint64 KbPerf::pushSniper ( ) [inline]

Definition at line 72 of file kbperf.h.

References pushDpi(), and sniperDpi().

Referenced by KeyAction::keyEvent().

```
72 { return pushDpi(sniperDpi()); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.38 void KbPerf::save (CkbSettings & *settings*)

Definition at line 207 of file kbperf.cpp.

References _angleSnap, _dpiIndicator, _liftHeight, _needsSave, DPI_COUNT, dpiClr, dpiCurX, dpiCurY, dpiLastIdx, dpiOn, dpiX, dpiY, HW_IMAX, hwIType, I_COUNT, iColor, iEnable, LIGHT, light100Color, MUTE, muteNAColor, OTHER, pushedDpis, and CkbSettings::setValue().

Referenced by KbMode::save().

```
                                    {
    _needsSave = false;
    SGroup group(settings, "Performance");
    {
        SGroup group(settings, "DPI");
        for(int i = 0; i < DPI_COUNT; i++){
            QString iStr = QString::number(i);
            settings.setValue(iStr, QPoint(dpiX[i], dpiY[i]));
            settings.setValue(iStr + "RGB", dpiClr[i].name(QColor::HexArgb));
            if(i != 0)
                settings.setValue(iStr + "Disabled", !dpiOn[i]);
        }
        settings.setValue("6RGB", dpiClr[OTHER].name(QColor::HexArgb));
        settings.setValue("LastIdx", dpiLastIdx);
        int curX = dpiCurX, curY = dpiCurY;
        // Ignore any pushed modes
        if(pushedDpis.count() > 0){
            curX = pushedDpis.value(0).x();
            curY = pushedDpis.value(0).y();
        }
        settings.setValue("Current", QPoint(curX, curY));
    }
    settings.setValue("LiftHeight", _liftHeight);
    settings.setValue("AngleSnap", _angleSnap);
    {
        SGroup group(settings, "Indicators");
        settings.setValue("DPI", _dpiIndicator);
        for(int i = 0; i < I_COUNT; i++){
            SGroup group(settings, QString::number(i));
            settings.setValue("RGB0", iColor[i][0].name(QColor::HexArgb));
            settings.setValue("RGB1", iColor[i][1].name(QColor::HexArgb));
            if(i == LIGHT)
                settings.setValue("RGB2", light100Color.name(QColor::HexArgb));
            else if(i == MUTE)
                settings.setValue("RGB2", muteNAColor.name(QColor::HexArgb));
            settings.setValue("Enable", iEnable[i]);
            if(i <= HW_IMAX)
                settings.setValue("Hardware", (int)hwIType[i]);
        }
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.39 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 344 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().

```
{
    if(index < 0 || index >= I_COUNT)
        return;
    iColor[index][0] = color1;
    iColor[index][1] = color2;
    if(index == LIGHT)
        light100Color = color3;
    else if(index == MUTE)
        muteNAColor = color3;
    iEnable[index] = software_enable;
    if(hardware_enable == NONE)
        hardware_enable = NORMAL;
    if(index <= HW_IMAX)
        hwIType[index] = hardware_enable;
    _needsUpdate = _needsSave = true;
}
```

Here is the caller graph for this function:

8.26.4.40 void KbPerf::setNeedsUpdate ( ) [inline]

Definition at line 113 of file kbperf.h.

References _needsUpdate.

Referenced by KbMode::setNeedsUpdate().

```
113 { _needsUpdate = true; }
```

Here is the caller graph for this function:

8.26.4.41 void KbPerf::settingsUpdated ( ) [signal]

Referenced by update().

Here is the caller graph for this function:

8.26.4.42 QPoint KbPerf::sniperDpi ( ) const [inline]

Definition at line 50 of file kbperf.h.

References dpi(), and SNIPER.

Referenced by pushSniper().

```
50 { return dpi(SNIPER); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.43 void KbPerf::sniperDpi (const QPoint & *newValue*) [inline]

Definition at line 51 of file kbperf.h.

References dpi(), and SNIPER.

```
51 { dpi(SNIPER, newValue); }
```

Here is the call graph for this function:

8.26.4.44 void KbPerf::sniperDpi (int *newValue*) [inline]

Definition at line 54 of file kbperf.h.

References sniperDpi().

Referenced by sniperDpi().

```
54 { sniperDpi(QPoint(newValue, newValue)); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.4.45 void KbPerf::update (QFile & *cmd*, bool *force* = `false`, bool *saveCustomDpi* = `true`)

Definition at line 373 of file kbperf.cpp.

References _angleSnap, _liftHeight, _needsUpdate, DPI_COUNT, dpiColor(), dpiCurIdx, dpiCurX, dpiCurY, dpiOn, dpiX, dpiY, HW_I_COUNT, hwIType, OFF, ON, and settingsUpdated().

Referenced by Kb::frameUpdate(), and Kb::hwSave().

```
{
    if(!force && !_needsUpdate)
        return;
    emit settingsUpdated();
    _needsUpdate = false;
    // Save DPI stage 0 (sniper)
    cmd.write(QString("dpi 0:%1,%2").arg(dpiX[0]).arg(dpiY[0]).toLatin1());
    // If the mouse is set to a custom DPI, save it in stage 1
    int stage = dpiCurIdx;
    if(stage < 0 && saveCustomDpi){
        stage = 1;
        cmd.write(QString(" 1:%1,%2").arg(dpiCurX).arg(dpiCurY).toLatin1());
    } else {
        // Otherwise, save stage 1 normally
        if(!dpiOn[1] && stage != 1)
            cmd.write(" 1:off");
        else
            cmd.write(QString(" 1:%1,%2").arg(dpiX[1]).arg(dpiY[1]).toLatin1());
    }
    // Save stages 1 - 5
    for(int i = 2; i < DPI_COUNT; i++){
        if(!dpiOn[i] && stage != i)
            cmd.write(QString(" %1:off").arg(i).toLatin1());
        else
        cmd.write(QString(" %1:%2,%3").arg(i).arg(dpiX[i]).arg(dpiY[i]).toLatin1());
    }
    // Save stage selection, lift height, and angle snap
    cmd.write(QString(" dpisel %1 lift %2 snap %3").arg(stage).arg(_liftHeight).arg(
_angleSnap ? "on" : "off").toLatin1());
    // Save DPI colors
    cmd.write(" rgb");
    for(int i = 0; i < DPI_COUNT; i++){
        QColor color = dpiColor(i);
        cmd.write(" dpi");
        char output[9];
        snprintf(output, sizeof(output), "%1d:%02x%02x%02x", i, color.red(), color.green(), color.blue());
        cmd.write(output);
    }
    // Enable indicator notifications
    cmd.write(" inotify all");
    // Set indicator state
    const char* iNames[HW_I_COUNT] = { "num", "caps", "scroll" };
    for(int i = 0; i < HW_I_COUNT; i++){
        if(hwIType[i] == ON)
            cmd.write(" ion ");
        else if(hwIType[i] == OFF)
            cmd.write(" ioff ");
        else
            cmd.write(" iauto ");
        cmd.write(iNames[i]);
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.26.5 Field Documentation

8.26.5.1 bool KbPerf::_angleSnap [private]

Definition at line 158 of file kbperf.h.

Referenced by angleSnap(), load(), operator=(), save(), and update().

8.26.5.2 bool KbPerf::_dpiIndicator [private]

Definition at line 154 of file kbperf.h.

Referenced by applyIndicators(), dpiIndicator(), load(), operator=(), and save().

8.26.5.3 float KbPerf::_iOpacity [private]

Definition at line 149 of file kbperf.h.

Referenced by applyIndicators(), iOpacity(), lightIndicator(), load(), and operator=().

8.26.5.4 height KbPerf::_liftHeight [private]

Definition at line 157 of file kbperf.h.

Referenced by liftHeight(), load(), operator=(), save(), and update().

8.26.5.5 bool KbPerf::_needsSave [private]

Definition at line 160 of file kbperf.h.

Referenced by _curDpi(), angleSnap(), dpi(), dpiColor(), dpiEnabled(), dpiIndicator(), iOpacity(), liftHeight(), load(), needsSave(), operator=(), popDpi(), save(), and setIndicator().

8.26.5.6 bool KbPerf::_needsUpdate [private]

Definition at line 160 of file kbperf.h.

Referenced by _curDpi(), angleSnap(), dpi(), dpiColor(), dpiEnabled(), liftHeight(), operator=(), popDpi(), setIndicator(), setNeedsUpdate(), and update().

8.26.5.7 const int KbPerf::DPI_COUNT = 6 [static]

Definition at line 46 of file kbperf.h.

Referenced by _curDpi(), dpi(), dpiUp(), KbPerf(), KeyAction::keyEvent(), load(), operator=(), Kb::readNotify(), save(), RebindWidget::setBind(), and update().

8.26.5.8 const int KbPerf::DPI_MAX = 12000 [static]

Definition at line 47 of file kbperf.h.

8.26.5.9 const int KbPerf::DPI_MIN = 100 [static]

Definition at line 47 of file kbperf.h.

Referenced by dpiExp(), and dpiLog().

8.26.5.10 QColor KbPerf::dpiClr[DPI_COUNT+1] [private]

Definition at line 135 of file kbperf.h.

Referenced by applyIndicators(), dpiColor(), KbPerf(), load(), operator=(), and save().

8.26.5.11 int KbPerf::dpiCurIdx [private]

Definition at line 134 of file kbperf.h.

Referenced by _curDpi(), curDpiIdx(), dpi(), KbPerf(), operator=(), and update().

8.26.5.12 int KbPerf::dpiCurX [private]

Definition at line 134 of file kbperf.h.

Referenced by _curDpi(), curDpi(), dpi(), KbPerf(), operator=(), save(), and update().

8.26.5.13 int KbPerf::dpiCurY [private]

Definition at line 134 of file kbperf.h.

Referenced by _curDpi(), curDpi(), dpi(), KbPerf(), operator=(), save(), and update().

8.26.5.14 int KbPerf::dpiLastIdx [private]

Definition at line 138 of file kbperf.h.

Referenced by _curDpi(), KbPerf(), load(), operator=(), and save().

8.26.5.15 bool KbPerf::dpiOn[DPI_COUNT] [private]

Definition at line 136 of file kbperf.h.

Referenced by dpiDown(), dpiEnabled(), dpiUp(), KbPerf(), load(), operator=(), save(), and update().

8.26.5.16 int KbPerf::dpiX[DPI_COUNT] [private]

Definition at line 132 of file kbperf.h.

Referenced by _curDpi(), dpi(), KbPerf(), load(), operator=(), save(), and update().

8.26.5.17 int KbPerf::dpiY[DPI_COUNT] [private]

Definition at line 133 of file kbperf.h.

Referenced by _curDpi(), dpi(), KbPerf(), load(), operator=(), save(), and update().

8.26.5.18 const int KbPerf::HW_I_COUNT = (int)HW_IMAX + 1 [static]

Definition at line 104 of file kbperf.h.

Referenced by KbPerf(), and update().

8.26.5.19 i_hw KbPerf::hwIType[HW_I_COUNT] [private]

Definition at line 153 of file kbperf.h.

Referenced by getIndicator(), KbPerf(), load(), operator=(), save(), setIndicator(), and update().

8.26.5.20 const int KbPerf::I_COUNT = (int)MUTE + 1 [static]

Definition at line 104 of file kbperf.h.

Referenced by getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

8.26.5.21 QColor KbPerf::iColor[I_COUNT][2] [private]

Definition at line 150 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

8.26.5.22 bool KbPerf::iEnable[I_COUNT] [private]

Definition at line 152 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

8.26.5.23 QColor KbPerf::light100Color [private]

Definition at line 151 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

8.26.5.24 QColor KbPerf::muteNAColor [private]

Definition at line 151 of file kbperf.h.

Referenced by applyIndicators(), getIndicator(), KbPerf(), load(), operator=(), save(), and setIndicator().

8.26.5.25 const int KbPerf::OTHER = DPI_COUNT [static]

Definition at line 81 of file kbperf.h.

Referenced by applyIndicators(), MPerfWidget::colorChanged(), load(), save(), and MPerfWidget::setPerf().

8.26.5.26 QMap<quint64, QPoint> KbPerf::pushedDpis [private]

Definition at line 142 of file kbperf.h.

Referenced by curDpi(), KbPerf(), load(), operator=(), popDpi(), pushDpi(), and save().

8.26.5.27 uint KbPerf::runningPushIdx [private]

Definition at line 143 of file kbperf.h.

Referenced by load(), operator=(), and pushDpi().

8.26.5.28 const int KbPerf::SNIPER = 0 [static]

Definition at line 46 of file kbperf.h.

Referenced by dpiDown(), and sniperDpi().

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

- src/ckb/kbperf.h
- src/ckb/kbperf.cpp

8.27 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

8.27.1 Detailed Description

Definition at line 11 of file kbprofile.h.

8.27.2 Member Typedef Documentation

8.27.2.1 typedef QList<KbMode*> KbProfile::ModeList

Definition at line 41 of file kbprofile.h.

8.27.3 Constructor & Destructor Documentation

8.27.3.1 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:

8.27.3.2 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 }
```

8.27.3.3 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(), 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:

8.27.4 Member Function Documentation

8.27.4.1 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:

8.27.4.2 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:

8.27.4.3 void KbProfile::currentMode (KbMode * *newCurrentMode*) [inline]

Definition at line 55 of file kbprofile.h.

References _currentMode, and _needsSave.

```
55 { _needsSave = true; _currentMode = newCurrentMode; }
```

8.27.4.4 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:

8.27.4.5 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:

8.27.4.6 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:

8.27.4.7 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:

8.27.4.8 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:

8.27.4.9 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:

8.27.4.10 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:

8.27.4.11 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:

8.27.4.12 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:

8.27.4.13 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:

8.27.4.14 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:

8.27.4.15 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:

8.27.4.16 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"; }
```

8.27.4.17 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:

8.27.4.18 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:

8.27.4.19 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:

8.27.4.20 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:

8.27.4.21 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:

8.27.4.22 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:

8.27.5 Field Documentation

8.27.5.1 KbMode* KbProfile::_currentMode [private]

Definition at line 58 of file kbprofile.h.

Referenced by currentMode(), KbProfile(), and save().

8.27.5.2 UsbId KbProfile::_id [private]

Definition at line 60 of file kbprofile.h.

Referenced by id(), KbProfile(), newId(), and save().

8.27.5.3 KeyMap KbProfile::_keyMap [private]

Definition at line 61 of file kbprofile.h.

Referenced by KbProfile(), and keyMap().

8.27.5.4 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().

8.27.5.5 QString KbProfile::_name [private]

Definition at line 59 of file kbprofile.h.

Referenced by KbProfile(), and name().

8.27.5.6 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

8.28 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

8.28.1 Detailed Description

Definition at line 11 of file kbprofiledialog.h.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 KbProfileDialog::KbProfileDialog (KbWidget * *parent* = 0) [explicit]

Definition at line 5 of file kbprofiledialog.cpp.

References profileList_reordered(), repopulate(), 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:

8.28.2.2 KbProfileDialog::~KbProfileDialog ( )

Definition at line 16 of file kbprofiledialog.cpp.

References ui.

```
16                                  {
17     delete ui;
18 }
```

8.28.3 Member Function Documentation

8.28.3.1 void KbProfileDialog::addNewProfileItem ( ) [private]

Definition at line 66 of file kbprofiledialog.cpp.

References NEW_FLAG, 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);
```

```
    item->setFlags((item->flags() | Qt::ItemIsEditable) & ~Qt::ItemIsDragEnabled & ~Qt::ItemIsDropEnabled);
    item->setData(NEW_FLAG, 1);
    QFont font = item->font();
    font.setItalic(true);
    item->setFont(font);
    item->setIcon(QIcon(":/img/icon_plus.png"));
    ui->profileList->addItem(item);
}
```

Here is the caller graph for this function:

8.28.3.2 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:

8.28.3.3 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(), Kb::profiles(), KbProfile::removeAll(), repopulate(), Kb::setCurrentProfile(), and ui.

```
                                                                                {
    QListWidgetItem* item = ui->profileList->itemAt(pos);
    KbProfile* currentProfile = device->currentProfile();
    if(!item || !currentProfile || item->data(GUID).toUuid() != currentProfile->id().guid)
        return;
    int index = device->indexOf(currentProfile);
    QList<KbProfile*> profiles = device->profiles();

    QMenu menu(this);
    QAction* rename = new QAction("Rename...", this);
    QAction* duplicate = new QAction("Duplicate", this);
    QAction* del = new QAction("Delete", this);
    bool canDelete = (profiles.count() > 1);
    if(!canDelete)
        // Can't delete the last profile on the device
        del->setEnabled(false);
    QAction* hwsave = new QAction("Save to Hardware", this);
    QAction* moveup = new QAction("Move Up", this);
    if(index == 0)
        moveup->setEnabled(false);
    QAction* movedown = new QAction("Move Down", this);
    if(index >= profiles.count() - 1)
        movedown->setEnabled(false);
    menu.addAction(rename);
    menu.addAction(duplicate);
    menu.addAction(del);
    menu.addSeparator();
    menu.addAction(hwsave);
    menu.addSeparator();
    menu.addAction(moveup);
    menu.addAction(movedown);
    QAction* result = menu.exec(QCursor::pos());
    if(result == rename){
        ui->profileList->editItem(item);
    } else if(result == duplicate){
        KbProfile* newProfile = device->newProfile(currentProfile);
        newProfile->newId();
        profiles.insert(index + 1, newProfile);
        device->profiles(profiles);
        device->setCurrentProfile(newProfile);
    } else if(result == del){
        if(!canDelete)
            return;
        profiles.removeAll(currentProfile);
        device->profiles(profiles);
        currentProfile->deleteLater();
        // Select next profile
        if(index < profiles.count())
            device->setCurrentProfile(profiles.at(index));
        else
            device->setCurrentProfile(profiles.last());
    } else if(result == hwsave){
        device->hwSave();
        // Refresh item icons
        int count = ui->profileList->count();
        for(int i = 0; i < count; i++){
            QListWidgetItem* item = ui->profileList->item(i);
            if(item->data(NEW_FLAG).toUInt() == 1)
                continue;
            KbProfile* profile = device->find(item->data(GUID).toUuid());
            item->setIcon(QIcon((profile == device->hwProfile()) ? ":/img/icon_profile_hardware.png" : ":/img/icon_profile.png"));
        }
    } else if(result == moveup){
        profiles.removeAll(currentProfile);
        profiles.insert(index - 1, currentProfile);
        device->profiles(profiles);
    } else if(result == movedown){
        profiles.removeAll(currentProfile);
        profiles.insert(index + 1, currentProfile);
        device->profiles(profiles);
    }
    repopulate();
}
```

Here is the call graph for this function:

8.28.3.4 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:

8.28.3.5 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(), 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:

8.28.3.6 void KbProfileDialog::profileList_reordered ( ) [private],[slot]

Definition at line 20 of file kbprofiledialog.cpp.

References KbProfile::append(), device, Kb::find(), GUID, 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:

8.28.3.7 void KbProfileDialog::repopulate ( ) [private]

Definition at line 48 of file kbprofiledialog.cpp.

References addNewProfileItem(), Kb::currentProfile(), device, UsbId::guid, GUID, Kb::hwProfile(), KbProfile::id(), KbProfile::name(), 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);
        item->setData(GUID, profile->id().guid);
        item->setFlags(item->flags() | Qt::ItemIsEditable);
        if(profile == device->currentProfile()){
            item->setSelected(true);
            current = item;
        }
        ui->profileList->addItem(item);
    }
    if(current)
        ui->profileList->setCurrentItem(current);
    addNewProfileItem();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.28.4 Field Documentation

8.28.4.1 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().

8.28.4.2 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().

8.28.4.3 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().

8.28.4.4 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

8.29 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

8.29.1 Detailed Description

Definition at line 15 of file kbwidget.h.

8.29.2 Constructor & Destructor Documentation

8.29.2.1 KbWidget::KbWidget (QWidget * *parent*, Kb * *_device*) [explicit]

Definition at line 15 of file kbwidget.cpp.

References device, devUpdate(), Kb::features, Kb::isKeyboard(), Kb::isMouse(), modeChanged(), modesList_reordered(), Kb::monochrome, profileChanged(), 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")){
```

```
        ui->pollLabel->hide();
        ui->pollLabel2->hide();
    }
    if(!device->features.contains("fwupdate")){
        ui->fwUpdButton->hide();
        ui->fwUpdLabel->hide();
        ui->fwUpdLayout->removeItem(ui->fwUpdLayout->itemAt(1));
    }
    // Set monochrome mode according to hardware
    if(device->monochrome)
        ui->lightWidget->setMonochrome();
}
```

Here is the call graph for this function:

8.29.2.2 KbWidget::~KbWidget ( )

Definition at line 60 of file kbwidget.cpp.

References ui.

```
60                  {
61      delete ui;
62  }
```

8.29.3 Member Function Documentation

8.29.3.1 void KbWidget::addNewModeItem ( ) [private],[slot]

Definition at line 136 of file kbwidget.cpp.

References 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:

8.29.3.2 void KbWidget::devUpdate ( ) [private],[slot]

Definition at line 300 of file kbwidget.cpp.

References device, Kb::firmware, Kb::pollrate, 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:

8.29.3.3 void KbWidget::modeChanged (bool *spontaneous* = `true`) [private],[slot]

Definition at line 148 of file kbwidget.cpp.

References Kb::currentBind(), Kb::currentLight(), currentMode, Kb::currentMode(), Kb::currentPerf(), Kb::currentProfile(), device, KbProfile::indexOf(), modeUpdate(), 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:

8.29.3.4 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:

8.29.3.5 void KbWidget::modesList_reordered ( ) [private],[slot]

Definition at line 177 of file kbwidget.cpp.

References Kb::currentProfile(), device, KbProfile::find(), GUID, modeIcon(), KbProfile::modes(), 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(modeIcon(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:

8.29.3.6 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:

8.29.3.7 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:

8.29.3.8 void KbWidget::on_fwUpdButton_clicked ( ) [private],[slot]

Definition at line 339 of file kbwidget.cpp.

References device, FwUpgradeDialog::exec(), Kb::features, Kb::firmware, 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:

8.29.3.9 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:

8.29.3.10 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:

8.29.3.11 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(), 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
```

```
        profileChanged();
        if(index < currentProfile->modeCount())
            device->setCurrentMode(currentProfile->modes()[index]);
        else
            device->setCurrentMode(currentProfile->modes().last());
    } else if(result == moveup){
        currentProfile->removeAll(currentMode);
        currentProfile->insert(index - 1, currentMode);
        // Update UI
        profileChanged();
        modeChanged(true);
    } else if(result == movedown){
        currentProfile->removeAll(currentMode);
        currentProfile->insert(index + 1, currentMode);
        // Update UI
        profileChanged();
        modeChanged(true);
    }
}
```

Here is the call graph for this function:

8.29.3.12 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:

8.29.3.13 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(), 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:

8.29.3.14 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().

```
                                                {
    if(index < 0)
        return;
    if(index >= device->profiles().count()){
        // "Manage profiles" option
        KbProfileDialog dialog(this);
        dialog.exec();
        updateProfileList();
        return;
    }
    device->setCurrentProfile(device->profiles()[index]);
    // Device will emit profileChanged() and modeChanged() signals to update UI
}
```

Here is the call graph for this function:

8.29.3.15 void KbWidget::on_tabWidget_currentChanged (int *index*) [private],[slot]

Definition at line 317 of file kbwidget.cpp.

References device, ui, and updateFwButton().

```
                                                {
    if(!device)
        return;
    if(index == ui->tabWidget->count() - 1){
        // Device tab
        updateFwButton();
    }
}
```

Here is the call graph for this function:

8.29.3.16 void KbWidget::profileChanged ( ) [private],[slot]

Definition at line 91 of file kbwidget.cpp.

References addNewModeItem(), currentMode, Kb::currentProfile(), device, UsbId::guid, GUID, KbMode::id(), modeIcon(), KbProfile::modes(), KbMode::name(), and ui.

Referenced by KbWidget(), on_hwSaveButton_clicked(), and on_modesList_customContextMenuRequested().

```
                                {
    // Rebuild mode list
    ui->modesList->clear();
    int i = 0;
    QListWidgetItem* current = 0;
    foreach(KbMode* mode, device->currentProfile()->modes()){
        QListWidgetItem* item = new QListWidgetItem(modeIcon(i), mode->name(), ui->modesList);
        item->setData(GUID, mode->id().guid);
        item->setFlags(item->flags() | Qt::ItemIsEditable);
        if(mode == currentMode){
            item->setSelected(true);
            current = item;
        }
        ui->modesList->addItem(item);
        i++;
    }
    if(current)
        ui->modesList->setCurrentItem(current);
    addNewModeItem();
    // Wait for modeChanged() to refresh the rest of the UI
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.29.3.17 void KbWidget::showFirstTab ( ) [slot]

Definition at line 64 of file kbwidget.cpp.

References ui.

```
64                              {
65     ui->tabWidget->setCurrentIndex(0);
66 }
```

8.29.3.18 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:

8.29.3.19 void KbWidget::showLastTab ( ) [slot]

Definition at line 68 of file kbwidget.cpp.

References ui.

Referenced by MainWindow::showFwUpdateNotification().

```
68                              {
69     ui->tabWidget->setCurrentIndex(ui->tabWidget->count() - 1);
70 }
```

Here is the caller graph for this function:

8.29.3.20 void KbWidget::updateFwButton ( )

Definition at line 326 of file kbwidget.cpp.

References device, Kb::features, Kb::firmware, 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:

8.29.3.21 void KbWidget::updateProfileList ( ) [private],[slot]

Definition at line 73 of file kbwidget.cpp.

References Kb::currentProfile(), device, Kb::hwProfile(), KbProfile::name(), Kb::profiles(), and ui.

Referenced by KbWidget(), on_hwSaveButton_clicked(), and on_profileBox_activated().

```
                                {
    // Clear profile list and rebuild
    KbProfile* hwProfile = device->hwProfile(), *currentProfile =
   device->currentProfile();
    ui->profileBox->clear();
    int i = 0;
    foreach(KbProfile* profile, device->profiles()){
        ui->profileBox->addItem((profile == hwProfile) ? QIcon(":/img/icon_profile_hardware.png") : QIcon
   (":/img/icon_profile.png"),
                                profile->name());
        if(profile == currentProfile)
            ui->profileBox->setCurrentIndex(i);
        i++;
    }
    ui->profileBox->addItem(QIcon(":/img/icon_blank.png"), "Manage profiles...");
    QFont font = ui->profileBox->font();
    font.setItalic(true);
    ui->profileBox->setItemData(ui->profileBox->count() - 1, font, Qt::FontRole);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.29.4 Field Documentation

8.29.4.1 KbMode* KbWidget::currentMode [private]

Definition at line 44 of file kbwidget.h.

Referenced by modeChanged(), on_modesList_customContextMenuRequested(), on_modesList_itemChanged(), and profileChanged().

8.29.4.2 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().

8.29.4.3 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().

8.29.4.4 bool KbWidget::hasShownNewFW

Definition at line 28 of file kbwidget.h.

Referenced by MainWindow::checkFwUpdates().

8.29.4.5 quint64 KbWidget::lastAutoSave [private]

Definition at line 42 of file kbwidget.h.

8.29.4.6 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().

8.29.4.7 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

8.30 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

8.30.1 Detailed Description

Definition at line 8 of file keymap.h.

8.30.2 Member Function Documentation

8.30.2.1 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:

8.30.2.2 Key::operator bool ( ) const [inline]

Definition at line 44 of file keymap.h.

References name.

```
44 { return name != 0; }
```

8.30.2.3 bool Key::operator! ( ) const [inline]

Definition at line 45 of file keymap.h.

```
45 { return !(bool)*this; }
```

8.30.2.4 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:

8.30.3 Field Documentation

8.30.3.1 const char* Key::_friendlyName

Definition at line 13 of file keymap.h.

Referenced by _patch(), and friendlyName().

8.30.3.2 const char* Key::_storageName

Definition at line 10 of file keymap.h.

Referenced by _patch(), and storageName().

8.30.3.3 bool Key::hasLed

Definition at line 19 of file keymap.h.

Referenced by ColorMap::init(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and KeyWidget::selectAll().

8.30.3.4 bool Key::hasScan

Definition at line 20 of file keymap.h.

Referenced by KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and KeyWidget::selectAll().

8.30.3.5 short Key::height

Definition at line 17 of file keymap.h.

Referenced by getMap(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and patchANSI().

8.30.3.6 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().

8.30.3.7 short Key::width

Definition at line 17 of file keymap.h.

Referenced by getMap(), KeyWidget::mouseMoveEvent(), KeyWidget::mousePressEvent(), KeyWidget::paintEvent(), and patchANSI().

8.30.3.8 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(), and patchANSI().

8.30.3.9 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

8.31 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 four elements, delimited by ":", we may assume, that is a structural correct macro action.
- 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).
- 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)
- 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_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

8.31.1 Detailed Description

Definition at line 13 of file keyaction.h.

8.31.2 Member Enumeration Documentation

8.31.2.1 enum KeyAction::Type

Enumerator

UNBOUND

NORMAL

SPECIAL

Definition at line 125 of file keyaction.h.

```
125             {
126         UNBOUND,
127         NORMAL,
128         SPECIAL,
129     };
```

8.31.3 Constructor & Destructor Documentation

8.31.3.1 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 }
```

8.31.3.2 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 }
```

8.31.3.3 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 }
```

8.31.3.4 KeyAction::KeyAction (const KeyAction & *rhs*) [inline],[private]

Definition at line 166 of file keyaction.h.

```
166 : QObject() {}
```

8.31.4 Member Function Documentation

8.31.4.1 void KeyAction::adjustDisplay ( )

Definition at line 445 of file keyaction.cpp.

Referenced by keyEvent().

```
445                             {
446 #ifdef USE_LIBX11
447     // Try to get the current display from the X server
448     char* display_name = XDisplayName(NULL);
449     if(!display_name)
450         return;
451     Display* display = XOpenDisplay(display_name);
452     if(!display)
453         return;
454     char* display_string = DisplayString(display);
455     if(!display_string || strlen(display_string) == 0){
```

```
        XCloseDisplay(display);
        return;
    }
    size_t envstr_size = strlen(display_string) + 4;
    char* envstr = new char[envstr_size];
    strncpy(envstr, display_string, envstr_size);
    envstr[envstr_size - 1] = 0;

    Window root_window = XRootWindow(display, DefaultScreen(display));
    Window root_window_ret, child_window_ret, window;
    XWindowAttributes attr;
    int root_x, root_y, win_x, win_y;
    unsigned int mask_ret;

    // Find the screen which currently has the mouse
    XQueryPointer(display, root_window, &root_window_ret, &child_window_ret, &root_x, &root_y, &win_x, &
  win_y, &mask_ret);
    if(child_window_ret == (Window)NULL)
        window = root_window_ret;
    else
        window = child_window_ret;
    XGetWindowAttributes(display, window,  &attr);

    char* ptr = strchr(envstr, ':');
    if(ptr){
        ptr = strchr(ptr, '.');
        if(ptr)
            *ptr = '\0';
        char buf[16];
        snprintf(buf, sizeof(buf), ".%i", XScreenNumberOfScreen(attr.screen));
        strncat(envstr, buf, envstr_size - 1 - strlen(envstr));

        // Update environment variable
        setenv("DISPLAY", envstr, 1);
    }

    delete[] envstr;
    XCloseDisplay(display);
#endif
}
```

Here is the caller graph for this function:

8.31.4.2 QString KeyAction::animAction (const QUuid & *guid*, bool *onlyOnce*, bool *stopOnRelease*) [static]

Definition at line 175 of file keyaction.cpp.

References ANIM_KRSTOP, and ANIM_ONCE.

Referenced by RebindWidget::applyChanges().

```
175                                                                                    {
176     int flags = (onlyOnce ? ANIM_ONCE : 0) | (stopOnRelease ?
  ANIM_KRSTOP : 0);
177     return "$anim:" + guid.toString() + QString("+%1").arg(flags);
178 }
```

Here is the caller graph for this function:

8.31.4.3 QUuid KeyAction::animInfo (bool & *onlyOnce,* bool & *stopOnRelease*) const

Definition at line 216 of file keyaction.cpp.

References _value, ANIM_KRSTOP, ANIM_ONCE, and isAnim().

Referenced by keyEvent(), and RebindWidget::setSelection().

```
216                                                                 {
217     if(!isAnim())
218         return QUuid();
219     QString param = _value.mid(6);
220     QStringList split = param.split("+");
221     if(split.length() < 2)
222         return QUuid();
223     QUuid id = split[0];
224     int flags = split[1].toInt();
225     onlyOnce = !!(flags & ANIM_ONCE);
226     stopOnRelease = !!(flags & ANIM_KRSTOP);
227     return id;
228 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.4 QString KeyAction::defaultAction (const QString & *key*) [static]

Definition at line 44 of file keyaction.cpp.

Referenced by KbBind::bindAction(), KbBind::defaultAction(), and KbBind::save().

```
                                                {
    // G1-G18 are unbound by default
    if(key.length() >= 2 && key[0] == 'g'
        && ((key.length() == 2 && key[1] >= '0' && key[1] <= '9')
        || (key.length() == 3 && key[1] == '1' && key[2] >= '0' && key[2] <= '8')))
        return "";
    // So are thumbgrid buttons
    if(key.startsWith("thumb"))
        return "";
    // TODO: default action for MR
    if(key == "mr")
        return "";
    // M1-M3 switch modes
    if(key == "m1")
        return "$mode:0";
    if(key == "m2")
        return "$mode:1";
    if(key == "m3")
        return "$mode:2";
    // Brightness and Win Lock are their own functions
    if(key == "light")
        return "$light:2";
    if(key == "lock")
        return "$lock:0";
    // DPI buttons
    if(key == "dpiup")
        return "$dpi:-2";
    if(key == "dpidn")
        return "$dpi:-1";
    if(key == "sniper")
        return "$dpi:0";
    // Everything else is a standard keypress
    return key;
}
```

Here is the caller graph for this function:

8.31.4.5 QString KeyAction::dpiAction (int *level*, int *customX* = 0, int *customY* = 0) [static]

Definition at line 153 of file keyaction.cpp.

References DPI_CUSTOM.

Referenced by RebindWidget::applyChanges().

```
                                                {
    QString action = tr("$dpi:%1").arg(level);
    if(level == DPI_CUSTOM)
        action += tr("+%1+%2").arg(customX).arg(customY);
    return action;
}
```

Here is the caller graph for this function:

8.31.4.6 int KeyAction::dpiInfo (QPoint & *custom*) const

Definition at line 202 of file keyaction.cpp.

References _value, DPI_CUSTOM, and isDPI().

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

```
                                        {
    if(!isDPI())
        return 0;
    QString param = _value.mid(5);
    QStringList lxy = param.split("+");
    int level = lxy[0].toInt();
    if(level == DPI_CUSTOM){
        if(lxy.length() != 3)
            return 0;
        custom = QPoint(lxy[1].toInt(), lxy[2].toInt());
    }
    return level;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.7 QString KeyAction::driverName ( ) const

Definition at line 230 of file keyaction.cpp.

References _value, and isSpecial().

Referenced by KbBind::update().

```
230                                     {
231     if(isSpecial())
232         return "";
233     return _value;
234 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.8 QString KeyAction::friendlyName (const KeyMap & *map*) const

Definition at line 79 of file keyaction.cpp.

References _value, DPI_CUSTOM, 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.

```
79                                                      {
80     if(_value.isEmpty())
81         return "Unbound";
82     QStringList parts = _value.split(":");
83     QString prefix = parts[0];
84     if(parts.length() < 2){
85         KeyMap::Layout layout = map.layout();
86         QString name = KeyMap::friendlyName(_value, layout);
87         if(name.isEmpty())
88             return "(Unknown)";
89         return name;
90     }
91     int suffix = parts[1].toInt();
92     if(prefix == "$mode"){
93         switch(suffix){
94         case MODE_PREV:
95         case MODE_PREV_WRAP:
96             return "Switch to previous mode";
97         case MODE_NEXT:
98         case MODE_NEXT_WRAP:
99             return "Switch to next mode";
100         default:
101             return tr("Switch to mode %1").arg(suffix + 1);
102         }
```

```
    } else if(prefix == "$dpi"){
        // Split off custom parameters (if any)
        int level = parts[1].split("+")[0].toInt();
        switch(level){
        case DPI_UP:
            return "DPI up";
        case DPI_DOWN:
            return "DPI down";
        case DPI_SNIPER:
            return "Sniper";
        case DPI_CUSTOM:{
            QPoint xy;
            dpiInfo(xy);
            return tr("DPI: %1, %2").arg(xy.x()).arg(xy.y());
        }
        default:
            return tr("DPI stage %1").arg(level);
        }
    } else if(prefix == "$light"){
        switch(suffix){
        case LIGHT_UP:
        case LIGHT_UP_WRAP:
            return "Brightness up";
        case LIGHT_DOWN:
        case LIGHT_DOWN_WRAP:
            return "Brightness down";
        }
    } else if(prefix == "$lock"){
        switch(suffix){
        case LOCK_TOGGLE:
            return "Toggle Windows lock";
        case LOCK_ON:
            return "Windows lock on";
        case LOCK_OFF:
            return "Windows lock off";
        }
    } else if(prefix == "$anim"){
        return "Start animation";
    } else if(prefix == "$program"){
        return "Launch program";
    } else if(prefix == "$macro"){
        return "Send G-key macro";
    }
    return "(Unknown)";
}
```

Here is the call graph for this function:

8.31.4.9 bool KeyAction::isAnim ( ) const [inline]

Definition at line 138 of file keyaction.h.

References _value.

Referenced by animInfo(), and RebindWidget::setSelection().

```
138 { return _value.startsWith("$anim:"); }
```

Here is the caller graph for this function:

8.31.4.10 bool KeyAction::isDPI ( ) const [inline]

Definition at line 141 of file keyaction.h.

References _value.

Referenced by dpiInfo(), isMouse(), and RebindWidget::setSelection().

```
141 { return _value.startsWith("$dpi:"); }
```

Here is the caller graph for this function:

8.31.4.11 bool KeyAction::isMacro ( ) const [inline]

Definition at line 139 of file keyaction.h.

References _value.

Referenced by isValidMacro(), macroDisplay(), and macroFullLine().

```
139 { return _value.startsWith("$macro:"); }
```

Here is the caller graph for this function:

8.31.4.12 bool KeyAction::isMedia ( ) const [inline]

Definition at line 135 of file keyaction.h.

References _value.

```
135 { return _value == "mute" || _value == "volup" || _value == "voldn" ||
      _value == "stop" || _value == "prev" || _value == "play" ||
      _value == "next"; }
```

8.31.4.13 bool KeyAction::isMouse ( ) const [inline]

Definition at line 142 of file keyaction.h.

References _value, isDPI(), and isNormal().

Referenced by RebindWidget::setSelection().

```
142 { 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:

8.31.4.14 bool KeyAction::isNormal ( ) const [inline]

Definition at line 132 of file keyaction.h.

References NORMAL, and type().

Referenced by isMouse(), RebindWidget::setBind(), and RebindWidget::setSelection().

```
132 { return type() == NORMAL; }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.15 bool KeyAction::isProgram ( ) const [inline]

Definition at line 137 of file keyaction.h.

References _value.

Referenced by programInfo(), and RebindWidget::setSelection().

```
137 { return _value.startsWith("$program:"); }
```

Here is the caller graph for this function:

8.31.4.16 bool KeyAction::isSpecial ( ) const [inline]

Definition at line 133 of file keyaction.h.

References SPECIAL, and type().

Referenced by driverName(), keyEvent(), and RebindWidget::setSelection().

```
133 { return type() == SPECIAL; }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.17 bool KeyAction::isUnbound ( ) const [inline]

Definition at line 131 of file keyaction.h.

References type(), and UNBOUND.

```
131 { return type() == UNBOUND; }
```

Here is the call graph for this function:

8.31.4.18 bool KeyAction::isValidMacro ( ) const [inline]

Returns

bool as true iff the macro definition contains all four elements.

Definition at line 54 of file keyaction.h.

References _value, and isMacro().

Referenced by macroContent(), macroDisplay(), macroLine(), RebindWidget::setSelection(), and KbBind::update().

```
                                        {
        if (isMacro()) {
            QStringList ret;
            ret =_value.split(":");
            return (ret.count() == 4);
        } else {
            return false;
        }
    }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.19 void KeyAction::keyEvent (KbBind * *bind*, bool *down*)

Definition at line 236 of file keyaction.cpp.

References _value, adjustDisplay(), animInfo(), KbProfile::currentMode(), Kb::currentProfile(), KbBind::devParent(), KbLight::dimming(), KbPerf::dpi(), KbPerf::DPI_COUNT, DPI_CUSTOM, DPI_DOWN, DPI_SNIPER, DPI_UP, 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().

```
                                        {
    // No need to respond to standard actions
    if(!isSpecial())
        return;
    QStringList parts = _value.split(":");
    if(parts.length() < 2)
        return;
    QString prefix = parts[0];
    int suffix = parts[1].toInt();
    if(prefix == "$mode"){
        if(!down)
```

```
            return;
        // Change mode
        Kb* device = bind->devParent();
        KbProfile* currentProfile = device->currentProfile();
        int mode = currentProfile->indexOf(currentProfile->currentMode());
        int modeCount = currentProfile->modeCount();
        switch(suffix){
        case MODE_PREV_WRAP:
            mode--;
            if(mode < 0)
                mode = modeCount - 1;
            break;
        case MODE_NEXT_WRAP:
            mode++;
            if(mode >= modeCount)
                mode = 0;
            break;
        case MODE_PREV:
            mode--;
            break;
        case MODE_NEXT:
            mode++;
            break;
        default:
            // Absolute
            mode = suffix;
            break;
        }
        if(mode < 0 || mode >= modeCount)
            return;
        device->setCurrentMode(currentProfile->modes()[mode]);
    } else if(prefix == "$dpi"){
        KbPerf* perf = bind->perf();
        int level = parts[1].split("+")[0].toInt();
        switch(level){
        case DPI_UP:
            if(!down)
                return;
            perf->dpiUp();
            break;
        case DPI_DOWN:
            if(!down)
                return;
            perf->dpiDown();
            break;
        case DPI_SNIPER:
            if(down)
                sniperValue = perf->pushSniper();
            else {
                perf->popDpi(sniperValue);
                sniperValue = 0;
            }
            break;
        case DPI_CUSTOM:{
            QPoint xy;
            dpiInfo(xy);
            if(xy.x() <= 0 || xy.y() <= 0)
                break;
            if(down)
                sniperValue = perf->pushDpi(xy);
            else {
                perf->popDpi(sniperValue);
                sniperValue = 0;
            }
            break;
        }
        default:
            if(level < 1 || level >= KbPerf::DPI_COUNT
                    || !down)
                return;
            perf->dpi(level);
            break;
        }
    } else if(prefix == "$light"){
        if(!down)
            return;
        // Change brightness
        KbLight* light = bind->light();
        int dim = light->dimming();
        switch(suffix){
        case LIGHT_UP:
            if(dim > 0)
                dim--;
            break;
        case LIGHT_DOWN:
            if(dim < KbLight::MAX_DIM)
                dim++;
```

```
            break;
        case LIGHT_UP_WRAP:
            dim--;
            if(dim < 0)
                dim = KbLight::MAX_DIM;
            break;
        case LIGHT_DOWN_WRAP:
            dim++;
            if(dim > KbLight::MAX_DIM)
                dim = 0;
            break;
        }
        light->dimming(dim);
    } else if(prefix == "$lock"){
        if(!down)
            return;
        // Change win lock
        switch(suffix){
        case LOCK_TOGGLE:
            bind->winLock(!bind->winLock());
            break;
        case LOCK_ON:
            bind->winLock(true);
            break;
        case LOCK_OFF:
            bind->winLock(false);
            break;
        }
    } else if(prefix == "$anim"){
        // Start animation
        bool onlyOnce = false, stopOnRelease = false;
        QUuid id = animInfo(onlyOnce, stopOnRelease);
        KbAnim* anim = bind->light()->findAnim(id);
        if(!anim)
            return;
        if(down){
            if(!onlyOnce || !anim->isActive())
                // If "only once" is enabled, don't start the animation when it's already running
                anim->trigger(QDateTime::currentMSecsSinceEpoch(), true);
        } else if(stopOnRelease){
            // Key released - stop animation
            anim->stop();
        }
    } else if(prefix == "$program"){
        // Launch program
        QString onPress, onRelease;
        int stop = programInfo(onPress, onRelease);
        // Stop running programs based on setting
        QProcess* process = 0;
        if(down){
            if(stop & PROGRAM_PR_KPSTOP){
                process = preProgram;
                if(process)
                    process->kill();
                process = 0;
            }
            if(stop & PROGRAM_RE_KPSTOP)
                process = relProgram;
        } else {
            if(stop & PROGRAM_PR_KRSTOP)
                process = preProgram;
        }
        if(process)
            process->kill();
        // Launch new process if requested
        QString& program = down ? onPress : onRelease;
        if(program.isEmpty())
            return;
        // Check if the program is running already. If so, don't start it again.
        process = down ? preProgram : relProgram;
        if(process){
            if(process->state() == QProcess::NotRunning)
                delete process;
            else
                return;
        }

        // Adjust the selected display.
        adjustDisplay();

        // Start the program. Wrap it around sh to parse arguments.
        if((down && (stop & PROGRAM_PR_MULTI))
                || (!down && (stop & PROGRAM_RE_MULTI))){
            // Multiple instances allowed? Start detached process
            QProcess::startDetached("sh", QStringList() << "-c" << program);
        } else {
            process = new QProcess(this);
```

```
            process->start("sh", QStringList() << "-c" << program);
            if(down)
                preProgram = process;
            else
                relProgram = process;
        }
    } else if (prefix == "$macro") {
        // Do nothing, because all work is done by the keyboard itself.
        // For now, there is no reason to react on G-key press or release.
        // If u find some reason, then here is the place for it.
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.20 void KeyAction::keyRelease (KbBind * *bind*)

8.31.4.21 QString KeyAction::lightAction (int *type* = LIGHT_UP_WRAP) [static]

Definition at line 160 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
160                                     {
161     return QString("$light:%1").arg(type);
162 }
```

Here is the caller graph for this function:

8.31.4.22 QString KeyAction::lockAction (int *type* = LOCK_TOGGLE) [static]

Definition at line 164 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
164                                     {
165     return QString("$lock:%1").arg(type);
166 }
```

Here is the caller graph for this function:

8.31.4.23 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 2-4 of a complete macro definition.

Returns

QString holding the complete G-Key macro definition (parts 1-4)

Definition at line 504 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
504                                             {
505     return QString ("$macro:%1").arg(macroDef);
506 }
```

Here is the caller graph for this function:

8.31.4.24 QString KeyAction::macroContent ( ) const [inline]

Returns

QString macroContent

Definition at line 83 of file keyaction.h.

References _value, and isValidMacro().

Referenced by RebindWidget::setSelection(), and KbBind::update().

```
83                                          {
84          return isValidMacro() ? _value.split(":")[1] : "";
85      }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.25 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

Definition at line 437 of file keyaction.cpp.

References _value, isMacro(), and isValidMacro().

Referenced by RebindWidget::setSelection().

```
437                                 {
438     qDebug() << "isMacro returns" << (isMacro() ? "true" : "false");
439     qDebug() << "isValidMacro returns" << (isValidMacro() ? "true" : "false");
440     QStringList ret =_value.split(":");
441     qDebug() << "Macro definition conains" << ret.count() << "elements";
442     qDebug() << "Macro definition is" << _value;
443 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.26 QString KeyAction::macroFullLine ( ) const [inline]

Returns

QString All 4 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:

8.31.4.27 QStringList KeyAction::macroLine ( ) const [inline]

Returns

QStringList returns the Macro Key Definition, Readble Macro String and Readable Macro Comment as QStringList.

Definition at line 70 of file keyaction.h.

References _value, and isValidMacro().

Referenced by RebindWidget::setSelection().

```
70                                          {
71          if (isValidMacro()) {
72              QStringList ret =_value.split(":");
73              ret.removeFirst();
74              return ret;
75          } else return QStringList();
76      }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.28 QString KeyAction::modeAction (int *mode*) [static]

Definition at line 149 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
{
    return QString("$mode:%1").arg(mode);
}
```

Here is the caller graph for this function:

8.31.4.29 static QString KeyAction::noAction ( ) [inline],[static]

Definition at line 25 of file keyaction.h.

```
{ return ""; }
```

8.31.4.30 KeyAction::operator QString ( ) const [inline]

Definition at line 20 of file keyaction.h.

References _value.

```
{ return _value; }
```

8.31.4.31 void KeyAction::operator= (const KeyAction & *rhs*) [inline],[private]

Definition at line 165 of file keyaction.h.

```
{}
```

8.31.4.32 QString KeyAction::programAction (const QString & *onPress*, const QString & *onRelease*, int *stop*) [static]

Definition at line 168 of file keyaction.cpp.

Referenced by RebindWidget::applyChanges().

```
{
    // URL-encode the commands and place them in the string (":" and "+" are both replaced, so they won't interfere)
    return "$program:" + QString::fromUtf8(QUrl::toPercentEncoding(onPress.trimmed())) + "+" + QString::fromUtf8(QUrl::toPercentEncoding(onRelease.trimmed())) + QString("+%1").arg(stop);
}
```

Here is the caller graph for this function:

8.31.4.33 int KeyAction::programInfo (QString & *onPress*, QString & *onRelease*) const

Definition at line 190 of file keyaction.cpp.

References _value, and isProgram().

Referenced by keyEvent(), and RebindWidget::setSelection().

```
190                                                                 {
191     if(!isProgram())
192         return 0;
193     QString param = _value.mid(9);
194     QStringList programs = param.split("+");
195     if(programs.length() != 3)
196         return 0;
197     onPress = QUrl::fromPercentEncoding(programs[0].toUtf8());
198     onRelease = QUrl::fromPercentEncoding(programs[1].toUtf8());
199     return programs[2].toInt();
200 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.31.4.34 QString KeyAction::specialInfo (int & *parameter*) const

Definition at line 180 of file keyaction.cpp.

References _value.

Referenced by RebindWidget::setSelection().

```
180                                                 {
181     QStringList list = _value.split(":");
182     if(list.length() < 2){
183         parameter = INT_MIN;
184         return "";
185     }
186     parameter = list[1].toInt();
187     return list[0].replace("$", "");
188 }
```

Here is the caller graph for this function:

8.31.4.35 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:

8.31.4.36 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:

8.31.5 Field Documentation

8.31.5.1 QString KeyAction::_value [private]

Definition at line 168 of file keyaction.h.

Referenced by animInfo(), dpiInfo(), driverName(), friendlyName(), isAnim(), isDPI(), isMacro(), isMedia(), isMouse(), isProgram(), isValidMacro(), keyEvent(), macroContent(), macroDisplay(), macroFullLine(), macroLine(), operator QString(), programInfo(), specialInfo(), type(), and value().

8.31.5.2 const int KeyAction::DPI_CUSTOM = 6 [static]

Definition at line 107 of file keyaction.h.

Referenced by dpiAction(), dpiInfo(), friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.3 const int KeyAction::DPI_DOWN = -1 [static]

Definition at line 106 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

8.31.5.4 const int KeyAction::DPI_SNIPER = 0 [static]

Definition at line 107 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

8.31.5.5 const int KeyAction::DPI_UP = -2 [static]

Definition at line 106 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

8.31.5.6 const int KeyAction::LIGHT_DOWN = 1 [static]

Definition at line 110 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.7 const int KeyAction::LIGHT_DOWN_WRAP = 3 [static]

Definition at line 111 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.8 const int KeyAction::LIGHT_UP = 0 [static]

Definition at line 110 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.9 const int KeyAction::LIGHT_UP_WRAP = 2 [static]

Definition at line 111 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.10 const int KeyAction::LOCK_OFF = 2 [static]

Definition at line 114 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

8.31.5.11 const int KeyAction::LOCK_ON = 1 [static]

Definition at line 114 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

8.31.5.12 const int KeyAction::LOCK_TOGGLE = 0 [static]

Definition at line 114 of file keyaction.h.

Referenced by friendlyName(), and keyEvent().

8.31.5.13 const int KeyAction::MODE_NEXT = -1 [static]

Definition at line 102 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.14 const int KeyAction::MODE_NEXT_WRAP = -3 [static]

Definition at line 103 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.15 const int KeyAction::MODE_PREV = -2 [static]

Definition at line 102 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.16 const int KeyAction::MODE_PREV_WRAP = -4 [static]

Definition at line 103 of file keyaction.h.

Referenced by friendlyName(), keyEvent(), and RebindWidget::setSelection().

8.31.5.17 QProcess* KeyAction::preProgram [private]

Definition at line 171 of file keyaction.h.

Referenced by keyEvent(), and ~KeyAction().

8.31.5.18 const int KeyAction::PROGRAM_PR_INDEF = 0x00 [static]

Definition at line 117 of file keyaction.h.

8.31.5.19 const int KeyAction::PROGRAM_PR_KPSTOP = 0x02 [static]

Definition at line 117 of file keyaction.h.

Referenced by keyEvent().

8.31.5.20 const int KeyAction::PROGRAM_PR_KRSTOP = 0x01 [static]

Definition at line 117 of file keyaction.h.

Referenced by keyEvent().

8.31.5.21 const int KeyAction::PROGRAM_PR_MULTI = 0x04 [static]

Definition at line 117 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), keyEvent(), and RebindWidget::setSelection().

8.31.5.22 const int KeyAction::PROGRAM_RE_INDEF = 0x00 [static]

Definition at line 118 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), and RebindWidget::setSelection().

8.31.5.23 const int KeyAction::PROGRAM_RE_KPSTOP = 0x20 [static]

Definition at line 118 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), keyEvent(), and RebindWidget::setSelection().

8.31.5.24 const int KeyAction::PROGRAM_RE_MULTI = 0x40 [static]

Definition at line 118 of file keyaction.h.

Referenced by RebindWidget::applyChanges(), keyEvent(), and RebindWidget::setSelection().

8.31.5.25 QProcess* KeyAction::relProgram [private]

Definition at line 172 of file keyaction.h.

Referenced by keyEvent(), and ~KeyAction().

8.31.5.26 quint64 KeyAction::sniperValue [private]

Definition at line 175 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

8.32 KeyMap Class Reference

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

Collaboration diagram for KeyMap:

Public Types

- enum Model {
  NO_MODEL = -1, K65, K70, K95,
  STRAFE, M65, SABRE, SCIMITAR,
  _MODEL_MAX }
- enum Layout {
  NO_LAYOUT = -1, DK, EU, EU_DVORAK,
  GB, GB_DVORAK, US, US_DVORAK,
  FR, DE, IT, NO,
  PL, MX, ES, SE,
  _LAYOUT_MAX }

Public Member Functions

- bool isISO () 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 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

8.32.1 Detailed Description

Definition at line 49 of file keymap.h.

8.32.2 Member Enumeration Documentation

8.32.2.1 enum KeyMap::Layout

Enumerator

NO_LAYOUT

DK

EU

EU_DVORAK

GB

GB_DVORAK

US

US_DVORAK

FR

DE

IT

NO

PL

MX

ES

SE

_LAYOUT_MAX

Definition at line 65 of file keymap.h.

```
65                  {
66          NO_LAYOUT = -1,
67          DK,                     // Danish
68          EU,                     // English (EU)
69          EU_DVORAK,
70          GB,                     // English (UK)
71          GB_DVORAK,
72          US,                     // English (US)
73          US_DVORAK,
74          FR,                     // French
75          DE,                     // German
76          IT,                     // Italian
77          NO,                     // Norwegian
78          PL,                     // Polish (identical to US)
79          MX,                     // Spanish (Mexico/Latin America)
80          ES,                     // Spanish (Spain)
81          SE,                     // Swedish
82          _LAYOUT_MAX
83      };
```

8.32.2.2 enum KeyMap::Model

Enumerator

NO_MODEL

K65

K70

K95

STRAFE

M65

SABRE

SCIMITAR

_MODEL_MAX

Definition at line 51 of file keymap.h.

```
51                {
52         NO_MODEL = -1,
53         // Keyboard models
54         K65,
55         K70,
56         K95,
57         STRAFE,
58         // Mouse models
59         M65,
60         SABRE,
61         SCIMITAR,
62         _MODEL_MAX
63     };
```

8.32.3 Constructor & Destructor Documentation

8.32.3.1 KeyMap::KeyMap ( )

Definition at line 583 of file keymap.cpp.

Referenced by friendlyName(), and fromName().

```
583                :
584      keyWidth(0), keyHeight(0),
585      keyModel(NO_MODEL), keyLayout(NO_LAYOUT)
586 {}
```

Here is the caller graph for this function:

8.32.3.2 KeyMap::KeyMap (Model *_keyModel,* Layout *_keyLayout*)

Definition at line 577 of file keymap.cpp.

```
577                                                :
578     _keys(getMap(_keyModel, _keyLayout)),
579     keyWidth(modelWidth(_keyModel)), keyHeight(
      modelHeight(_keyModel)),
580     keyModel(_keyModel), keyLayout(_keyLayout)
581 {}
```

8.32.4 Member Function Documentation

8.32.4.1 QStringList KeyMap::byPosition ( ) const

Definition at line 588 of file keymap.cpp.

References key(), Key::x, and Key::y.

Referenced by RebindWidget::setBind().

```
588                                     {
589     // Use QMaps to order the keys
590     QMap<int, QMap<int, QString> > ordered;
591     QHashIterator<QString, Key> i(*this);
592     while(i.hasNext()){
593         i.next();
594         const Key& key = i.value();
595         ordered[key.y][key.x] = i.key();
596     }
597     // Merge them into a single list
598     QStringList result;
599     QMapIterator<int, QMap<int, QString> > y(ordered);
600     while(y.hasNext()){
601         y.next();
602         QMapIterator<int, QString> x(y.value());
603         while(x.hasNext()){
604             x.next();
605             result << x.value();
606         }
607     }
608     return result;
609 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.2 bool KeyMap::contains (const QString & *name*) const [inline]

Definition at line 127 of file keymap.h.

References _keys.

Referenced by friendlyName(), and KbAnimWidget::on_keyButton_clicked().

```
127 { return _keys.contains(name); }
```

Here is the caller graph for this function:

8.32.4.3 uint KeyMap::count ( ) const [inline]

Definition at line 118 of file keymap.h.

References _keys.

Referenced by KeyWidget::map(), KbBindWidget::on_copyButton_clicked(), and KbBindWidget::on_resetButton_clicked().

```
118 { return _keys.count(); }
```

Here is the caller graph for this function:

8.32.4.4 QString KeyMap::friendlyName (const QString & *key*, Layout *layout* = US) [static]

Definition at line 611 of file keymap.cpp.

References contains(), K95, KeyMap(), M65, map(), SCIMITAR, and STRAFE.

Referenced by KeyAction::friendlyName(), and RebindWidget::RebindWidget().

```
611                                                         {
612     // Try K95 map first
613     // FIXME: This is an odd function and probably should be refactored
```

```
    // it would probably be best to remove the friendly names from the maps and have a completely separate name->friendlyName store
    KeyMap map(K95, layout);
    if(map.contains(key))
        return map[key].friendlyName();

    // The only key missing from it should be Fn, which is found on STRAFE
    map = KeyMap(STRAFE, layout);
    if(map.contains(key))
        return map[key].friendlyName();

    // Additionally, there are a handful of keys not present on any physical keyboard, but we need names for them
    if(key == "f13" || key == "f14" || key == "f15" || key == "f16" || key == "f17" || key == "f18" || key == "f19" || key == "f20")
        return key.toUpper();
    else if(key == "lightup")
        return "Screen Brightness Up";
    else if(key == "lightdn")
        return "Screen Brightness Down";
    else if(key == "eject" || key == "power")
        return key[0].toUpper() + key.mid(1);   // capitalize first letter

    // All other names are found on mice
    map = KeyMap(SCIMITAR, layout);
    if(map.contains(key))
        return map[key].friendlyName();
    map = KeyMap(M65, layout);
    if(map.contains(key))
        return map[key].friendlyName();

    // Not found at all
    return "";
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.5 KeyMap KeyMap::fromName (const QString & *name*) [static]

Definition at line 535 of file keymap.cpp.

References getLayout(), getModel(), and KeyMap().

Referenced by KbBind::load(), and KbLight::load().

```
535                                         {
536     QStringList list = name.trimmed().split(" ");
537     if(list.length() != 2)
538         return KeyMap();
539     return KeyMap(getModel(list[0]), getLayout(list[1]));
540 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.6 QString KeyMap::fromStorage (const QString & *storage*) [inline]

Definition at line 135 of file keymap.h.

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

```
135 { 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:

8.32.4.7 KeyMap::Layout KeyMap::getLayout (const QString & *name*) [static]

Definition at line 407 of file keymap.cpp.

References DE, DK, ES, EU, EU_DVORAK, FR, GB, GB_DVORAK, IT, MX, NO, NO_LAYOUT, PL, SE, US, and US_DVORAK.

Referenced by fromName(), SettingsWidget::on_layoutBox_activated(), SettingsWidget::SettingsWidget(), and strLayout().

```
407                                                 {
408     QString lower = name.toLower();
409     if(lower == "dk")
410         return DK;
411     if(lower == "eu")
412         return EU;
413     if(lower == "eu_dvorak")
414         return EU_DVORAK;
415     if(lower == "gb_dvorak")
416         return GB_DVORAK;
417     if(lower == "us")
418         return US;
419     if(lower == "us_dvorak")
420         return US_DVORAK;
421     if(lower == "fr")
422         return FR;
423     if(lower == "de")
424         return DE;
425     if(lower == "it")
426         return IT;
427     if(lower == "no")
428         return NO;
429     if(lower == "pl")
430         return PL;
431     if(lower == "mx")
432         return MX;
433     if(lower == "es")
434         return ES;
435     if(lower == "se")
436         return SE;
437     if(lower == "gb")
438         return GB;
439     return NO_LAYOUT;
440 }
```

Here is the caller graph for this function:

8.32.4.8 QString KeyMap::getLayout (KeyMap::Layout *layout*) [static]

Definition at line 442 of file keymap.cpp.

References DE, DK, ES, EU, EU_DVORAK, FR, GB, GB_DVORAK, IT, MX, NO, PL, SE, US, and US_DVORAK.

```
442                                                 {
443     switch(layout){
444     case DK:
445         return "dk";
446     case EU:
447         return "eu";
448     case EU_DVORAK:
449         return "eu_dvorak";
450     case GB:
451         return "gb";
452     case GB_DVORAK:
453         return "gb_dvorak";
454     case US:
455         return "us";
456     case US_DVORAK:
457         return "us_dvorak";
458     case FR:
459         return "fr";
460     case DE:
461         return "de";
462     case IT:
463         return "it";
464     case NO:
465         return "no";
466     case PL:
467         return "pl";
468     case MX:
469         return "mx";
470     case ES:
471         return "es";
472     case SE:
473         return "se";
474     default:
475         return "";
476     }
477 }
```

8.32.4.9 KeyMap::Model KeyMap::getModel (const QString & *name*) [static]

Definition at line 495 of file keymap.cpp.

References K65, K70, K95, M65, NO_MODEL, SABRE, SCIMITAR, and STRAFE.

Referenced by fromName(), Kb::Kb(), and strModel().

```
495                                                 {
496     QString lower = name.toLower();
497     if(lower == "k65")
498         return K65;
499     if(lower == "k70")
500         return K70;
501     if(lower == "k95")
502         return K95;
503     if(lower == "strafe")
504         return STRAFE;
505     if(lower == "m65")
506         return M65;
507     if(lower == "sabre")
508         return SABRE;
509     if(lower == "scimitar")
510         return SCIMITAR;
511     return NO_MODEL;
512 }
```

Here is the caller graph for this function:

8.32.4.10 QString KeyMap::getModel (KeyMap::Model *model*) [static]

Definition at line 514 of file keymap.cpp.

References K65, K70, K95, M65, SABRE, SCIMITAR, and STRAFE.

```
514                                                 {
515     switch(model){
516     case K65:
517         return "k65";
518     case K70:
519         return "k70";
520     case K95:
521         return "k95";
522     case STRAFE:
523         return "strafe";
524     case M65:
525         return "m65";
526     case SABRE:
527         return "sabre";
528     case SCIMITAR:
529         return "scimitar";
530     default:
531         return "";
532     }
533 }
```

8.32.4.11 uint KeyMap::height ( ) const [inline]

Definition at line 122 of file keymap.h.

References keyHeight.

Referenced by KeyWidget::drawInfo(), and KeyWidget::map().

```
122 { return keyHeight; }
```

Here is the caller graph for this function:

8.32.4.12 static bool KeyMap::isISO (Layout *layout*) [inline], [static]

Definition at line 87 of file keymap.h.

References PL, US, and US_DVORAK.

Referenced by KbBind::isISO(), and RebindWidget::setBind().

```
87 { return layout != US && layout != US_DVORAK && layout !=
      PL; }
```

Here is the caller graph for this function:

8.32.4.13 bool KeyMap::isISO ( ) const [inline]

Definition at line 88 of file keymap.h.

References isISO(), and keyLayout.

Referenced by getMap(), isISO(), Kb::Kb(), and Kb::updateLayout().

```
88 { return isISO(keyLayout); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.14 static bool KeyMap::isKeyboard (Model *model*) [inline],[static]

Definition at line 93 of file keymap.h.

References isMouse(), and NO_MODEL.

Referenced by KbBind::isKeyboard().

```
93 { return !isMouse(model) && model != NO_MODEL; }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.15 bool KeyMap::isKeyboard ( ) const [inline]

Definition at line 94 of file keymap.h.

References isKeyboard(), and keyModel.

Referenced by Kb::isKeyboard(), and isKeyboard().

```
94 { return isKeyboard(keyModel); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.16 static bool KeyMap::isMouse (Model *model*) [inline], [static]

Definition at line 95 of file keymap.h.

References M65, SABRE, and SCIMITAR.

Referenced by KbBind::isMouse(), KeyWidget::map(), KbLightWidget::newSelection(), and KeyWidget::paintEvent().

```
95 { return model == M65 || model == SABRE || model ==
     SCIMITAR; }
```

Here is the caller graph for this function:

8.32.4.17 bool KeyMap::isMouse ( ) const [inline]

Definition at line 96 of file keymap.h.

References isMouse(), and keyModel.

Referenced by isKeyboard(), Kb::isMouse(), isMouse(), and KeyWidget::paintEvent().

```
96 { return isMouse(keyModel); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.18 Key KeyMap::key (const QString & *name*) const [inline]

Definition at line 125 of file keymap.h.

References _keys.

Referenced by AnimScript::begin(), byPosition(), AnimScript::keypress(), KbBind::noAction(), operator[]() , KbBind::setAction(), and toStorage().

```
125 { Key empty = {0,0,0,0,0,0,0,0,0}; return _keys.value(name, empty); }
```

Here is the caller graph for this function:

8.32.4.19 QStringList KeyMap::keys ( ) const [inline]

Definition at line 131 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().

```
131 { return _keys.keys(); }
```

Here is the caller graph for this function:

8.32.4.20 Layout KeyMap::layout ( ) const [inline]

Definition at line 114 of file keymap.h.

References keyLayout.

Referenced by KeyAction::friendlyName(), KeyWidget::paintEvent(), and RebindWidget::setBind().

```
114 { return keyLayout; }
```

Here is the caller graph for this function:

8.32.4.21 QStringList KeyMap::layoutNames ( ) [static]

Definition at line 479 of file keymap.cpp.

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

```
479                                 {
480     return QStringList()
481             << "Danish"
482             << "English (ISO/European)" << "English (ISO/European, Dvorak)"
483             << "English (United Kingdom)" << "English (United Kingdom, Dvorak)"
484             << "English (United States)" << "English (United States, Dvorak)"
485             << "French"
486             << "German"
487             << "Italian"
488             << "Norwegian"
489             << "Polish"
490             << "Spanish (Latin America)"
491             << "Spanish (Spain)"
492             << "Swedish";
493 }
```

Here is the caller graph for this function:

8.32.4.22 KeyMap::Layout KeyMap::locale ( ) [static]

Definition at line 375 of file keymap.cpp.

References DE, DK, ES, FR, GB, IT, MX, NO, PL, SE, and US.

Referenced by SettingsWidget::SettingsWidget().

```
375                                 {
376     setlocale(LC_ALL, "");
377     QString loc = setlocale(LC_CTYPE, 0);
378     loc = loc.toLower().replace('_', '-');
379     if(loc.startsWith("dk-"))
380         return KeyMap::DK;
381     else if(loc.startsWith("fr-"))
382         return KeyMap::FR;
383     else if(loc.startsWith("de-"))
384         return KeyMap::DE;
385     else if(loc.startsWith("it-"))
386         return KeyMap::IT;
387     else if(loc.startsWith("pl-"))
388         return KeyMap::PL;
389     else if(loc.startsWith("no-"))
390         return KeyMap::NO;
391     else if(loc.startsWith("es-es"))
392         // Spain uses the ES layout
393         return KeyMap::ES;
394     else if(loc.startsWith("es-"))
395         // Other Spanish locales use MX
396         return KeyMap::MX;
397     else if(loc.startsWith("se-"))
398         return KeyMap::SE;
399     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"))
        // Most English locales use US
        return KeyMap::US;
    else
        // Default to GB
        return KeyMap::GB;
}
```

Here is the caller graph for this function:

8.32.4.23 const QHash<QString, Key>& KeyMap::map ( ) const [inline]

Definition at line 129 of file keymap.h.

References _keys.

Referenced by friendlyName().

```
129 { return _keys; }
```

Here is the caller graph for this function:

8.32.4.24 Model KeyMap::model ( ) const [inline]

Definition at line 113 of file keymap.h.

References keyModel.

Referenced by KeyWidget::mouseMoveEvent(), KeyWidget::paintEvent(), and KPerfWidget::setPerf().

```
113 { return keyModel; }
```

Here is the caller graph for this function:

8.32.4.25 int KeyMap::modelHeight (Model *model*) [static],[private]

Definition at line 561 of file keymap.cpp.

References K65, K70, K95, K95_HEIGHT, M65, M65_HEIGHT, SABRE, SCIMITAR, and STRAFE.

```
561                                     {
562     switch(model){
563     case K65:
564     case K70:
565     case K95:
566     case STRAFE:
567         return K95_HEIGHT;
568     case M65:
569     case SABRE:
570     case SCIMITAR:
571         return M65_HEIGHT;
572     default:
573         return 0;
574     }
575 }
```

8.32.4.26 int KeyMap::modelWidth (Model *model*) [static],[private]

Definition at line 542 of file keymap.cpp.

References K65, K65_WIDTH, K70, K70_WIDTH, K95, K95_WIDTH, KSTRAFE_WIDTH, M65, M65_WIDTH, SABRE, SCIMITAR, and STRAFE.

```
542                                     {
543     switch(model){
544     case K65:
545         return K65_WIDTH;
546     case K70:
547         return K70_WIDTH;
548     case K95:
549         return K95_WIDTH;
550     case STRAFE:
551         return KSTRAFE_WIDTH;
552     case M65:
553     case SABRE:
554     case SCIMITAR:
555         return M65_WIDTH;
556     default:
557         return 0;
558     }
559 }
```

8.32.4.27 QString KeyMap::name ( ) const [inline]

Definition at line 115 of file keymap.h.

References strLayout(), and strModel().

Referenced by KbBind::save(), KbLight::save(), and toStorage().

```
115 { return (strModel() + " " + strLayout()).toUpper(); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.28 KeyMap::operator const QHash< QString, Key > & ( ) const [inline]

Definition at line 130 of file keymap.h.

References _keys.

```
130 { return _keys; }
```

8.32.4.29 Key KeyMap::operator[] (const QString & *name*) const [inline]

Definition at line 126 of file keymap.h.

References key().

```
126 { return key(name); }
```

Here is the call graph for this function:

8.32.4.30 QList<Key> KeyMap::positions ( ) const [inline]

Definition at line 132 of file keymap.h.

References _keys.

Referenced by ColorMap::init(), and KeyWidget::selectAll().

```
132 { return _keys.values(); }
```

Here is the caller graph for this function:

8.32.4.31 QString KeyMap::strLayout ( ) const [inline]

Definition at line 106 of file keymap.h.

References getLayout(), and keyLayout.

Referenced by name().

```
106 { return getLayout(keyLayout); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.32 QString KeyMap::strModel ( ) const [inline]

Definition at line 110 of file keymap.h.

References getModel(), and keyModel.

Referenced by name().

```
110 { return getModel(keyModel); }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.32.4.33 QString KeyMap::toStorage (const QString & *name*) [inline]

Definition at line 134 of file keymap.h.

References key(), name(), and Key::storageName().

```
134 { const char* storage = key(name).storageName(); if(!storage) return
      name; return storage; }
```

Here is the call graph for this function:

8.32.4.34 uint KeyMap::width ( ) const [inline]

Definition at line 120 of file keymap.h.

References keyWidth.

Referenced by KeyWidget::drawInfo(), and KeyWidget::map().

```
120 { return keyWidth; }
```

Here is the caller graph for this function:

8.32.5 Field Documentation

8.32.5.1 QHash<QString, Key> KeyMap::_keys [private]

Definition at line 147 of file keymap.h.

Referenced by contains(), count(), key(), keys(), map(), operator const QHash< QString, Key > &(), and positions().

8.32.5.2 short KeyMap::keyHeight [private]

Definition at line 148 of file keymap.h.

Referenced by height().

8.32.5.3 Layout KeyMap::keyLayout [private]

Definition at line 150 of file keymap.h.

Referenced by isISO(), layout(), and strLayout().

8.32.5.4 Model KeyMap::keyModel [private]

Definition at line 149 of file keymap.h.

Referenced by isKeyboard(), isMouse(), model(), and strModel().

8.32.5.5 short KeyMap::keyWidth [private]

Definition at line 148 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

8.33 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

8.33.1 Detailed Description

Definition at line 11 of file keywidget.h.

8.33.2 Member Typedef Documentation

8.33.2.1 typedef QHash<QString, QString> KeyWidget::BindMap

Definition at line 30 of file keywidget.h.

8.33.3 Member Enumeration Documentation

8.33.3.1 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;
```

8.33.4 Constructor & Destructor Documentation

8.33.4.1 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 }
```

8.33.5 Member Function Documentation

8.33.5.1 const BindMap& KeyWidget::bindMap ( ) const [inline]

Definition at line 31 of file keywidget.h.

References _bindMap.

```
31 { return _bindMap; }
```

8.33.5.2 void KeyWidget::bindMap (const BindMap & *newBindMap*)

Definition at line 65 of file keywidget.cpp.

References _bindMap.

```
65                                                  {
66     _bindMap = newBindMap;
67     update();
68 }
```

8.33.5.3 void KeyWidget::clearAnimation ( )

Definition at line 637 of file keywidget.cpp.

References animation.

```
637                                 {
638     animation.fill(false);
639     update();
640 }
```

8.33.5.4 void KeyWidget::clearSelection ( )

Definition at line 610 of file keywidget.cpp.

References mouseDownMode, newSelection, NONE, selection, and selectionChanged().

```
610                                 {
611     selection.fill(false);
612     newSelection.fill(false);
613     mouseDownMode = NONE;
614     update();
615     emit selectionChanged(QStringList());
616 }
```

8.33.5.5 const QColorMap& KeyWidget::colorMap ( ) const [inline]

Definition at line 27 of file keywidget.h.

References _colorMap.

```
27 { return _colorMap; }
```

8.33.5.6 void KeyWidget::colorMap (const QColorMap & *newColorMap*)

Definition at line 52 of file keywidget.cpp.

References _colorMap.

```
52                                                  {
53     _colorMap = newColorMap;
54     update();
55 }
```

8.33.5.7 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.

```
57                                                                                  {
58     if(!isVisible())
59         return;
60     _displayColorMap = newDisplayMap;
61     _indicators = indicators;
62     update();
63 }
```

8.33.5.8 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, 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:

8.33.5.9 const KeyMap& KeyWidget::map ( ) const [inline]

Definition at line 24 of file keywidget.h.

References keyMap.

```
24 { return keyMap; }
```

8.33.5.10 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, 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:

8.33.5.11 bool KeyWidget::monochrome ( ) [inline]

Definition at line 20 of file keywidget.h.

References _monochrome.

```
20 { return _monochrome; }
```

8.33.5.12 void KeyWidget::monochrome (bool *newMono*) [inline]

Definition at line 21 of file keywidget.h.

References _monochrome.

```
21 { _monochrome = newMono; update(); }
```

8.33.5.13 void KeyWidget::mouseMoveEvent (QMouseEvent * *event*) [private]

Definition at line 480 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.

```
480                                                 {
481     event->accept();
482     QString tooltip;
483
484     // Find selection rectangle
485     mouseCurrentX = event->x();
486     mouseCurrentY = event->y();
487     float scale, offX, offY;
488     drawInfo(scale, offX, offY);
489     float mx = mouseCurrentX / scale - offX, my = mouseCurrentY / scale - offY;
490     float mx1, mx2, my1, my2;
491     if(mouseCurrentX >= mouseDownX){
492         mx1 = mouseDownX / scale - offX;
493         mx2 = mouseCurrentX / scale - offX;
494     } else {
495         mx1 = mouseCurrentX / scale - offX;
496         mx2 = mouseDownX / scale - offX;
497     }
498     if(mouseCurrentY >= mouseDownY){
499         my1 = mouseDownY / scale - offY;
500         my2 = mouseCurrentY / scale - offY;
501     } else {
502         my1 = mouseCurrentY / scale - offY;
503         my2 = mouseDownY / scale - offY;
504     }
505     // Clear new selection
506     if(mouseDownMode != NONE)
507         newSelection.fill(false);
508     // See if the event hit any keys
509     QHashIterator<QString, Key> k(keyMap);
510     uint i = -1;
511     while(k.hasNext()){
512         k.next();
513         i++;
514         const Key& key = k.value();
515         if((_rgbMode && !key.hasLed)
516                 || (!_rgbMode && !key.hasScan))
517             continue;
518         // Update tooltip with the mouse hover (if any), even if it's not selectable
519         if(fabs(key.x - mx) <= key.width / 2.f - 1.f && fabs(key.y - my) <= key.
    height / 2.f - 1.f
520                 && tooltip.isEmpty())
521             tooltip = key.friendlyName(false);
522         // on STRAFE Sidelights and indicators can't be assigned color the way other keys are colored
523         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
524             continue;
525         float kx1 = key.x - key.width / 2.f + 1.f;
526         float ky1 = key.y - key.height / 2.f + 1.f;
527         float kx2 = kx1 + key.width - 2.f;
528         float ky2 = ky1 + key.height - 2.f;
529         // If they overlap, add the key to the selection
530         if(!(mx1 >= kx2 || kx1 >= mx2)
531                 && !(my1 >= ky2 || ky1 >= my2)
532                 && mouseDownMode != NONE)
533             newSelection.setBit(i);
534     }
535
536     if(mouseDownMode != NONE)
537         update();
538     setToolTip(tooltip);
539 }
```

Here is the call graph for this function:

8.33.5.14 void KeyWidget::mousePressEvent (QMouseEvent * *event*) [private]

Definition at line 448 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.

```
448                                                 {
449     event->accept();
450     mouseDownMode = (event->modifiers() & Qt::AltModifier) ?
      SUBTRACT : (event->modifiers() & Qt::ShiftModifier) ? ADD : (event->modifiers() &
      Qt::ControlModifier) ? TOGGLE : SET;
451     mouseDownX = mouseCurrentX = event->x();
452     mouseDownY = mouseCurrentY = event->y();
453     // See if the event hit a key
454     float scale, offX, offY;
455     drawInfo(scale, offX, offY);
456     float mx = mouseCurrentX / scale - offX, my = mouseCurrentY / scale - offY;
457     QHashIterator<QString, Key> k(keyMap);
458     uint i = -1;
459     while(k.hasNext()){
460         k.next();
461         i++;
462         const Key& key = k.value();
463         if((_rgbMode && !key.hasLed)
464                 || (!_rgbMode && !key.hasScan))
465             continue;
466         if(fabs(key.x - mx) <= key.width / 2.f - 1.f && fabs(key.y - my) <= key.
      height / 2.f - 1.f){
467             // Sidelights can't have a color, but they can be toggled
468             if(!strcmp(key.name, "lsidel") || !strcmp(key.name, "rsidel")){
469                 emit sidelightToggled(); // get the kblightwidget to record it
470                 update();
471                 break;
472             }
473             newSelection.setBit(i);
474             update();
475             break;
476         }
477     }
478 }
```

Here is the call graph for this function:

8.33.5.15 void KeyWidget::mouseReleaseEvent (QMouseEvent * *event*) [private]

Definition at line 541 of file keywidget.cpp.

References ADD, keyMap, KeyMap::keys(), mouseDownMode, newSelection, NONE, selection, selectionChanged(), SET, SUBTRACT, and TOGGLE.

```
541                                                 {
542     event->accept();
543     if(mouseDownMode == NONE)
544         return;
545     // Apply the new selection
546     switch(mouseDownMode){
547     case SET:
548         selection = newSelection;
549         break;
550     case ADD:
551         selection |= newSelection;
552         break;
553     case SUBTRACT:
554         selection &= ~newSelection;
555         break;
556     case TOGGLE:
557         selection ^= newSelection;
558         break;
559     default:;
560     }
561     // Clear mousedown state.
562     newSelection.fill(false);
563     mouseDownMode = NONE;
564     // Emit signal with the names of the keys
565     QStringList selectedNames;
566     uint i = 0;
567     foreach(const QString& key, keyMap.keys()){
568         if(selection.testBit(i++))
569             selectedNames << key;
570     }
571     emit selectionChanged(selectedNames);
572     update();
573 }
```

Here is the call graph for this function:

8.33.5.16 void KeyWidget::paintEvent (QPaintEvent *) [private]

Attention: Missing code here. Please refer to html docs

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(), KbBind::globalRemap(), Key::hasLed, Key::hasScan, Key::height, KeyMap::isMouse(), KeyMap::K65, 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, Key::x, and Key::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         }
137         if(overlay){
```

```
            painter.setBrush(palette().brush(QPalette::Window));
            painter.drawRect(0, 0, width(), height());
            float oXScale = scale / 9.f, oYScale = scale / 9.f;             // The overlay has a resolution of 9px per keymap unit
            float x = (xpos + offX) * scale, y = (ypos + offY) * scale;
            int w = overlay->width() * oXScale, h = overlay->height() * oYScale;
            // We need to transform the image with QImage::scaled() because painter.drawImage() will butcher it, even with smoothing enabled
            // However, the width/height need to be rounded to integers
            int iW = round(w), iH = round(h);
            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));
        }
    } else {
        // Otherwise, draw a solid background
        painter.setBrush(QBrush(bgColor));
        painter.drawRect(0, 0, width(), height());
    }

    // Draw mouse highlight (if any)
    if(mouseDownMode != NONE && (mouseDownX != mouseCurrentX || mouseDownY != mouseCurrentY)){
        int x1 = (mouseDownX > mouseCurrentX) ? mouseCurrentX : mouseDownX;
        int x2 = (mouseDownX > mouseCurrentX) ? mouseDownX : mouseCurrentX;
        int y1 = (mouseDownY > mouseCurrentY) ? mouseCurrentY : mouseDownY;
        int y2 = (mouseDownY > mouseCurrentY) ? mouseDownY : mouseCurrentY;
        painter.setPen(QPen(highlightColor, 0.5));
        QColor bColor = highlightColor;
        bColor.setAlpha(128);
        painter.setBrush(QBrush(bColor));
        painter.drawRect(x1, y1, x2 - x1, y2 - y1);
    }

    // Draw key backgrounds on a separate pixmap so that a drop shadow can be applied to them.
    QPixmap keyBG(wWidth * ratio, wHeight * ratio);
    keyBG.fill(QColor(0, 0, 0, 0));
    QPainter bgPainter(&keyBG);
    bgPainter.setRenderHints(QPainter::Antialiasing | QPainter::TextAntialiasing);
    bgPainter.setPen(Qt::NoPen);
    QHashIterator<QString, Key> k(keyMap);
    uint i = -1;
    while(k.hasNext()){
        k.next();
        i++;
        const Key& key = k.value();
        float x = key.x + offX - key.width / 2.f + 1.f;
        float y = key.y + offY - key.height / 2.f + 1.f;
        float w = key.width - 2.f;
        float h = key.height - 2.f;
        // In RGB mode, ignore keys without LEDs
        if((_rgbMode && !key.hasLed)
                || (!_rgbMode && !key.hasScan))
            continue;
        // Set color based on key highlight
        bgPainter.setOpacity(1.);
        if(highlight.testBit(i)){
            if(animation.testBit(i))
                bgPainter.setBrush(QBrush(highlightAnimColor));
            else
                bgPainter.setBrush(QBrush(highlightColor));
        } else if(animation.testBit(i)){
            bgPainter.setBrush(QBrush(animColor));
        } else {
            if(!strcmp(key.name, "sniper"))
                // Sniper key uses a reddish base color instead of the usual grey
                bgPainter.setBrush(QBrush(sniperColor));
            else if(model == KeyMap::SCIMITAR && !strncmp(key.name, "thumb", 5) && strcmp(key.name, "thumb"))
                // Thumbgrid keys use a black color
                bgPainter.setBrush(QBrush(thumbColor));
            else if(!strcmp(key.name, "lsidel") || !strcmp(key.name, "rsidel") || !strcmp(key.name, "logo"))
                // Strafe side lights have different background
                bgPainter.setBrush(QBrush(transparentColor));
            else {
                bgPainter.setBrush(QBrush(keyColor));
                if(KeyMap::isMouse(model))
                    bgPainter.setOpacity(0.7);
            }
        }
        if(model != KeyMap::STRAFE && (!strcmp(key.name, "mr") || !strcmp(key.name, "m1") || !strcmp(key.name, "m2") || !strcmp(key.name, "m3")
                || !strcmp(key.name, "light") || !strcmp(key.name, "lock") || (model ==
```

```
KeyMap::K65 && !strcmp(key.name, "mute")))){
            // Switch keys are circular except for Strafe. All Strafe keys are square
            x += w / 8.f;
            y += h / 8.f;
            w *= 0.75f;
            h *= 0.75f;
            bgPainter.drawEllipse(QRectF(x * scale, y * scale, w * scale, h * scale));
        } else {
            if(!strcmp(key.name, "enter")){
                if(key.height == 24){
                    // ISO enter key isn't rectangular
                    y = key.y + 1.f;
                    h = 10.f;
                    bgPainter.drawRect(QRectF((x + w - 13.f) * scale, y * scale, 13.f * scale, 22.f * scale
));
                } else {
                    // US enter key isn't perfectly centered, needs an extra pixel on the left to appear
 correctly
                    x -= 1.f;
                    w += 1.f;
                }
            } else if(!strcmp(key.name, "rshift") || !strcmp(key.name, "stop")){
                // A few other keys also need extra pixels
                x -= 1.f;
                w += 1.f;
            } else if(!strcmp(key.name, "caps") || !strcmp(key.name, "lshift") || !strcmp(key.
name, "next")){
                w += 1.f;
            }
            bgPainter.drawRect(QRectF(x * scale, y * scale, w * scale, h * scale));
        }
    }

    // Render the key decorations (RGB -> light circles, binding -> key names) on yet another layer
    QPixmap decoration(wWidth * ratio, wHeight * ratio);
    decoration.fill(QColor(0, 0, 0, 0));
    QPainter decPainter(&decoration);
    decPainter.setRenderHints(QPainter::Antialiasing | QPainter::TextAntialiasing);
    if(_rgbMode){
        // Draw key colors (RGB mode)
        QHashIterator<QString, Key> k(keyMap);
        uint i = -1;
        while(k.hasNext()){
            k.next();
            i++;
            const Key& key = k.value();
            if(!key.hasLed)
                continue;
            float x = key.x + offX - 1.8f;
            float y = key.y + offY - 1.8f;
            float w = 3.6f;
            float h = 3.6f;
            // Display a white circle around regular keys, red circle around indicators
            if(_indicators.contains(key.name))
                decPainter.setPen(QPen(QColor(255, 248, 136), 1.5));
            else
                decPainter.setPen(QPen(QColor(255, 255, 255), 1.5));
            QRgb color;
            const QRgb* inDisplay = _displayColorMap.
colorForName(key.name);
            if(inDisplay)
                // Color in display map? Grab it from there
                // (monochrome conversion not necessary as this would have been done by the animation)
                color = *inDisplay;
            else {
                // Otherwise, read from base map
                color = _colorMap.value(k.key());
                if(_monochrome)
                    color = monoRgb(qRed(color), qGreen(color), qBlue(color));
            }
            decPainter.setBrush(QBrush(color));
            if (model == KeyMap::STRAFE) { // STRAFE custom design and special keys
                float kx = key.x + offX - key.width / 2.f + 1.f;
                float ky = key.y + offY - key.height / 2.f + 1.f;
                float kw = key.width - 2.f;
                float kh = key.height - 2.f;
                decPainter.setPen(QPen(QColor(255, 255, 255), 1.2)); // less invasive outline to show the
 key color better
                if(!strcmp(key.name, "logo")) { // stylized logo
                    float lx = key.x + offX - key.width / 2.f + 2.f;
                    float ly = key.y + offY - key.height / 2.f + 2.f;
                    float lw = key.width - 4.f;
                    float lh = key.height - 4.f;
                    QPainterPath logo;
                    logo.moveTo(lx*scale,(ly+lh)*scale);
                    logo.quadTo((lx+2.f)*scale,(ly+lh/2.f)*scale,lx*scale,ly*scale);
                    logo.quadTo((lx+lw)*scale,ly*scale,(lx+lw)*scale,(ly+lh)*scale);
```

```
                    logo.quadTo((lx+lw/2.f)*scale,(ly+lh-4.f)*scale,lx*scale,(ly+lh)*scale);
                    decPainter.drawPath(logo);
                    //decPainter.setPen(QPen(Qt::green, 1.2)); //QColor(125,125,125)
                    //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
                } else if(!strcmp(key.name, "lsidel") || !strcmp(key.name, "rsidel")) { // Strafe side lights (toggle lights with no animation)
                    QRadialGradient gradient(QPointF(wWidth/2.f * ratio, wHeight/2.f * ratio), wWidth/2.f * ratio);//,QPointF(10, 5));
                    gradient.setColorAt(0, color);
                    gradient.setColorAt(0.9, color); // bring up intensity
                    gradient.setColorAt(1, bgColor);
                    decPainter.setBrush(QBrush(gradient));
                    decPainter.setPen(QPen(keyColor, 1.2)); //QColor(125,125,125)
                    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
                } else if(_indicators.contains(key.name)) { // FIX: This check fails whenever _indicators is empty because "show animated" is unchecked
                    decPainter.setPen(QPen(QColor(0,0,0,0), 1));    // no outline for the indicators, you can't change their color the standard way
                    decPainter.drawRect(QRectF((kx+2.f) * scale, (ky+2.f) * scale, (kw-4.f) * scale, (kh-4.f) * scale)); // square indicators
                } else //everything else is a circle, just a tad bigger to show the key color better
                    decPainter.drawEllipse(QRectF((x-1.f) * scale, (y-1.f) * scale, (w+2.f) * scale, (h+2.f) * scale));
            } else
                decPainter.drawEllipse(QRectF(x * scale, y * scale, w * scale, h * scale));
        }
    } else {
        // Draw key names
        decPainter.setBrush(Qt::NoBrush);
        QFont font = painter.font();
        font.setBold(true);
        font.setPixelSize(5.25f * scale);
        QFont font0 = font;
        QHashIterator<QString, Key> k(keyMap);
        uint i = -1;
        while(k.hasNext()){
            k.next();
            i++;
            const Key& key = k.value();
            if(!key.hasScan)
                continue;
            float x = key.x + offX - key.width / 2.f + 1.f;
            float y = key.y + offY - key.height / 2.f;
            float w = key.width - 2.f;
            float h = key.height;
            // Print the key's friendly name (with some exceptions)
            QString keyName = KbBind::globalRemap(key.name);
            QString name = key.friendlyName(false);
            name = name.split(" ").last();
            struct {
                const char* keyName, *displayName;
            } names[] = {
                {"light", ""}, {"lock", ""}, {"mute", ""}, {"volup", keyMap.model() == KeyMap::K65 ? ")" : ""}, {"voldn", keyMap.model() == KeyMap::K65 ? ")" : ""},
                {"prtscn",  "PrtScn\nSysRq"}, {"scroll", "Scroll\nLock"}, {"pause", "Pause\nBreak"}, {"stop", ""}, {"prev", "|"}, {"play", "||"}, {"next", "|"},
                {"pgup", "Page\nUp"}, {"pgdn", "Page\nDown"}, {"numlock", "Num\nLock"},
                {"caps", "Caps"}, {"lshift", "Shift"}, {"rshift", "Shift"},
#ifdef Q_OS_MACX
                {"lctrl", ""}, {"rctrl", ""}, {"lwin", ""}, {"rwin", ""}, {"lalt", ""}, {"ralt", ""},
#else
                {"lctrl", "Ctrl"}, {"rctrl", "Ctrl"}, {"lwin", ""}, {"rwin", ""}, {"lalt", "Alt"}, {"ralt", "Alt"},
#endif
                {"rmenu", ""}, {"up", ""}, {"left", ""}, {"down", ""}, {"right", ""}, {"fn","Fn"},
                {"mouse1", ""}, {"mouse2", ""}, {"mouse3", ""}, {"dpiup", ""}, {"dpidn", ""}, {"wheelup", ""}, {"wheeldn", ""}, {"dpi", ""}, {"mouse5", ""}, {"mouse4", ""}, {"sniper", ""}
            };
            for(uint k = 0; k < sizeof(names) / sizeof(names[0]); k++){
                if(keyName == names[k].keyName){
                    name = names[k].displayName;
                    break;
                }
            }
            if(keyName == "thumb1" && model == KeyMap::SABRE)
                name = "";
            if(keyName == "mr" || keyName == "m1" || keyName == "m2" || keyName == "m3" || keyName == "up" || keyName == "down" || keyName == "left" || keyName == "right")
                // Use a smaller size for MR, M1 - M3, and arrow keys
                font.setPixelSize(font.pixelSize() * 0.75);
            else if(keyName == "end")
                // Use a smaller size for "End" to match everything else in that area
                font.setPixelSize(font.pixelSize() * 0.65);
            else if(keyName == "light"
```

```
#ifndef Q_OS_MACX
                    || keyName == "lwin" || keyName == "rwin"
#endif
                    )
                // Use a larger font size for Super (Linux only) and Brightness to compensate for the unicode symbols looking smaller
                font.setPixelSize(font.pixelSize() * 1.3);
            if((layout == KeyMap::EU || layout == KeyMap::EU_DVORAK) && (keyName == "hash" || keyName == "bslash_iso"))
                // Don't differentiate backslashes on the EU layout
                name = "\\";
            // Determine the appropriate size to draw the text at
            decPainter.setFont(font);
            QRectF rect(x * scale, y * scale - 1, w * scale, h * scale);
            int flags = Qt::AlignHCenter | Qt::AlignVCenter | Qt::TextWordWrap;
            QRectF bounds = decPainter.boundingRect(rect, flags, name);
            while((bounds.height() >= rect.height() - 8. || bounds.width() >= rect.width() - 2.) && font.pixelSize() >= 5){
                // Scale font size down until it fits inside the key
                font.setPixelSize(font.pixelSize() - 2);
                decPainter.setFont(font);
                bounds = decPainter.boundingRect(rect, flags, name);
            }
            // Pick color based on key function
            QString bind = _bindMap.value(key.name);
            QString def = KbBind::defaultAction(key.name);
            if(bind.isEmpty())
                // Unbound - red
                decPainter.setPen(QColor(255, 136, 136));
            else if(KeyAction(bind).isProgram())
                // Custom program - orange
                decPainter.setPen(QColor(255, 224, 192));
            else if(KeyAction(bind).isSpecial() && (bind == def || !KeyAction(def).isSpecial()))
                // Special function - blue (only if not mapped to a different function - if a special function is remapped, color it yellow)
                decPainter.setPen(QColor(128, 224, 255));
            else if(KeyAction(bind).isMedia() && (bind == def || !KeyAction(def).isMedia()))
                // Media key - green
                decPainter.setPen(QColor(160, 255, 168));
            else if(bind == def)
                // Standard key - white
                decPainter.setPen(QColor(255, 255, 255));
            else
                // Remapped key - yellow
                decPainter.setPen(QColor(255, 248, 128));
            decPainter.drawText(rect, flags, name);
            font = font0;
        }
    }
    // Create drop shadow effects
    QGraphicsDropShadowEffect* bgEffect = new QGraphicsDropShadowEffect;  // Have to use "new", creating these on the stack causes a crash...
    bgEffect->setBlurRadius(2.);
    bgEffect->setColor(QColor(0, 0, 0, 32));
    bgEffect->setOffset(0, 1);
    QGraphicsDropShadowEffect* decEffect = new QGraphicsDropShadowEffect;
    decEffect->setBlurRadius(4.);
    decEffect->setColor(QColor(0, 0, 0, 104));
    decEffect->setOffset(0, 1);
    // Apply them to the pixmaps
    QGraphicsPixmapItem* bgItem = new QGraphicsPixmapItem(keyBG);
    bgItem->setGraphicsEffect(bgEffect);
    QGraphicsPixmapItem* decItem = new QGraphicsPixmapItem(decoration);
    decItem->setGraphicsEffect(decEffect);
    // Render everything
    QGraphicsScene* scene = new QGraphicsScene;
    scene->addItem(bgItem);
    scene->addItem(decItem);
    // It has to be rendered onto yet another pixmap or else DPI scaling will look terrible...
    QPixmap final(wWidth * ratio, wHeight * ratio);
    final.fill(QColor(0, 0, 0, 0));
    QPainter finalPainter(&final);
    scene->render(&finalPainter, QRectF(0, 0, wWidth * ratio, wHeight * ratio), QRectF(0, 0, wWidth * ratio, wHeight * ratio));
    delete scene;   // <- Automatically cleans up the rest of the objects
#if QT_VERSION >= QT_VERSION_CHECK(5, 3, 0)
    final.setDevicePixelRatio(ratio);
#endif
    painter.drawPixmap(QPointF(0., 0.), final);
}
```

Here is the call graph for this function:

8.33.5.17 bool KeyWidget::rgbMode ( ) [inline]

Definition at line 17 of file keywidget.h.

References _rgbMode.

```
17 { return _rgbMode; }
```

8.33.5.18 void KeyWidget::rgbMode (bool *newRgbMode*) [inline]

Definition at line 18 of file keywidget.h.

References _rgbMode.

```
18 { _rgbMode = newRgbMode; update(); }
```

8.33.5.19 void KeyWidget::selectAll ( )

Definition at line 589 of file keywidget.cpp.

References _rgbMode, Key::hasLed, Key::hasScan, keyMap, mouseDownMode, Key::name, newSelection, NONE, KeyMap::positions(), selection, and selectionChanged().

```
589                         {
590     selection.fill(false);
591     // Fill the selection with all keys that have an LED/scancode (depending on widget mode)
592     int i = 0;
593     QStringList selectedNames;
594     foreach(const Key& key, keyMap.positions()){
595         // Sidelights can't be selected
```

```
        if(strcmp(key.name, "lsidel") && strcmp(key.name, "rsidel")
           && ((_rgbMode && key.hasLed) || !(_rgbMode && key.
    hasScan))){
            selection.setBit(i);
            selectedNames << key.name;
        }
        i++;
    }
    // Clear mousedown state
    newSelection.fill(false);
    mouseDownMode = NONE;
    update();
    emit selectionChanged(selectedNames);
}
```

Here is the call graph for this function:

8.33.5.20 void KeyWidget::selectionChanged (QStringList *selected*) [signal]

Referenced by clearSelection(), mouseReleaseEvent(), selectAll(), and setSelection().

Here is the caller graph for this function:

8.33.5.21 void KeyWidget::setAnimation (const QStringList & *keys*)

Definition at line 618 of file keywidget.cpp.

References animation, keyMap, and KeyMap::keys().

```
618                                                 {
619     animation.fill(false);
620     QStringList allNames = keyMap.keys();
621     foreach(const QString& key, keys){
622         // Sidelights can't be selected
623         if(!strcmp(key.toLatin1(), "lsidel") || !strcmp(key.toLatin1(), "rsidel"))
```

```
            continue;
        int index = allNames.indexOf(key);
        if(index >= 0)
            animation.setBit(index);
    }
    update();
}
```

Here is the call graph for this function:

8.33.5.22 void KeyWidget::setAnimationToSelection ( )

Definition at line 632 of file keywidget.cpp.

References animation, and selection.

```
632                                     {
633     animation = selection;
634     update();
635 }
```

8.33.5.23 void KeyWidget::setSelection (const QStringList & *keys*)

Definition at line 575 of file keywidget.cpp.

References keyMap, KeyMap::keys(), mouseDownMode, newSelection, NONE, selection, and selectionChanged().

```
575                                                 {
576     selection.fill(false);
577     QStringList allNames = keyMap.keys();
578     foreach(const QString& key, keys){
579         int index = allNames.indexOf(key);
580         if(index >= 0)
581             selection.setBit(index);
582     }
583     newSelection.fill(false);
584     mouseDownMode = NONE;
585     update();
586     emit selectionChanged(keys);
587 }
```

Here is the call graph for this function:

8.33.5.24 void KeyWidget::sidelightToggled ( ) [signal]

Referenced by mousePressEvent().

Here is the caller graph for this function:

8.33.6 Field Documentation

8.33.6.1 BindMap KeyWidget::_bindMap [private]

Definition at line 58 of file keywidget.h.

Referenced by bindMap(), and paintEvent().

8.33.6.2 QColorMap KeyWidget::_colorMap [private]

Definition at line 56 of file keywidget.h.

Referenced by colorMap(), and paintEvent().

8.33.6.3 ColorMap KeyWidget::_displayColorMap [private]

Definition at line 57 of file keywidget.h.

Referenced by displayColorMap(), and paintEvent().

8.33.6.4 QSet<QString> KeyWidget::_indicators [private]

Definition at line 59 of file keywidget.h.

Referenced by displayColorMap(), mouseMoveEvent(), and paintEvent().

8.33.6.5 bool KeyWidget::_monochrome [private]

Definition at line 73 of file keywidget.h.

Referenced by monochrome(), and paintEvent().

8.33.6.6 bool KeyWidget::_rgbMode [private]

Definition at line 73 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), paintEvent(), rgbMode(), and selectAll().

8.33.6.7 QBitArray KeyWidget::animation [private]

Definition at line 63 of file keywidget.h.

Referenced by clearAnimation(), map(), paintEvent(), setAnimation(), and setAnimationToSelection().

8.33.6.8 KeyMap KeyWidget::keyMap [private]

Definition at line 55 of file keywidget.h.

Referenced by drawInfo(), map(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), selectAll(), setAnimation(), and setSelection().

8.33.6.9 int KeyWidget::mouseCurrentX [private]

Definition at line 65 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

8.33.6.10 int KeyWidget::mouseCurrentY [private]

Definition at line 65 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

8.33.6.11 enum { ... } KeyWidget::mouseDownMode [private]

Referenced by clearSelection(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), selectAll(), and setSelection().

8.33.6.12 int KeyWidget::mouseDownX [private]

Definition at line 64 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

8.33.6.13 int KeyWidget::mouseDownY [private]

Definition at line 64 of file keywidget.h.

Referenced by mouseMoveEvent(), mousePressEvent(), and paintEvent().

8.33.6.14 QBitArray KeyWidget::newSelection [private]

Definition at line 62 of file keywidget.h.

Referenced by clearSelection(), map(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), paintEvent(), selectAll(), and setSelection().

8.33.6.15 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

8.34 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 ()
- 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)
- void mode2Raw (HwMode mode, bool &sw_enable, i_hw &hw_enable)

Private Attributes

- Ui::KPerfWidget * ui
- KbPerf * perf
- KbProfile * profile
- IndicatorUi indicators [I_COUNT]
- QList< QWidget * > k95Widgets
- QSignalMapper updateMapper

8.34.1 Detailed Description

Definition at line 16 of file kperfwidget.h.

8.34.2 Data Structure Documentation

8.34.2.1 struct KPerfWidget::IndicatorUi

Definition at line 48 of file kperfwidget.h.

Collaboration diagram for KPerfWidget::IndicatorUi:

Data Fields

ColorButton *	color1	
ColorButton *	color2	
ColorButton *	color3	
QCheckBox *	enable	
QComboBox *	hwEnable	

8.34.3 Member Typedef Documentation

8.34.3.1 typedef KbPerf::i_hw KPerfWidget::i_hw

Definition at line 29 of file kperfwidget.h.

8.34.3.2 typedef KbPerf::indicator KPerfWidget::indicator

Definition at line 28 of file kperfwidget.h.

8.34.4 Member Enumeration Documentation

8.34.4.1 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     };
```

8.34.5 Constructor & Destructor Documentation

8.34.5.1 KPerfWidget::KPerfWidget (QWidget * *parent* = 0) [explicit]

Definition at line 6 of file kperfwidget.cpp.

References ColorButton::allowAlpha(), ColorButton::bigIcons(), KPerfWidget::IndicatorUi::color1, KPerfWidget::IndicatorUi::color2, KPerfWidget::IndicatorUi::color3, KPerfWidget::IndicatorUi::enable, KPerfWidget::IndicatorUi::hwEnable, I_COUNT, indicators, k95Widgets, ColorButton::setLabel(), ui, uiUpdated(), and updateMapper.

```
6                                          :
7     QWidget(parent),
8     ui(new Ui::KPerfWidget)
9 {
10     ui->setupUi(this);
11     // Set up indicators
12     indicators[0].enable = 0; indicators[0].hwEnable =
      ui->numBox; indicators[0].color1 = ui->numColorOn;
      indicators[0].color2 = ui->numColorOff; indicators[0].
      color3 = 0;
13     indicators[1].enable = 0; indicators[1].hwEnable =
      ui->capsBox; indicators[1].color1 = ui->capsColorOn;
      indicators[1].color2 = ui->capsColorOff; indicators[1].
      color3 = 0;
14     indicators[2].enable = 0; indicators[2].hwEnable =
      ui->scrollBox; indicators[2].color1 = ui->scrollColorOn;
      indicators[2].color2 = ui->scrollColorOff; indicators[2].
      color3 = 0;
15     indicators[3].enable = ui->modeBox; indicators[3].
      hwEnable = 0; indicators[3].color1 = ui->modeColorOn;
      indicators[3].color2 = ui->modeColorOff; indicators[3].
      color3 = 0;
16     indicators[4].enable = ui->macroBox; indicators[4].
      hwEnable = 0; indicators[4].color1 = ui->macroColorOn;
      indicators[4].color2 = ui->macroColorOff; indicators[4].
      color3 = 0;
```

```
 indicators[5].enable = ui->lightBox; indicators[5].hwEnable = 0; indicators[5].color1 = ui->lightColor1; indicators[5].color2 = ui->lightColor2; indicators[5].color3 = ui->lightColor3;
 indicators[6].enable = ui->lockBox; indicators[6].hwEnable = 0; indicators[6].color1 = ui->lockColorOn; indicators[6].color2 = ui->lockColorOff; indicators[6].color3 = 0;
 indicators[7].enable = ui->muteBox; indicators[7].hwEnable = 0; indicators[7].color1 = ui->muteColorOn; indicators[7].color2 = ui->muteColorOff; indicators[7].color3 = ui->muteColorNA;
 for(int i = 0; i < I_COUNT; i++){
     indicators[i].color1->setLabel(false);
     indicators[i].color1->bigIcons(true);
     indicators[i].color1->allowAlpha(true);
     indicators[i].color2->setLabel(false);
     indicators[i].color2->bigIcons(true);
     indicators[i].color2->allowAlpha(true);
     if(indicators[i].color3){
         indicators[i].color3->setLabel(false);
         indicators[i].color3->bigIcons(true);
         indicators[i].color3->allowAlpha(true);
     }
     // Map signals
     if(indicators[i].enable){
         connect(indicators[i].enable, SIGNAL(clicked(bool)), &updateMapper, SLOT(map()));
         updateMapper.setMapping(indicators[i].enable, i);
     }
     if(indicators[i].hwEnable){
         connect(indicators[i].hwEnable, SIGNAL(activated(int)), &updateMapper, SLOT(map()));
         updateMapper.setMapping(indicators[i].hwEnable, i);
     }
     connect(indicators[i].color1, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
     updateMapper.setMapping(indicators[i].color1, i);
     connect(indicators[i].color2, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
     updateMapper.setMapping(indicators[i].color2, i);
     if(indicators[i].color3){
         connect(indicators[i].color3, SIGNAL(colorChanged(QColor)), &updateMapper, SLOT(map()));
         updateMapper.setMapping(indicators[i].color3, i);
     }
 }
 connect(&updateMapper, SIGNAL(mapped(int)), this, SLOT(uiUpdated(int)));
 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;
}
```

Here is the call graph for this function:

8.34.5.2 KPerfWidget::~KPerfWidget ( )

Definition at line 54 of file kperfwidget.cpp.

References ui.

```
54                          {
55     delete ui;
56 }
```

8.34.6 Member Function Documentation

8.34.6.1 void KPerfWidget::mode2Raw (HwMode *mode*, bool & *sw_enable*, i_hw & *hw_enable*) [private]

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 uiUpdated().

```
72                                                                  {
73     switch(mode){
74     case NORMAL:
75         sw_enable = false;
76         hw_enable = KbPerf::NORMAL;
77         break;
78     case ALWAYS_ON:
79         sw_enable = false;
80         hw_enable = KbPerf::ON;
81         break;
82     case ALWAYS_OFF:
83         sw_enable = false;
84         hw_enable = KbPerf::OFF;
85         break;
86     case RGB:
87         sw_enable = true;
88         hw_enable = KbPerf::OFF;
89         break;
90     case BOTH:
91         sw_enable = true;
92         hw_enable = KbPerf::NORMAL;
93         break;
94     }
95 }
```

Here is the caller graph for this function:

8.34.6.2 void KPerfWidget::on_copyButton_clicked ( ) [private],[slot]

Definition at line 193 of file kperfwidget.cpp.

References KbProfile::currentMode(), KbProfile::modes(), perf, KbMode::perf(), profile, and ModeSelectDialog::selection().

```
193                                     {
194     ModeSelectDialog dialog(this, profile->currentMode(),
      profile->modes(), "Copy performance settings to:");
195     if(dialog.exec() != QDialog::Accepted)
196         return;
```

```
    QList<KbMode*> selectedModes = dialog.selection();
    foreach(KbMode* mode, selectedModes){
        *mode->perf() = *perf;
    }
}
```

Here is the call graph for this function:

8.34.6.3 void KPerfWidget::on_intensityBox_valueChanged (int *arg1*) [private],[slot]

Definition at line 187 of file kperfwidget.cpp.

References KbPerf::iOpacity(), and perf.

```
187                                                     {
188     if(!perf)
189         return;
190     perf->iOpacity(arg1 / 100.f);
191 }
```

Here is the call graph for this function:

8.34.6.4 KPerfWidget::HwMode KPerfWidget::raw2Mode (bool *sw_enable*, i_hw *hw_enable*) [private]

Definition at line 58 of file kperfwidget.cpp.

References ALWAYS_OFF, ALWAYS_ON, BOTH, NORMAL, KbPerf::NORMAL, KbPerf::OFF, KbPerf::ON, and RGB.

Referenced by setPerf().

```
                                                                    {
    if(sw_enable){
        if(hw_enable == KbPerf::NORMAL)
            return BOTH;
        return RGB;
    } else {
        if(hw_enable == KbPerf::ON)
            return ALWAYS_ON;
        else if(hw_enable == KbPerf::OFF)
            return ALWAYS_OFF;
        return NORMAL;
    }
}
```

Here is the caller graph for this function:

8.34.6.5 void KPerfWidget::setPerf (KbPerf * *newPerf*, KbProfile * *newProfile*)

Definition at line 97 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, KbPerf::iOpacity(), KeyMap::K95, k95Widgets, KbProfile::keyMap(), KeyMap::model(), perf, profile, raw2Mode(), RGB, KeyMap::STRAFE, and ui.

```
97                                                                  {
98      perf = newPerf;
99      profile = newProfile;
100      // Set intensity
101      ui->intensityBox->setValue(round(perf->iOpacity() * 100.f));
102      // Set hardware indicator values
103      for(int i = 0; i < HW_I_COUNT; i++){
104          QColor c1, c2, c3;
105          bool software;
106          i_hw hardware;
107          perf->getIndicator((indicator)i, c1, c2, c3, software, hardware);
108          HwMode mode = raw2Mode(software, hardware);
109          indicators[i].hwEnable->setCurrentIndex((int)mode);
110          indicators[i].color1->color(c1);
111          indicators[i].color2->color(c2);
112          if(mode == RGB || mode == BOTH){
113              indicators[i].color1->setEnabled(true);
114              indicators[i].color2->setEnabled(true);
115          } else {
116              indicators[i].color1->setEnabled(false);
117              indicators[i].color2->setEnabled(false);
118          }
119      }
120      // Set software indicators
121      for(int i = HW_I_COUNT; i < I_COUNT; i++){
122          QColor c1, c2, c3;
123          bool software;
124          i_hw hardware;
125          perf->getIndicator((indicator)i, c1, c2, c3, software, hardware);
126          indicators[i].enable->setChecked(software);
127          indicators[i].color1->color(c1);
128          indicators[i].color2->color(c2);
129          if(indicators[i].color3) indicators[i].color3->
     color(c3);
130          if(software){
131              indicators[i].color1->setEnabled(true);
132              indicators[i].color2->setEnabled(true);
133              if(indicators[i].color3) indicators[i].color3->setEnabled(true);
```

```
        } else {
            indicators[i].color1->setEnabled(false);
            indicators[i].color2->setEnabled(false);
            if(indicators[i].color3) indicators[i].color3->setEnabled(false);
        }
    }
    // Hide K95 indicators on non-K95s
    if(profile->keyMap().model() == KeyMap::K95){
        foreach(QWidget* w, k95Widgets)
            w->show();
    } else {
        foreach(QWidget* w, k95Widgets)
            w->hide();
        if(profile->keyMap().model() == KeyMap::STRAFE) { // hide indicators unsupported by STRAFE
            ui->muteBox->hide();
            ui->muteColorOn->hide();
            ui->muteColorOff->hide();
            ui->muteColorNA->hide();
            ui->label_12->hide();
            ui->label_14->hide();
            ui->label_17->hide();
        }
    }
}
```

Here is the call graph for this function:

8.34.6.6 void KPerfWidget::uiUpdated (int *index*) [private],[slot]

Definition at line 159 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().

```
159                                     {
160     if(!perf)
```

```
        return;
    // Read HW/SW enable state
    bool software;
    i_hw hardware;
    if(indicators[index].enable){
        software = indicators[index].enable->isChecked();
        hardware = KbPerf::NONE;
    } else {
        mode2Raw((HwMode)indicators[index].hwEnable->currentIndex(), software, hardware);
    }
    // Enable/disable color buttons based on whether or not the indicator is turned on
    if(software){
        indicators[index].color1->setEnabled(true);
        indicators[index].color2->setEnabled(true);
        if(indicators[index].color3) indicators[index].color3->setEnabled(true);
    } else {
        indicators[index].color1->setEnabled(false);
        indicators[index].color2->setEnabled(false);
        if(indicators[index].color3) indicators[index].color3->setEnabled(false);
    }
    QColor c1 = indicators[index].color1->color();
    QColor c2 = indicators[index].color2->color();
    QColor c3 = indicators[index].color3 ? indicators[index].color3->color() : QColor();
    perf->setIndicator((indicator)index, c1, c2, c3, software, hardware);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.34.7 Field Documentation

8.34.7.1 const int KPerfWidget::HW_I_COUNT = KbPerf::HW_I_COUNT [static]

Definition at line 27 of file kperfwidget.h.

Referenced by setPerf().

8.34.7.2 const int KPerfWidget::I_COUNT = KbPerf::I_COUNT [static]

Definition at line 26 of file kperfwidget.h.

Referenced by KPerfWidget(), and setPerf().

8.34.7.3 IndicatorUi KPerfWidget::indicators[I_COUNT] [private]

Definition at line 53 of file kperfwidget.h.

Referenced by KPerfWidget(), setPerf(), and uiUpdated().

8.34.7.4 QList<QWidget*> KPerfWidget::k95Widgets [private]

Definition at line 54 of file kperfwidget.h.

Referenced by KPerfWidget(), and setPerf().

8.34.7.5 KbPerf* KPerfWidget::perf [private]

Definition at line 34 of file kperfwidget.h.

Referenced by on_copyButton_clicked(), on_intensityBox_valueChanged(), setPerf(), and uiUpdated().

8.34.7.6 KbProfile* KPerfWidget::profile [private]

Definition at line 35 of file kperfwidget.h.

Referenced by on_copyButton_clicked(), and setPerf().

8.34.7.7 Ui::KPerfWidget* KPerfWidget::ui [private]

Definition at line 32 of file kperfwidget.h.

Referenced by KPerfWidget(), setPerf(), and ~KPerfWidget().

8.34.7.8 QSignalMapper KPerfWidget::updateMapper [private]

Definition at line 56 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

8.35 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

8.35.1 Detailed Description

Definition at line 13 of file layoutdialog.h.

8.35.2 Constructor & Destructor Documentation

8.35.2.1 LayoutDialog::LayoutDialog (QWidget * *parent* = 0) [explicit]

Definition at line 5 of file layoutdialog.cpp.

References Kb::layout(), KeyMap::layoutNames(), 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:

8.35.2.2 LayoutDialog::~LayoutDialog ( )

Definition at line 14 of file layoutdialog.cpp.

References ui.

```
14                                {
15     delete ui;
16 }
```

8.35.3 Member Function Documentation

8.35.3.1 KeyMap::Layout LayoutDialog::selected ( ) const

Definition at line 18 of file layoutdialog.cpp.

References ui.

Referenced by SettingsWidget::showLayoutDialog().

```
                                        {
    return (KeyMap::Layout)ui->comboBox->currentIndex();
}
```

Here is the caller graph for this function:

8.35.4 Field Documentation

8.35.4.1 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

8.36 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

8.36.1 Detailed Description

Definition at line 87 of file macroreader.h.

8.36.2 Constructor & Destructor Documentation

8.36.2.1 MacroReader::MacroReader ( )

Definition at line 8 of file macroreader.cpp.

```
8                           {
9     qDebug() << "Calling MacroReader without params is not allowed.";
10 }
```

8.36.2.2 MacroReader::MacroReader (int *macroNumber*, QString *macroPath*, QPlainTextEdit * *macBox*, QPlainTextEdit * *macText*)

Parameters

macroNumber	
macroPath	
macBox	
macText	

Definition at line 12 of file macroreader.cpp.

References startWorkInAThread().

```
12
        {
13      startWorkInAThread(macroNumber, macroPath, macBox, macText);
14 }
```

Here is the call graph for this function:

8.36.2.3 MacroReader::~MacroReader ( )

Definition at line 15 of file macroreader.cpp.

```
15 {}
```

8.36.3 Member Function Documentation

8.36.3.1 void MacroReader::startWorkInAThread (int *macroNumber*, QString *macroPath*, QPlainTextEdit * *macBox*, QPlainTextEdit * *macText*)

Parameters

macroNumber	
macroPath	
macBox	
macText	

Definition at line 17 of file macroreader.cpp.

References macroReaderThread.

Referenced by MacroReader().

```
17                                                  {
18     macroReaderThread = new MacroReaderThread(macroNumber, macroPath,
       macBox, macText);
19     connect(macroReaderThread, &MacroReaderThread::finished,
       macroReaderThread, &QObject::deleteLater);
20     macroReaderThread->start();
21 }
```

Here is the caller graph for this function:

8.36.4 Field Documentation

8.36.4.1 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

8.37 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.

8.37.1 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.

8.37.2 Constructor & Destructor Documentation

8.37.2.1 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      }
```

8.37.3 Member Function Documentation

8.37.3.1 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 26 of file macroreader.cpp.

References macroBox, and macroText.

```
26                                                  {
31      macroText->setFocus();
32      QTextCursor c = macroText->textCursor();
33      c.setPosition(macroText->toPlainText().length());
34      macroText->setTextCursor(c);
35      macroBox->appendPlainText(line.left(line.size()-1));
36  }
```

8.37.3.2 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 46 of file macroreader.cpp.

References macroBox, macroNumber, macroPath, and macroText.

```
46                                  {
47      qDebug() << "MacroReader::run() started with" << macroNumber << "and" <<
       macroPath << "and" << macroBox << "and" << macroText;
48
49      QFile macroFile(macroPath);
50      // Wait a small amount of time for the node to open (100ms) (code reused from kb.cpp)
51      QThread::usleep(100000);
52      if(!macroFile.open(QIODevice::ReadOnly)){
53          // If it's still not open, try again before giving up (1s at a time, 10s total)
54          QThread::usleep(900000);
55          for(int i = 1; i < 10; i++){
56              if(macroFile.open(QIODevice::ReadOnly))
57                  break;
58              QThread::sleep(1);
59          }
60          if(!macroFile.isOpen()) {
61              qDebug() << QString("unable to open macroFile (%1)").arg(macroPath);
62              return;
63          }
64      }
65      // Read data from notification node macroPath
66      QByteArray line;
67      while(macroFile.isOpen() && (line = macroFile.readLine()).length() > 0){
68          QString text = QString::fromUtf8(line);
69          metaObject()->invokeMethod(this, "readMacro", Qt::QueuedConnection, Q_ARG(QString, text));
70      }
71      qDebug() << "MacroReader::run() ends.";
72      macroFile.close();
73      QThread::exit ();
74  }
```

8.37.4 Field Documentation

8.37.4.1 QPlainTextEdit* MacroReaderThread::macroBox [private]

Definition at line 40 of file macroreader.h.

Referenced by MacroReaderThread(), readMacro(), and run().

8.37.4.2 int MacroReaderThread::macroNumber [private]

Definition at line 31 of file macroreader.h.

Referenced by MacroReaderThread(), and run().

8.37.4.3 QString MacroReaderThread::macroPath [private]

See Also

macroNumber

Definition at line 36 of file macroreader.h.

Referenced by MacroReaderThread(), and run().

8.37.4.4 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

8.38 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

8.38.1 Detailed Description

Definition at line 27 of file mainwindow.h.

8.38.2 Constructor & Destructor Documentation

8.38.2.1 MainWindow::MainWindow (QWidget ∗ *parent* = 0) [explicit]

Definition at line 35 of file mainwindow.cpp.

References addDevice(), cleanup(), closeAction, configLabel, CkbSettings::get(), iconClicked(), KbManager::init(), KbManager::kbManager(), mainWindow, quitApp(), removeDevice(), restoreAction, KbManager::scanTimer(), settingsWidget, SettingsWidget::setVersion(), showWindow(), stateChange(), timerTick(), toggleTrayIcon(), trayIcon, trayIconMenu, ui, and updateVersion().

```
35                                      :
36     QMainWindow(parent),
37     ui(new Ui::MainWindow)
38 {
39     ui->setupUi(this);
40     mainWindow = this;
41
42     // Start device manager
43     KbManager::init(CKB_VERSION_STR);
44     connect(KbManager::kbManager(), SIGNAL(kbConnected(Kb*)), this, SLOT(
      addDevice(Kb*)));
45     connect(KbManager::kbManager(), SIGNAL(kbDisconnected(Kb*)), this, SLOT(
      removeDevice(Kb*)));
46     connect(KbManager::kbManager(), SIGNAL(versionUpdated()), this, SLOT(
      updateVersion()));
47     connect(KbManager::scanTimer(), SIGNAL(timeout()), this, SLOT(
      timerTick()));
48
49     // Set up tray icon
50     restoreAction = new QAction(tr("Restore"), this);
51     closeAction = new QAction(tr("Quit ckb"), this);
52 #ifdef USE_LIBAPPINDICATOR
53     QString desktop = std::getenv("XDG_CURRENT_DESKTOP");
54     unityDesktop = (desktop.toLower() == "unity");
55
56     if(unityDesktop){
57         trayIcon = 0;
```

```

        indicatorMenu = gtk_menu_new();
        indicatorMenuRestoreItem = gtk_menu_item_new_with_label("Restore");
        indicatorMenuQuitItem = gtk_menu_item_new_with_label("Quit ckb");

        gtk_menu_shell_append(GTK_MENU_SHELL(indicatorMenu), indicatorMenuRestoreItem);
        gtk_menu_shell_append(GTK_MENU_SHELL(indicatorMenu), indicatorMenuQuitItem);

        g_signal_connect(indicatorMenuQuitItem, "activate",
            G_CALLBACK(quitIndicator), this);
        g_signal_connect(indicatorMenuRestoreItem, "activate",
            G_CALLBACK(restoreIndicator), this);

        gtk_widget_show(indicatorMenuRestoreItem);
        gtk_widget_show(indicatorMenuQuitItem);

        indicator = app_indicator_new("ckb", "indicator-messages",
    APP_INDICATOR_CATEGORY_APPLICATION_STATUS);

        app_indicator_set_status(indicator, APP_INDICATOR_STATUS_ACTIVE);
        app_indicator_set_menu(indicator, GTK_MENU(indicatorMenu));
        app_indicator_set_icon(indicator, "ckb");
    } else
#endif // USE_LIBAPPINDICATOR
    {
        trayIconMenu = new QMenu(this);
        trayIconMenu->addAction(restoreAction);
        trayIconMenu->addAction(closeAction);
        trayIcon = new QSystemTrayIcon(QIcon(":/img/ckb-logo.png"), this);
        trayIcon->setContextMenu(trayIconMenu);
        connect(trayIcon, SIGNAL(activated(QSystemTrayIcon::ActivationReason)), this, SLOT(
    iconClicked(QSystemTrayIcon::ActivationReason)));
    }
    toggleTrayIcon(!CkbSettings::get("Program/SuppressTrayIcon").toBool());

#ifdef Q_OS_MACX
    // Make a custom "Close" menu action for OSX, as the default one brings up the "still running" popup
     unnecessarily
    QMenuBar* menuBar = new QMenuBar(this);
    setMenuBar(menuBar);
    this->menuBar()->addMenu("ckb")->addAction(closeAction);
#else
    // On linux, add a handler for Ctrl+Q
    new QShortcut(QKeySequence("Ctrl+Q"), this, SLOT(quitApp()));
#endif

    connect(ui->actionExit, SIGNAL(triggered()), this, SLOT(quitApp()));
    connect(closeAction, SIGNAL(triggered()), this, SLOT(quitApp()));
    connect(restoreAction, SIGNAL(triggered()), this, SLOT(
    showWindow()));
    connect(qApp, SIGNAL(applicationStateChanged(Qt::ApplicationState)), this, SLOT(
    stateChange(Qt::ApplicationState)));

    connect(qApp, SIGNAL(aboutToQuit()), this, SLOT(cleanup()));

    ui->tabWidget->addTab(settingsWidget = new SettingsWidget(this),
    configLabel);
    settingsWidget->setVersion("ckb-next " CKB_VERSION_STR);
}
```

Here is the call graph for this function:

8.38.2.2 MainWindow::~MainWindow ( )

Definition at line 303 of file mainwindow.cpp.

References cleanup(), and ui.

```
303                         {
304     cleanup();
305     delete ui;
306 }
```

Here is the call graph for this function:

8.38.3 Member Function Documentation

8.38.3.1 void MainWindow::addDevice (Kb * *device*) [private],[slot]

Definition at line 121 of file mainwindow.cpp.

References KbWidget::device, kbWidgets, KbWidget::name(), ui, and updateVersion().

Referenced by MainWindow().

```
121                                     {
122     // Connected already?
123     foreach(KbWidget* w, kbWidgets){
124         if(w->device == device)
125             return;
126     }
127     // Add the keyboard
128     KbWidget* widget = new KbWidget(this, device);
129     kbWidgets.append(widget);
130     // Add to tabber; switch to this device if on the settings screen
131     int count = ui->tabWidget->count();
132     ui->tabWidget->insertTab(count - 1, widget, widget->name());
133     if(ui->tabWidget->currentIndex() == count)
134         ui->tabWidget->setCurrentIndex(count - 1);
135     // Update connected device count
136     updateVersion();
137 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.2 void MainWindow::checkFwUpdates ( ) [private],[slot]

Definition at line 172 of file mainwindow.cpp.

References KbWidget::device, Kb::features, Kb::firmware, KbWidget::hasShownNewFW, kbWidgets, mainWindow, KbWidget::updateFwButton(), and KbFirmware::versionForBoard().

Referenced by timerTick().

```
172                                 {
173     if(!mainWindow->isVisible())
174         return;
175     foreach(KbWidget* w, kbWidgets){
176         // Display firmware upgrade notification if a new version is available
177         float version = KbFirmware::versionForBoard(w->
    device->features);
178         if(version > w->device->firmware.toFloat()){
179             if(w->hasShownNewFW)
180                 continue;
181             w->hasShownNewFW = true;
182             w->updateFwButton();
183             // Don't run this method here because it will lock up the timer and prevent devices from
     working properly
184             // Use a queued invocation instead
185             metaObject()->invokeMethod(this, "showFwUpdateNotification", Qt::QueuedConnection, Q_ARG(
    QWidget*, w), Q_ARG(float, version));
186             // Don't display more than one of these at once
187             return;
188         }
189     }
190 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.3 void MainWindow::cleanup ( ) [private],[slot]

Definition at line 295 of file mainwindow.cpp.

References CkbSettings::cleanUp(), kbWidgets, and KbManager::stop().

Referenced by MainWindow(), and ~MainWindow().

```
295                         {
296     foreach(KbWidget* w, kbWidgets)
297         delete w;
298     kbWidgets.clear();
299     KbManager::stop();
300     CkbSettings::cleanUp();
301 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.4 void MainWindow::closeEvent (QCloseEvent * *event*) [private]

Definition at line 209 of file mainwindow.cpp.

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

```
209                                         {
210     // 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
211     if(!event->spontaneous() || isHidden()){
212         event->accept();
213         return;
214     }
215     if(!CkbSettings::get("Popups/BGWarning").toBool()){
216         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.");
217         CkbSettings::set("Popups/BGWarning", true);
218     }
219     hide();
220     event->ignore();
221 }
```

Here is the call graph for this function:

8.38.3.5 void MainWindow::iconClicked (QSystemTrayIcon::ActivationReason *reason*) [private],[slot]

Definition at line 257 of file mainwindow.cpp.

References showWindow().

Referenced by MainWindow().

```
257                                                                 {
258     // On Linux, hide/show the app when the tray icon is clicked
259     // On OSX this just shows the menu
260 #ifndef Q_OS_MACX
261     if(reason == QSystemTrayIcon::DoubleClick || reason == QSystemTrayIcon::Trigger){
262         if(isVisible())
263             hide();
264         else
265             showWindow();
266     }
267 #endif
268 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.6 void MainWindow::quitApp ( ) [slot]

Definition at line 291 of file mainwindow.cpp.

Referenced by MainWindow().

```
                        {
    qApp->quit();
}
```

Here is the caller graph for this function:

8.38.3.7 void MainWindow::removeDevice (Kb * *device*) [private], [slot]

Definition at line 139 of file mainwindow.cpp.

References KbWidget::device, kbWidgets, ui, and updateVersion().

Referenced by MainWindow().

```
                                    {
    foreach(KbWidget* w, kbWidgets){
        // Remove this device from the UI
        if(w->device == device){
            int i = kbWidgets.indexOf(w);
            ui->tabWidget->removeTab(i);
            kbWidgets.removeAt(i);
            w->deleteLater();
        }
    }
    // Update connected device count
    updateVersion();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.8 void MainWindow::showFwUpdateNotification (QWidget * *widget*, float *version*) [private],[slot]

Definition at line 192 of file mainwindow.cpp.

References KbWidget::device, kbWidgets, KbWidget::showFwUpdate(), KbWidget::showLastTab(), showWindow(), ui, and Kb::usbModel.

```
192                                                                 {
193     static bool isShowing = false;
194     if(isShowing)
195         return;
196     isShowing = true;
197     showWindow();
198     KbWidget* w = (KbWidget*)widget;
199     // Ask for update
200     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){
201         // If accepted, switch to firmware tab and bring up update window
202         w->showLastTab();
203         ui->tabWidget->setCurrentIndex(kbWidgets.indexOf(w));
204         w->showFwUpdate();
205     }
206     isShowing = false;
207 }
```

Here is the call graph for this function:

8.38.3.9 void MainWindow::showWindow ( ) [slot]

Definition at line 270 of file mainwindow.cpp.

Referenced by iconClicked(), MainWindow(), showFwUpdateNotification(), stateChange(), and timerTick().

```
270                                 {
271     showNormal();
272     raise();
273     activateWindow();
274 }
```

Here is the caller graph for this function:

8.38.3.10 void MainWindow::stateChange (Qt::ApplicationState *state*) [slot]

Definition at line 276 of file mainwindow.cpp.

References showWindow().

Referenced by MainWindow().

```
276                                                 {
277     // On OSX it's possible for the app to be brought to the foreground without the window actually
        reappearing.
278     // We want to make sure it's shown when this happens.
279 #ifdef Q_OS_MAC
280     static quint64 lastStateChange = 0;
281     quint64 now = QDateTime::currentMSecsSinceEpoch();
282     if(state == Qt::ApplicationActive){
283         // This happens once at startup so ignore it. Also don't allow it to be called more than once every
        2s.
284         if(lastStateChange != 0 && now >= lastStateChange + 2 * 1000)
285             showWindow();
286         lastStateChange = now;
287     }
288 #endif
289 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.11 void MainWindow::timerTick ( ) [private],[slot]

Definition at line 223 of file mainwindow.cpp.

References appShare, checkFwUpdates(), KbFirmware::checkUpdates(), CkbSettings::get(), SettingsWidget::pollUpdates(), settingsWidget, and showWindow().

Referenced by MainWindow().

```
223                         {
224     // Check shared memory for changes
225     if(appShare.lock()){
226         void* data = appShare.data();
227         QStringList commands = QString((const char*)data).split("\n");
228         // Restore PID, remove all other data
229         snprintf((char*)appShare.data(), appShare.size(), "PID %ld", (long)getpid());
230         appShare.unlock();
231         // Parse commands
232         foreach(const QString& line, commands){
233             // Old ckb option line - bring application to foreground
234             if(line == "Open")
235                 showWindow();
236             if(line.startsWith("Option ")){
237                 // New ckb option line
238                 QString option = line.split(" ")[1];
239                 if(option == "Open")
240                     // Bring to foreground
241                     showWindow();
242                 else if(option == "Close")
243                     // Quit application
244                     qApp->quit();
245             }
246         }
247     }
248     // Check for firmware updates (when appropriate)
249     if(!CkbSettings::get("Program/DisableAutoFWCheck").toBool()){
250         KbFirmware::checkUpdates();
251         checkFwUpdates();
252     }
253     // Poll for setting updates
254     settingsWidget->pollUpdates();
255 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.3.12 void MainWindow::toggleTrayIcon (bool *visible*)

Definition at line 112 of file mainwindow.cpp.

References trayIcon.

Referenced by MainWindow(), and ExtraSettingsWidget::on_trayBox_clicked().

```
112                                                 {
113 #ifdef USE_LIBAPPINDICATOR
114     if(unityDesktop)
115         app_indicator_set_status(indicator, visible ? APP_INDICATOR_STATUS_ACTIVE :
    APP_INDICATOR_STATUS_PASSIVE);
116     else
117 #endif // USE_LIBAPPINDICATOR
118         trayIcon->setVisible(visible);
119 }
```

Here is the caller graph for this function:

8.38.3.13 void MainWindow::updateVersion ( ) [private],[slot]

Definition at line 153 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().

```
153                                 {
154     QString daemonVersion = KbManager::ckbDaemonVersion();
155     if(daemonVersion == DAEMON_UNAVAILABLE_STR){
156         settingsWidget->setStatus("Driver inactive");
157         return;
158     }
159     int count = kbWidgets.count();
160     // Warn if the daemon version doesn't match the GUI
```

```
    QString daemonWarning;
    if(daemonVersion != CKB_VERSION_STR)
        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.";
    if(count == 0)
        settingsWidget->setStatus("No devices connected" + daemonWarning);
    else if(count == 1)
        settingsWidget->setStatus("1 device connected" + daemonWarning);
    else
        settingsWidget->setStatus(QString("%1 devices connected").arg(count) +
daemonWarning);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.38.4 Field Documentation

8.38.4.1 QAction* MainWindow::closeAction [private]

Definition at line 44 of file mainwindow.h.

Referenced by MainWindow().

8.38.4.2 QList<KbWidget*> MainWindow::kbWidgets [private]

Definition at line 41 of file mainwindow.h.

Referenced by addDevice(), checkFwUpdates(), cleanup(), removeDevice(), showFwUpdateNotification(), and updateVersion().

8.38.4.3 MainWindow * MainWindow::mainWindow = 0 [static]

Definition at line 35 of file mainwindow.h.

Referenced by checkFwUpdates(), MainWindow(), and ExtraSettingsWidget::on_trayBox_clicked().

8.38.4.4 QAction* MainWindow::restoreAction [private]

Definition at line 43 of file mainwindow.h.

Referenced by MainWindow().

8.38.4.5 SettingsWidget* MainWindow::settingsWidget [private]

Definition at line 40 of file mainwindow.h.

Referenced by MainWindow(), timerTick(), and updateVersion().

8.38.4.6 QSystemTrayIcon* MainWindow::trayIcon [private]

Definition at line 54 of file mainwindow.h.

Referenced by MainWindow(), and toggleTrayIcon().

8.38.4.7 QMenu* MainWindow::trayIconMenu [private]

Definition at line 53 of file mainwindow.h.

Referenced by MainWindow().

8.38.4.8 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

8.39 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

8.39.1 Detailed Description

Definition at line 13 of file modeselectdialog.h.

8.39.2 Constructor & Destructor Documentation

8.39.2.1 ModeSelectDialog::ModeSelectDialog (QWidget * *parent*, KbMode * *currentMode*, QList< KbMode * > *modeList*, const QString & *textLabel*)

Definition at line 4 of file modeselectdialog.cpp.

References _modeList, KbMode::name(), 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:

8.39.2.2 ModeSelectDialog::~ModeSelectDialog ( )

Definition at line 16 of file modeselectdialog.cpp.

References ui.

```
16                                      {
17     delete ui;
18 }
```

8.39.3 Member Function Documentation

8.39.3.1 void ModeSelectDialog::on_selAllButton_clicked ( ) [private],[slot]

Definition at line 30 of file modeselectdialog.cpp.

References ui.

```
30                                              {
31     int count = ui->modeList->count();
32     for(int i = 0; i < count; i++)
33         ui->modeList->item(i)->setSelected(true);
34 }
```

8.39.3.2 void ModeSelectDialog::on_selNoneButton_clicked ( ) [private],[slot]

Definition at line 36 of file modeselectdialog.cpp.

References ui.

```
36                                              {
37     int count = ui->modeList->count();
38     for(int i = 0; i < count; i++)
39         ui->modeList->item(i)->setSelected(false);
40 }
```

8.39.3.3 QList< KbMode * > ModeSelectDialog::selection ( )

Definition at line 20 of file modeselectdialog.cpp.

References _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:

8.39.4 Field Documentation

8.39.4.1 QList<KbMode*> ModeSelectDialog::_modeList [private]

Definition at line 31 of file modeselectdialog.h.

Referenced by ModeSelectDialog(), and selection().

8.39.4.2 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

8.40 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

8.40.1 Detailed Description

Definition at line 18 of file mperfwidget.h.

8.40.2 Data Structure Documentation

8.40.2.1 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	

8.40.3 Constructor & Destructor Documentation

8.40.3.1 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, enableChanged(), MPerfWidget::DpiUi::enableCheck, enableMapper, MPerfWidget::DpiUi::indicator, ColorButton::setLabel(), sliderXMapper, sliderXMoved(), sliderYMapper, sliderYMoved(), stages, ui, MPerfWidget::DpiUi::xBox, MPerfWidget::DpiUi::xSlider, MPerfWidget::DpiUi::yBox, and MPerfWidget::DpiUi::ySlider.

```
                                        :
    QWidget(parent), ui(new Ui::MPerfWidget), perf(0), profile(0), _xyLink(!CkbSettings::get(xyLinkPath).toBool()), colorLink(false), isSetting(false) {
     ui->setupUi(this);
     ui->xyBox->setChecked(!_xyLink);
     // Set up DPI stages
     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;
     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;
     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;
     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;
     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;
     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;
     ui->iButton0->setLabel(false);
     ui->iButton0->bigIcons(true);
     ui->iButton0->allowAlpha(true);
     for(int i = 0; i < DPI_COUNT; i++){
         stages[i].indicator->setLabel(false);
         stages[i].indicator->bigIcons(true);
         stages[i].indicator->allowAlpha(true);
         // Map signals
         connect(stages[i].indicator, SIGNAL(clicked(bool)), &buttonMapper1, SLOT(map()));
         connect(stages[i].indicator, SIGNAL(colorChanged(QColor)), &buttonMapper2, SLOT(map()));
         connect(stages[i].xSlider, SIGNAL(valueChanged(int)), &sliderXMapper, SLOT(map()));
         connect(stages[i].ySlider, SIGNAL(valueChanged(int)), &sliderYMapper, SLOT(map()));
         connect(stages[i].xBox, SIGNAL(valueChanged(int)), &boxXMapper, SLOT(map()));
         connect(stages[i].yBox, SIGNAL(valueChanged(int)), &boxYMapper, SLOT(map()));
         if(stages[i].enableCheck)
             // Sniper has no enable
             connect(stages[i].enableCheck, SIGNAL(stateChanged(int)), &enableMapper, SLOT(map()));
         // Set names
         buttonMapper1.setMapping(stages[i].indicator, i);
         buttonMapper2.setMapping(stages[i].indicator, i);
         sliderXMapper.setMapping(stages[i].xSlider, i);
         sliderYMapper.setMapping(stages[i].ySlider, i);
         boxXMapper.setMapping(stages[i].xBox, i);
         boxYMapper.setMapping(stages[i].yBox, i);
         if(stages[i].enableCheck)
             enableMapper.setMapping(stages[i].enableCheck, i);
     }
     // Connect to slots
     connect(&buttonMapper1, SIGNAL(mapped(int)), this, SLOT(colorClicked(int)));
     connect(&buttonMapper2, SIGNAL(mapped(int)), this, SLOT(colorChanged(int)));
     connect(&sliderXMapper, SIGNAL(mapped(int)), this, SLOT(sliderXMoved(int)));
     connect(&sliderYMapper, SIGNAL(mapped(int)), this, SLOT(sliderYMoved(int)));
     connect(&boxXMapper, SIGNAL(mapped(int)), this, SLOT(boxXChanged(int)));
     connect(&boxYMapper, SIGNAL(mapped(int)), this, SLOT(boxYChanged(int)));
     connect(&enableMapper, SIGNAL(mapped(int)), this, SLOT(enableChanged(int)));
}
```

Here is the call graph for this function:

8.40.3.2 MPerfWidget::~MPerfWidget ( )

Definition at line 56 of file mperfwidget.cpp.

References ui.

```
56                          {
57     delete ui;
58 }
```

8.40.4 Member Function Documentation

8.40.4.1 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:

8.40.4.2 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:

8.40.4.3 void MPerfWidget::colorChanged (int *index*) [private],[slot]

Definition at line 86 of file mperfwidget.cpp.

References ColorButton::color(), colorLink, DPI_COUNT, KbPerf::dpiColor(), 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:

8.40.4.4 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:

8.40.4.5 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:

8.40.4.6 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:

8.40.4.7 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:

8.40.4.8 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:

8.40.4.9 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:

8.40.4.10 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:

8.40.4.11 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:

8.40.4.12 void MPerfWidget::setPerf (KbPerf * *newPerf,* KbProfile * *newProfile*)

Definition at line 60 of file mperfwidget.cpp.

References _xyLink, KbPerf::angleSnap(), ColorButton::color(), KbPerf::dpi(), DPI_COUNT, KbPerf::dpiColor(), KbPerf::dpiEnabled(), KbPerf::dpiIndicator(), MPerfWidget::DpiUi::enableCheck, MPerfWidget::DpiUi::indicator, KbPerf::iOpacity(), KbPerf::liftHeight(), KbPerf::OTHER, perf, profile, stages, ui, MPerfWidget::DpiUi::xBox, and MPerfWidget::DpiUi::yBox.

```
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:

8.40.4.13 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:

8.40.4.14 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, 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:

8.40.5 Field Documentation

8.40.5.1 bool MPerfWidget::_xyLink [private]

Definition at line 44 of file mperfwidget.h.

Referenced by boxXChanged(), boxYChanged(), MPerfWidget(), on_xyBox_clicked(), setPerf(), sliderXMoved(), and sliderYMoved().

8.40.5.2 QSignalMapper MPerfWidget::boxXMapper [private]

Definition at line 49 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.3 QSignalMapper MPerfWidget::boxYMapper [private]

Definition at line 49 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.4 QSignalMapper MPerfWidget::buttonMapper1 [private]

Definition at line 47 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.5 QSignalMapper MPerfWidget::buttonMapper2 [private]

Definition at line 47 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.6 bool MPerfWidget::colorLink [private]

Definition at line 45 of file mperfwidget.h.

Referenced by colorChanged(), and colorClicked().

8.40.5.7 const int MPerfWidget::DPI_COUNT = KbPerf::DPI_COUNT [static]

Definition at line 28 of file mperfwidget.h.

Referenced by colorChanged(), MPerfWidget(), and setPerf().

8.40.5.8 const int MPerfWidget::DPI_MAX = KbPerf::DPI_MAX [static]

Definition at line 29 of file mperfwidget.h.

8.40.5.9 const int MPerfWidget::DPI_MIN = KbPerf::DPI_MIN [static]

Definition at line 29 of file mperfwidget.h.

8.40.5.10 QSignalMapper MPerfWidget::enableMapper [private]

Definition at line 50 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.11 bool MPerfWidget::isSetting [private]

Definition at line 53 of file mperfwidget.h.

8.40.5.12 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().

8.40.5.13 KbProfile* MPerfWidget::profile [private]

Definition at line 35 of file mperfwidget.h.

Referenced by on_copyButton_clicked(), and setPerf().

8.40.5.14 QSignalMapper MPerfWidget::sliderXMapper [private]

Definition at line 48 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.15 QSignalMapper MPerfWidget::sliderYMapper [private]

Definition at line 48 of file mperfwidget.h.

Referenced by MPerfWidget().

8.40.5.16 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().

8.40.5.17 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

8.41 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

8.41.1 Detailed Description

Definition at line 23 of file gradientdialog.h.

8.41.2 Constructor & Destructor Documentation

8.41.2.1 GradientDialog::Preset::Preset (QString *_name*, bool *_builtIn* = `false`) [inline]

Definition at line 27 of file gradientdialog.h.

```
27 : name(_name), builtIn(_builtIn) {}
```

8.41.2.2 GradientDialog::Preset::Preset (QString *_name*, const QGradientStops & *_gradient*) [inline]

Definition at line 28 of file gradientdialog.h.

```
28 : name(_name), gradient(_gradient), builtIn(false) {}
```

8.41.2.3 GradientDialog::Preset::Preset ( ) [inline]

Definition at line 29 of file gradientdialog.h.

```
29 : builtIn(false) {}
```

8.41.3 Field Documentation

8.41.3.1 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().

8.41.3.2 QGradientStops GradientDialog::Preset::gradient

Definition at line 25 of file gradientdialog.h.

Referenced by GradientDialog::GradientDialog(), GradientDialog::makeIcon(), GradientDialog::setPreset(), and GradientDialog::~GradientDialog().

8.41.3.3 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

8.42 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

8.43 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

8.44 QIODevice_descriptor Struct Reference

Collaboration diagram for QIODevice_descriptor:

Public Member Functions

- QIODevice_descriptor ()

Data Fields

- qint64 pos

8.44.1 Detailed Description

Definition at line 78 of file qioapi.cpp.

8.44.2 Constructor & Destructor Documentation

8.44.2.1 QIODevice_descriptor::QIODevice_descriptor ( ) [inline]

Definition at line 81 of file qioapi.cpp.

```
81                                  :
82          pos(0)
83      {}
```

8.44.3 Field Documentation

8.44.3.1 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

8.45 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

8.46 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

8.47 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

8.48 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

8.49 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

8.50 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

8.50.1 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.

8.50.2 Constructor & Destructor Documentation

8.50.2.1 QuaAdler32::QuaAdler32 ( )

Definition at line 30 of file quaadler32.cpp.

References reset().

```
31 {
32     reset();
33 }
```

Here is the call graph for this function:

8.50.3 Member Function Documentation

8.50.3.1 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 }
```

8.50.3.2 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:

8.50.3.3 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 }
```

8.50.3.4 quint32 QuaAdler32::value ( ) [virtual]

Returns

checksum

Implements QuaChecksum32.

Definition at line 50 of file quaadler32.cpp.

References checksum.

```
51 {
52     return checksum;
53 }
```

8.50.4 Field Documentation

8.50.4.1 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

8.51 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().

8.51.1 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.

8.51.2 Member Function Documentation

8.51.2.1 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.

8.51.2.2 virtual void QuaChecksum32::reset ( ) [pure virtual]

Implemented in QuaAdler32, and QuaCrc32.

8.51.2.3 virtual void QuaChecksum32::update (const QByteArray & *buf*) [pure virtual]

buf next portion of data from the stream

Implemented in QuaAdler32, and QuaCrc32.

8.51.2.4 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

8.52 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

8.52.1 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.

8.52.2 Constructor & Destructor Documentation

8.52.2.1 QuaCrc32::QuaCrc32 ( )

Definition at line 29 of file quacrc32.cpp.

References reset().

```
30 {
31     reset();
32 }
```

Here is the call graph for this function:

8.52.3 Member Function Documentation

8.52.3.1 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 }
```

8.52.3.2 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:

8.52.3.3 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 }
```

8.52.3.4 quint32 QuaCrc32::value ( ) [virtual]

Returns

checksum

Implements QuaChecksum32.

Definition at line 49 of file quacrc32.cpp.

References checksum.

```
50 {
51     return checksum;
52 }
```

8.52.4 Field Documentation

8.52.4.1 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

8.53 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

8.53.1 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.

8.53.2 Constructor & Destructor Documentation

8.53.2.1 QuaGzipFile::QuaGzipFile ( )

Must call setFileName() before trying to open.

Definition at line 87 of file quagzipfile.cpp.

```
87                          :
88 d(new QuaGzipFilePrivate())
89 {
90 }
```

8.53.2.2 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 }
```

8.53.2.3 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 }
```

8.53.2.4 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:

8.53.2.5 QuaGzipFile::QuaGzipFile (const QuaGzipFile & *that*) [private]

8.53.3 Member Function Documentation

8.53.3.1 void QuaGzipFile::close ( ) [virtual]

Definition at line 152 of file quagzipfile.cpp.

References d.

Referenced by ~QuaGzipFile().

```
{
  QIODevice::close();
  gzclose(d->gzd);
}
```

Here is the caller graph for this function:

8.53.3.2 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.

```
{
    return gzflush(d->gzd, Z_SYNC_FLUSH) == Z_OK;
}
```

8.53.3.3 QString QuaGzipFile::getFileName ( ) const

Definition at line 117 of file quagzipfile.cpp.

References d.

```
{
    return d->fileName;
}
```

8.53.3.4 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.

```
{
  return true;
}
```

8.53.3.5 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 }
```

8.53.3.6 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 }
```

8.53.3.7 QuaGzipFile& QuaGzipFile::operator= (const QuaGzipFile & *that*) [private]

8.53.3.8 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 }
```

8.53.3.9 void QuaGzipFile::setFileName (const QString & *fileName*)

Definition at line 112 of file quagzipfile.cpp.

References d.

```
113 {
114     d->fileName = fileName;
115 }
```

8.53.3.10 qint64 QuaGzipFile::writeData (const char * *data*, qint64 *maxSize*) [protected], [virtual]

Definition at line 163 of file quagzipfile.cpp.

References d.

```
{
    if (maxSize == 0)
        return 0;
    int written = gzwrite(d->gzd, (voidp)data, (unsigned)maxSize);
    if (written == 0)
        return -1;
    else
        return written;
}
```

8.53.4 Field Documentation

8.53.4.1 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

8.54 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

8.54.1 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.

8.54.2 Constructor & Destructor Documentation

8.54.2.1 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.

```
:
    QIODevice(parent),
    d(new QuaZIODevicePrivate(io))
{
  connect(io, SIGNAL(readyRead()), SIGNAL(readyRead()));
}
```

8.54.2.2 QuaZIODevice::~QuaZIODevice ( )

Definition at line 130 of file quaziodevice.cpp.

References close(), and d.

```
{
    if (isOpen())
        close();
    delete d;
}
```

Here is the call graph for this function:

8.54.3 Member Function Documentation

8.54.3.1 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().

```
{
    if ((openMode() & QIODevice::ReadOnly) != 0) {
        if (inflateEnd(&d->zins) != Z_OK) {
            setErrorString(d->zins.msg);
        }
    }
    if ((openMode() & QIODevice::WriteOnly) != 0) {
        flush();
        if (deflateEnd(&d->zouts) != Z_OK) {
            setErrorString(d->zouts.msg);
        }
    }
    QIODevice::close();
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.54.3.2 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().

```
{
    QString error;
    if (d->doFlush(error) < 0) {
```

```
        setErrorString(error);
        return false;
    }
    // can't flush buffer, some data is still waiting
    if (d->outBufPos < d->outBufSize)
        return true;
    Bytef c = 0;
    d->zouts.next_in = &c; // fake input buffer
    d->zouts.avail_in = 0; // of zero size
    do {
        d->zouts.next_out = (Bytef *) d->outBuf;
        d->zouts.avail_out = QUAZIO_OUTBUFSIZE;
        switch (deflate(&d->zouts, Z_SYNC_FLUSH)) {
        case Z_OK:
          d->outBufSize = (char *) d->zouts.next_out - d->outBuf;
          if (d->doFlush(error) < 0) {
              setErrorString(error);
              return false;
          }
          if (d->outBufPos < d->outBufSize)
              return true;
          break;
        case Z_BUF_ERROR: // nothing to write?
          return true;
        default:
          setErrorString(QString::fromLocal8Bit(d->zouts.msg));
          return false;
        }
    } while (d->zouts.avail_out == 0);
    return true;
}
```

Here is the caller graph for this function:

8.54.3.3 QIODevice * QuaZIODevice::getIoDevice ( ) const

Definition at line 137 of file quaziodevice.cpp.

References d.

```
{
    return d->io;
}
```

8.54.3.4 bool QuaZIODevice::isSequential ( ) const [virtual]

Definition at line 317 of file quaziodevice.cpp.

```
{
  return true;
}
```

8.54.3.5 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.

```
{
    if ((mode & QIODevice::Append) != 0) {
        setErrorString(trUtf8("QIODevice::Append is not supported for"
                    " QuaZIODevice"));
        return false;
    }
    if ((mode & QIODevice::ReadWrite) == QIODevice::ReadWrite) {
        setErrorString(trUtf8("QIODevice::ReadWrite is not supported for"
                    " QuaZIODevice"));
        return false;
    }
    if ((mode & QIODevice::ReadOnly) != 0) {
        if (inflateInit(&d->zins) != Z_OK) {
            setErrorString(d->zins.msg);
            return false;
        }
    }
    if ((mode & QIODevice::WriteOnly) != 0) {
        if (deflateInit(&d->zouts, Z_DEFAULT_COMPRESSION) != Z_OK) {
            setErrorString(d->zouts.msg);
            return false;
        }
    }
    return QIODevice::open(mode);
}
```

8.54.3.6 qint64 QuaZIODevice::readData (char * *data*, qint64 *maxSize*) [protected],[virtual]

Definition at line 185 of file quaziodevice.cpp.

References d, and QUAZIO_INBUFSIZE.

```
{
  int read = 0;
  while (read < maxSize) {
    if (d->inBufPos == d->inBufSize) {
      d->inBufPos = 0;
      d->inBufSize = d->io->read(d->inBuf, QUAZIO_INBUFSIZE);
      if (d->inBufSize == -1) {
        d->inBufSize = 0;
        setErrorString(d->io->errorString());
        return -1;
      }
      if (d->inBufSize == 0)
        break;
    }
    while (read < maxSize && d->inBufPos < d->inBufSize) {
      d->zins.next_in = (Bytef *) (d->inBuf + d->inBufPos);
      d->zins.avail_in = d->inBufSize - d->inBufPos;
      d->zins.next_out = (Bytef *) (data + read);
      d->zins.avail_out = (uInt) (maxSize - read); // hope it's less than 2GB
      int more = 0;
      switch (inflate(&d->zins, Z_SYNC_FLUSH)) {
      case Z_OK:
        read = (char *) d->zins.next_out - data;
        d->inBufPos = (char *) d->zins.next_in - d->inBuf;
        break;
      case Z_STREAM_END:
        read = (char *) d->zins.next_out - data;
        d->inBufPos = (char *) d->zins.next_in - d->inBuf;
        return read;
      case Z_BUF_ERROR: // this should never happen, but just in case
        if (!d->zBufError) {
          qWarning("Z_BUF_ERROR detected with %d/%d in/out, weird",
              d->zins.avail_in, d->zins.avail_out);
          d->zBufError = true;
        }
        memmove(d->inBuf, d->inBuf + d->inBufPos, d->inBufSize - d->inBufPos);
        d->inBufSize -= d->inBufPos;
        d->inBufPos = 0;
        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 }
```

8.54.3.7 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 }
```

8.54.4 Field Documentation

8.54.4.1 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

8.55 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

8.55.1 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.

8.55.2 Member Enumeration Documentation

8.55.2.1 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     };
```

8.55.2.2 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     };
```

8.55.2.3 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     };
```

8.55.3 Constructor & Destructor Documentation

8.55.3.1 QuaZip::QuaZip (const QuaZip & *that*) [private]

8.55.3.2 QuaZip::QuaZip ( )

Call setName() before opening constructed object.

Definition at line 193 of file quazip.cpp.

```
193              :
194   p(new QuaZipPrivate(this))
195 {
196 }
```

8.55.3.3 QuaZip::QuaZip (const QString & *zipName*)

Definition at line 198 of file quazip.cpp.

```
198                                  :
199   p(new QuaZipPrivate(this, zipName))
200 {
201 }
```

8.55.3.4 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 }
```

8.55.3.5 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:

8.55.4 Member Function Documentation

8.55.4.1 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().

```
{
  p->zipError=UNZ_OK;
  switch(p->mode) {
    case mdNotOpen:
      qWarning("QuaZip::close(): ZIP is not open");
      return;
    case mdUnzip:
      p->zipError=unzClose(p->unzFile_f);
      break;
    case mdCreate:
    case mdAppend:
    case mdAdd:
      p->zipError=zipClose(p->zipFile_f,
          p->comment.isNull() ? NULL
          : p->commentCodec->fromUnicode(p->comment).constData());
      break;
    default:
      qWarning("QuaZip::close(): unknown mode: %d", (int)p->mode);
      return;
  }
  // opened by name, need to delete the internal IO device
  if (!p->zipName.isEmpty()) {
      delete p->ioDevice;
      p->ioDevice = NULL;
  }
  p->clearDirectoryMap();
  if(p->zipError==UNZ_OK)
    p->mode=mdNotOpen;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.55.4.2 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().

```
{
  if (cs == csDefault) {
#ifdef Q_WS_WIN
      return Qt::CaseInsensitive;
#else
```

```
        return Qt::CaseSensitive;
#endif
    } else {
        return cs == csSensitive ? Qt::CaseSensitive : Qt::CaseInsensitive;
    }
}
```

Here is the caller graph for this function:

8.55.4.3 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.

```
{
  QuaZip *fakeThis=(QuaZip*)this; // non-const
  fakeThis->p->zipError=UNZ_OK;
  if(p->mode!=mdUnzip) {
    qWarning("QuaZip::getComment(): ZIP is not open in mdUnzip mode");
    return QString();
  }
  unz_global_info64 globalInfo;
  QByteArray comment;
  if((fakeThis->p->zipError=unzGetGlobalInfo64(p->unzFile_f, &globalInfo))!=
  UNZ_OK)
    return QString();
  comment.resize(globalInfo.size_comment);
  if((fakeThis->p->zipError=unzGetGlobalComment(p->unzFile_f, comment.data(),
  comment.size())) < 0)
    return QString();
  fakeThis->p->zipError = UNZ_OK;
  return p->commentCodec->toUnicode(comment);
}
```

Here is the call graph for this function:

8.55.4.4 QTextCodec * QuaZip::getCommentCodec ( ) const

Definition at line 597 of file quazip.cpp.

References QuaZipPrivate::commentCodec, and p.

Referenced by QuaZipFile::open().

```
{
  return p->commentCodec;
}
```

Here is the caller graph for this function:

8.55.4.5 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().

```
{
    QuaZipFileInfo64 info64;
    if (info == NULL) { // Very unlikely because of the overloads
        return false;
    }
    if (getCurrentFileInfo(&info64)) {
        info64.toQuaZipFileInfo(*info);
        return true;
    } else {
        return false;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.55.4.6 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.

```
{
  QuaZip *fakeThis=(QuaZip*)this; // non-const
  fakeThis->p->zipError=UNZ_OK;
  if(p->mode!=mdUnzip) {
    qWarning("QuaZip::getCurrentFileInfo(): ZIP is not open in mdUnzip mode");
    return false;
  }
  unz_file_info64 info_z;
  QByteArray fileName;
  QByteArray extra;
  QByteArray comment;
  if(info==NULL) return false;
  if(!isOpen()||!hasCurrentFile()) return false;
  if((fakeThis->p->zipError=unzGetCurrentFileInfo64(
  p->unzFile_f, &info_z, NULL, 0, NULL, 0, NULL, 0))!=UNZ_OK)
    return false;
  fileName.resize(info_z.size_filename);
  extra.resize(info_z.size_file_extra);
  comment.resize(info_z.size_file_comment);
  if((fakeThis->p->zipError=unzGetCurrentFileInfo64(
  p->unzFile_f, NULL,
      fileName.data(), fileName.size(),
      extra.data(), extra.size(),
      comment.data(), comment.size()))!=UNZ_OK)
    return false;
  info->versionCreated=info_z.version;
  info->versionNeeded=info_z.version_needed;
  info->flags=info_z.flag;
  info->method=info_z.compression_method;
  info->crc=info_z.crc;
  info->compressedSize=info_z.compressed_size;
  info->uncompressedSize=info_z.uncompressed_size;
  info->diskNumberStart=info_z.disk_num_start;
  info->internalAttr=info_z.internal_fa;
  info->externalAttr=info_z.external_fa;
  info->name=p->fileNameCodec->toUnicode(fileName);
  info->comment=p->commentCodec->toUnicode(comment);
  info->extra=extra;
  info->dateTime=QDateTime(
      QDate(info_z.tmu_date.tm_year, info_z.tmu_date.
  tm_mon+1, info_z.tmu_date.tm_mday),
      QTime(info_z.tmu_date.tm_hour, info_z.tmu_date.
  tm_min, info_z.tmu_date.tm_sec));
  // Add to directory map
  p->addCurrentFileToDirectoryMap(info->name);
  return true;
}
```

Here is the call graph for this function:

8.55.4.7 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().

```
{
  QuaZip *fakeThis=(QuaZip*)this; // non-const
  fakeThis->p->zipError=UNZ_OK;
  if(p->mode!=mdUnzip) {
    qWarning("QuaZip::getCurrentFileName(): ZIP is not open in mdUnzip mode");
    return QString();
  }
  if(!isOpen()||!hasCurrentFile()) return QString();
  QByteArray fileName(MAX_FILE_NAME_LENGTH, 0);
  if((fakeThis->p->zipError=unzGetCurrentFileInfo64(
  p->unzFile_f, NULL, fileName.data(), fileName.size(),
      NULL, 0, NULL, 0))!=UNZ_OK)
    return QString();
  QString result = p->fileNameCodec->toUnicode(fileName.constData());
  if (result.isEmpty())
      return result;
  // Add to directory map
  p->addCurrentFileToDirectoryMap(result);
  return result;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.55.4.8 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.

```
{
  QuaZip *fakeThis=(QuaZip*)this; // non-const
  fakeThis->p->zipError=UNZ_OK;
  if(p->mode!=mdUnzip) {
    qWarning("QuaZip::getEntriesCount(): ZIP is not open in mdUnzip mode");
    return -1;
```

```
  }
  unz_global_info64 globalInfo;
  if((fakeThis->p->zipError=unzGetGlobalInfo64(p->unzFile_f, &globalInfo))!=
  UNZ_OK)
    return p->zipError;
  return (int)globalInfo.number_entry;
}
```

Here is the call graph for this function:

8.55.4.9 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.

```
{
    QList<QuaZipFileInfo> list;
    if (p->getFileInfoList(&list))
        return list;
    else
        return QList<QuaZipFileInfo>();
}
```

Here is the call graph for this function:

8.55.4.10 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.

```
{
    QList<QuaZipFileInfo64> list;
    if (p->getFileInfoList(&list))
        return list;
    else
        return QList<QuaZipFileInfo64>();
}
```

Here is the call graph for this function:

8.55.4.11 QTextCodec * QuaZip::getFileNameCodec ( ) const

Definition at line 582 of file quazip.cpp.

References QuaZipPrivate::fileNameCodec, and p.

Referenced by QuaZipFile::open().

```
{
  return p->fileNameCodec;
}
```

Here is the caller graph for this function:

8.55.4.12 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:

8.55.4.13 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 }
```

8.55.4.14 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().

```
{
  return p->mode;
}
```

Here is the caller graph for this function:

8.55.4.15 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().

```
{
  return p->unzFile_f;
}
```

Here is the caller graph for this function:

8.55.4.16 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().

```
{
  return p->zipError;
}
```

Here is the caller graph for this function:

8.55.4.17 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:

8.55.4.18 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:

8.55.4.19 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:

8.55.4.20 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().

```
{
  p->zipError=UNZ_OK;
  if(p->mode!=mdUnzip) {
    qWarning("QuaZip::goToFirstFile(): ZIP is not open in mdUnzip mode");
    return false;
  }
  p->zipError=unzGoToNextFile(p->unzFile_f);
  p->hasCurrentFile_f=p->zipError==UNZ_OK;
  if(p->zipError==UNZ_END_OF_LIST_OF_FILE)
    p->zipError=UNZ_OK;
  return p->hasCurrentFile_f;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.55.4.21 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:

8.55.4.22 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 }
```

8.55.4.23 bool QuaZip::isDataDescriptorWritingEnabled ( ) const

See Also

setDataDescriptorWritingEnabled()

Definition at line 654 of file quazip.cpp.

References QuaZipPrivate::dataDescriptorWritingEnabled, and p.

Referenced by QuaZipFile::open().

```
{
    return p->dataDescriptorWritingEnabled;
}
```

Here is the caller graph for this function:

8.55.4.24 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().

```
{
  return p->mode!=mdNotOpen;
}
```

Here is the caller graph for this function:

8.55.4.25 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:

8.55.4.26 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().

```
{
  p->zipError=UNZ_OK;
  if(isOpen()) {
    qWarning("QuaZip::open(): ZIP already opened");
    return false;
  }
  QIODevice *ioDevice = p->ioDevice;
  if (ioDevice == NULL) {
    if (p->zipName.isEmpty()) {
      qWarning("QuaZip::open(): set either ZIP file name or IO device first");
      return false;
    } else {
      ioDevice = new QFile(p->zipName);
    }
  }
  unsigned flags = 0;
  switch(mode) {
    case mdUnzip:
      if (ioApi == NULL) {
          if (p->autoClose)
              flags |= UNZ_AUTO_CLOSE;
          p->unzFile_f=unzOpenInternal(ioDevice, NULL, 1, flags);
      } else {
          // QuaZIP pre-zip64 compatibility mode
          p->unzFile_f=unzOpen2(ioDevice, ioApi);
          if (p->unzFile_f != NULL) {
              if (p->autoClose) {
                  unzSetFlags(p->unzFile_f, UNZ_AUTO_CLOSE);
              } else {
                  unzClearFlags(p->unzFile_f, UNZ_AUTO_CLOSE);
              }
          }
      }
      if(p->unzFile_f!=NULL) {
        if (ioDevice->isSequential()) {
            unzClose(p->unzFile_f);
            if (!p->zipName.isEmpty())
                delete ioDevice;
            qWarning("QuaZip::open(): "
                     "only mdCreate can be used with "
                     "sequential devices");
            return false;
        }
        p->mode=mode;
        p->ioDevice = ioDevice;
        return true;
      } else {
        p->zipError=UNZ_OPENERROR;
        if (!p->zipName.isEmpty())
          delete ioDevice;
        return false;
      }
    case mdCreate:
    case mdAppend:
    case mdAdd:
      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:

8.55.4.27 QuaZip& QuaZip::operator= (const QuaZip & *that*) [private]

8.55.4.28 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 }
```

8.55.4.29 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 }
```

8.55.4.30 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 }
```

8.55.4.31 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 }
```

8.55.4.32 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(), 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().

```
{
  p->zipError=UNZ_OK;
  if(p->mode!=mdUnzip) {
    qWarning("QuaZip::setCurrentFile(): ZIP is not open in mdUnzip mode");
    return false;
  }
  if(fileName.isEmpty()) {
    p->hasCurrentFile_f=false;
    return true;
  }
  // Unicode-aware reimplementation of the unzLocateFile function
  if(p->unzFile_f==NULL) {
    p->zipError=UNZ_PARAMERROR;
    return false;
  }
  if(fileName.length()>MAX_FILE_NAME_LENGTH) {
    p->zipError=UNZ_PARAMERROR;
    return false;
  }
  // Find the file by name
  bool sens = convertCaseSensitivity(cs) == Qt::CaseSensitive;
  QString lower, current;
  if(!sens) lower=fileName.toLower();
  p->hasCurrentFile_f=false;

  // Check the appropriate Map
  unz64_file_pos fileDirPos;
  fileDirPos.pos_in_zip_directory = 0;
  if (sens) {
      if (p->directoryCaseSensitive.contains(fileName))
          fileDirPos = p->directoryCaseSensitive.value(fileName);
  } else {
      if (p->directoryCaseInsensitive.contains(lower))
          fileDirPos = p->directoryCaseInsensitive.value(lower);
  }

  if (fileDirPos.pos_in_zip_directory != 0) {
      p->zipError = unzGoToFilePos64(p->unzFile_f, &fileDirPos);
      p->hasCurrentFile_f = p->zipError == UNZ_OK;
  }
```

```
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:

8.55.4.33 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.

```
{
    p->dataDescriptorWritingEnabled = enabled;
}
```

8.55.4.34 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().

```
{
    QuaZipPrivate::defaultFileNameCodec = codec;
}
```

Here is the caller graph for this function:

8.55.4.35 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:

8.55.4.36 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 }
```

8.55.4.37 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 }
```

8.55.4.38 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:

8.55.4.39 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 }
```

8.55.4.40 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.

```
{
  if(isOpen()) {
    qWarning("QuaZip::setZipName(): ZIP is already open!");
    return;
  }
  p->zipName=zipName;
  p->ioDevice = NULL;
}
```

Here is the call graph for this function:

8.55.5 Friends And Related Function Documentation

8.55.5.1 friend class QuaZipPrivate [friend]

Definition at line 85 of file quazip.h.

8.55.6 Field Documentation

8.55.6.1 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

8.56 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

8.56.1 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.

8.56.2 Constructor & Destructor Documentation

8.56.2.1 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:

8.56.2.2 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 }
```

8.56.2.3 QuaZipDir::~QuaZipDir ( )

Definition at line 62 of file quazipdir.cpp.

```
63 {
64 }
```

8.56.3 Member Function Documentation

8.56.3.1 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:

8.56.3.2 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();
```

```
                ++i) {
            const QString &step = *i;
#ifdef QUAZIP_QUAZIPDIR_DEBUG
            qDebug("QuaZipDir::cd(%s): going to %s",
                    dirName.toUtf8().constData(),
                    step.toUtf8().constData());
#endif
            if (!dir.cd(step))
                return false;
        }
        d->dir = dir.path();
        return true;
    } else { // no '/'
        if (dirName == ".") {
            return true;
        } else if (dirName == "..") {
            if (isRoot()) {
                return false;
            } else {
                int slashPos = d->dir.lastIndexOf('/');
                if (slashPos == -1) {
                    d->dir = "";
                } else {
                    d->dir = d->dir.left(slashPos);
                }
                return true;
            }
        } else { // a simple subdirectory
            if (exists(dirName)) {
                if (isRoot())
                    d->dir = dirName;
                else
                    d->dir += "/" + dirName;
                return true;
            } else {
                return false;
            }
        }
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.56.3.3 bool QuaZipDir::cdUp ( )

Definition at line 150 of file quazipdir.cpp.

References cd().

```
{
    return cd("..");
}
```

Here is the call graph for this function:

8.56.3.4 uint QuaZipDir::count ( ) const

Definition at line 155 of file quazipdir.cpp.

References entryList().

```
{
    return entryList().count();
}
```

Here is the call graph for this function:

8.56.3.5 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:

8.56.3.6 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:

8.56.3.7 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:

8.56.3.8 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:

8.56.3.9 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:

8.56.3.10 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:

8.56.3.11 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().

```
{
    return entryList(QStringList(), filters, sort);
}
```

Here is the call graph for this function:

8.56.3.12 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().

```
{
    if (filePath == "/" || filePath.isEmpty())
        return true;
    QString fileName = filePath;
    if (fileName.endsWith('/'))
        fileName.chop(1);
    if (fileName.contains('/')) {
        QFileInfo fileInfo(fileName);
#ifdef QUAZIP_QUAZIPDIR_DEBUG
        qDebug("QuaZipDir::exists(): fileName=%s, fileInfo.fileName()=%s, "
                "fileInfo.path()=%s", fileName.toUtf8().constData(),
                fileInfo.fileName().toUtf8().constData(),
                fileInfo.path().toUtf8().constData());
#endif
        QuaZipDir dir(*this);
        return dir.cd(fileInfo.path()) && dir.exists(fileInfo.fileName());
    } else {
        if (fileName == "..") {
            return !isRoot();
        } else if (fileName == ".") {
            return true;
        } else {
            QStringList entries = entryList(QDir::AllEntries, QDir::NoSort);
#ifdef QUAZIP_QUAZIPDIR_DEBUG
            qDebug("QuaZipDir::exists(): looking for %s",
                    fileName.toUtf8().constData());
            for (QStringList::const_iterator i = entries.constBegin();
                    i != entries.constEnd();
                    ++i) {
                qDebug("QuaZipDir::exists(): entry: %s",
                        i->toUtf8().constData());
            }
#endif
            Qt::CaseSensitivity cs = QuaZip::convertCaseSensitivity(
                    d->caseSensitivity);
            if (filePath.endsWith('/')) {
                return entries.contains(filePath, cs);
            } else {
                return entries.contains(fileName, cs)
                    || entries.contains(fileName + "/", cs);
            }
        }
    }
}
```

Here is the call graph for this function:

8.56.3.13 bool QuaZipDir::exists ( ) const

Definition at line 495 of file quazipdir.cpp.

References d, and QuaZipDir().

Referenced by cd().

```
{
    return QuaZipDir(d->zip).exists(d->dir);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.56.3.14 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().

```
{
    return QDir(d->dir).filePath(fileName);
}
```

Here is the caller graph for this function:

8.56.3.15 QDir::Filters QuaZipDir::filter ( )

Definition at line 505 of file quazipdir.cpp.

References d.

```
506 {
507     return d->filter;
508 }
```

8.56.3.16 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:

8.56.3.17 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:

8.56.3.18 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);}
```

8.56.3.19 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 }
```

8.56.3.20 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 }
```

8.56.3.21 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:

8.56.3.22 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:

8.56.3.23 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 }
```

8.56.3.24 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:

8.56.3.25 void QuaZipDir::setFilter (QDir::Filters *filters*)

Definition at line 535 of file quazipdir.cpp.

References d.

```
536 {
537     d->filter = filters;
538 }
```

8.56.3.26 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:

8.56.3.27 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;
```

```
    if (newDir == "/") {
        d->dir = "";
    } else {
        if (newDir.endsWith('/'))
            newDir.chop(1);
        if (newDir.startsWith('/'))
            newDir = newDir.mid(1);
        d->dir = newDir;
    }
}
```

Here is the call graph for this function:

8.56.3.28 void QuaZipDir::setSorting (QDir::SortFlags *sort*)

Definition at line 559 of file quazipdir.cpp.

References d.

```
{
    d->sorting = sort;
}
```

8.56.3.29 QDir::SortFlags QuaZipDir::sorting ( ) const

Definition at line 564 of file quazipdir.cpp.

References d.

```
{
    return d->sorting;
}
```

8.56.4 Field Documentation

8.56.4.1 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

8.57 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::cs-Default, 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_WB-ITS, 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

8.57.1 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.

8.57.2 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.

8.57.3 Constructor & Destructor Documentation

8.57.3.1 QuaZipFile::QuaZipFile (const QuaZipFile & *that*) [private]

8.57.3.2 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 }
```

8.57.3.3 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 }
```

8.57.3.4 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 }
```

8.57.3.5 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 }
```

8.57.3.6 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.

```
                                                    :
  QIODevice(parent),
  p(new QuaZipFilePrivate(this, zip))
{
}
```

8.57.3.7 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.

```
{
  if (isOpen())
    close();
  delete p;
}
```

Here is the call graph for this function:

8.57.4 Member Function Documentation

8.57.4.1 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.

```
{
  if(p->zip==NULL) {
    qWarning("QuaZipFile::atEnd(): call setZipName() or setZip() first");
    return false;
  }
  if(!isOpen()) {
    qWarning("QuaZipFile::atEnd(): file is not open");
    return false;
  }
  if(openMode()&ReadOnly)
      // the same problem as with pos()
    return QIODevice::bytesAvailable() == 0
        && unzeof(p->zip->getUnzFile())==1;
```

```
  else
    return true;
}
```

Here is the call graph for this function:

8.57.4.2 qint64 QuaZipFile::bytesAvailable ( ) const [virtual]

Definition at line 499 of file quazipfile.cpp.

References pos(), and size().

```
{
    return size() - pos();
}
```

Here is the call graph for this function:

8.57.4.3 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().

```
{
  p->resetZipError();
  if(p->zip==NULL||!p->zip->isOpen()) return;
```

```
  if(!isOpen()) {
    qWarning("QuaZipFile::close(): file isn't open");
    return;
  }
  if(openMode()&ReadOnly)
    p->setZipError(unzCloseCurrentFile(p->zip->getUnzFile()));
  else if(openMode()&WriteOnly)
    if(isRaw()) p->setZipError(zipCloseFileInZipRaw64(p->zip->getZipFile(), p->uncompressedSize, p->crc));
    else p->setZipError(zipCloseFileInZip(p->zip->getZipFile()));
  else {
    qWarning("Wrong open mode: %d", (int)openMode());
    return;
  }
  if(p->zipError==UNZ_OK) setOpenMode(QIODevice::NotOpen);
  else return;
  if(p->internal) {
    p->zip->close();
    p->setZipError(p->zip->getZipError());
  }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.57.4.4 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().

```
{
  unz_file_info64 info_z;
  p->setZipError(UNZ_OK);
  if(p->zip==NULL||p->zip->getMode()!=QuaZip::mdUnzip) return -1;
  p->setZipError(unzGetCurrentFileInfo64(p->
  zip->getUnzFile(), &info_z, NULL, 0, NULL, 0, NULL, 0));
  if(p->zipError!=UNZ_OK)
    return -1;
  return info_z.compressed_size;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.57.4.5 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.

```
{
  p->setZipError(UNZ_OK);
  if (p->zip == NULL || (openMode() & WriteOnly))
    return QString();
  QString name=p->zip->getCurrentFileName();
  if(name.isNull())
    p->setZipError(p->zip->getZipError());
  return name;
}
```

Here is the call graph for this function:

8.57.4.6 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.

```
{
  return p->caseSensitivity;
}
```

8.57.4.7 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:

8.57.4.8 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:

8.57.4.9 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.

```
{
  return p->fileName;
}
```

8.57.4.10 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.

```
{
    return p->internal ? NULL : p->zip;
}
```

8.57.4.11 int QuaZipFile::getZipError ( ) const

Definition at line 494 of file quazipfile.cpp.

References p, and QuaZipFilePrivate::zipError.

Referenced by JlCompress::compressFile(), and JlCompress::extractFile().

```
{
  return p->zipError;
}
```

Here is the caller graph for this function:

8.57.4.12 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.

```
{
  return p->zip==NULL ? QString() : p->zip->getZipName();
}
```

Here is the call graph for this function:

8.57.4.13 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:

8.57.4.14 bool QuaZipFile::isSequential ( ) const [virtual]

Definition at line 337 of file quazipfile.cpp.

```
338 {
339   return true;
340 }
```

8.57.4.15 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(), and JICompress::extractFile().

```
222 {
223   return open(mode, NULL);
224 }
```

Here is the caller graph for this function:

8.57.4.16 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);}
```

8.57.4.17 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;
```

```
      }
      if(p->zip->getMode()!=QuaZip::mdUnzip) {
        qWarning("QuaZipFile::open(): file open mode %d incompatible with ZIP open mode %d",
            (int)mode, (int)p->zip->getMode());
        return false;
      }
      if(!p->zip->hasCurrentFile()) {
        qWarning("QuaZipFile::open(): zip does not have current file");
        return false;
      }
    }
    p->setZipError(unzOpenCurrentFile3(p->zip->
  getUnzFile(), method, level, (int)raw, password));
    if(p->zipError==UNZ_OK) {
      setOpenMode(mode);
      p->raw=raw;
      return true;
    } else
      return false;
  }
  qWarning("QuaZipFile::open(): open mode %d not supported by this function", (int)mode);
  return false;
}
```

Here is the call graph for this function:

8.57.4.18 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().

```
{
  zip_fileinfo info_z;
  p->resetZipError();
  if(isOpen()) {
    qWarning("QuaZipFile::open(): already opened");
    return false;
  }
  if((mode&WriteOnly)&&!(mode&ReadOnly)) {
    if(p->internal) {
      qWarning("QuaZipFile::open(): write mode is incompatible with internal QuaZip approach");
      return false;
    }
    if(p->zip==NULL) {
      qWarning("QuaZipFile::open(): zip is NULL");
      return false;
    }
    if(p->zip->getMode()!=QuaZip::mdCreate&&p->zip->getMode()!=QuaZip::mdAppend&&p->zip->getMode()!=QuaZip::mdAdd) {
      qWarning("QuaZipFile::open(): file open mode %d incompatible with ZIP open mode %d",
          (int)mode, (int)p->zip->getMode());
      return false;
    }
    info_z.tmz_date.tm_year=info.dateTime.date().year();
    info_z.tmz_date.tm_mon=info.dateTime.date().month() - 1;
    info_z.tmz_date.tm_mday=info.dateTime.date().day();
    info_z.tmz_date.tm_hour=info.dateTime.time().hour();
    info_z.tmz_date.tm_min=info.dateTime.time().minute();
    info_z.tmz_date.tm_sec=info.dateTime.time().second();
    info_z.dosDate = 0;
    info_z.internal_fa=(uLong)info.internalAttr;
    info_z.external_fa=(uLong)info.externalAttr;
    if (p->zip->isDataDescriptorWritingEnabled())
        zipSetFlags(p->zip->getZipFile(), ZIP_WRITE_DATA_DESCRIPTOR);
    else
        zipClearFlags(p->zip->getZipFile(), ZIP_WRITE_DATA_DESCRIPTOR);
    p->setZipError(zipOpenNewFileInZip3_64(p->zip->getZipFile(),
            p->zip->getFileNameCodec()->fromUnicode(info.name).constData(), &info_z,
            info.extraLocal.constData(), info.extraLocal.length(),
            info.extraGlobal.constData(), info.extraGlobal.length(),
            p->zip->getCommentCodec()->fromUnicode(info.comment).constData(),
            method, level, (int)raw,
            windowBits, memLevel, strategy,
            password, (uLong)crc, p->zip->isZip64Enabled()));
    if(p->zipError==UNZ_OK) {
      p->writePos=0;
      setOpenMode(mode);
      p->raw=raw;
      if(raw) {
        p->crc=crc;
        p->uncompressedSize=info.uncompressedSize;
      }
      return true;
    } else
      return false;
  }
  qWarning("QuaZipFile::open(): open mode %d not supported by this function", (int)mode);
  return false;
}
```

Here is the call graph for this function:

8.57.4.19 QuaZipFile& QuaZipFile::operator= (const QuaZipFile & *that*) [private]

8.57.4.20 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().

```
{
  if(p->zip==NULL) {
    qWarning("QuaZipFile::pos(): call setZipName() or setZip() first");
    return -1;
  }
  if(!isOpen()) {
    qWarning("QuaZipFile::pos(): file is not open");
    return -1;
  }
  if(openMode()&ReadOnly)
      // QIODevice::pos() is broken for sequential devices,
      // but thankfully bytesAvailable() returns the number of
      // bytes buffered, so we know how far ahead we are.
    return unztell(p->zip->getUnzFile()) - QIODevice::bytesAvailable();
  else
    return p->writePos;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.57.4.21 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.

```
{
  p->setZipError(UNZ_OK);
  qint64 bytesRead=unzReadCurrentFile(p->zip->getUnzFile(), data, (unsigned
  )maxSize);
  if (bytesRead < 0) {
    p->setZipError((int) bytesRead);
    return -1;
  }
  return bytesRead;
}
```

Here is the call graph for this function:

8.57.4.22 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.

```
{
  if(p->zip==NULL) {
    qWarning("QuaZipFile::setFileName(): call setZipName() first");
    return;
  }
  if(!p->internal) {
    qWarning("QuaZipFile::setFileName(): should not be used when not using internal QuaZip");
    return;
  }
  if(isOpen()) {
    qWarning("QuaZipFile::setFileName(): can not set file name for already opened file");
    return;
  }
  p->fileName=fileName;
  if (p->fileName.startsWith('/'))
      p->fileName = p->fileName.mid(1);
  p->caseSensitivity=cs;
}
```

8.57.4.23 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.

```
{
  if(isOpen()) {
    qWarning("QuaZipFile::setZip(): file is already open - can not set ZIP");
    return;
  }
  if(p->zip!=NULL && p->internal)
    delete p->zip;
  p->zip=zip;
  p->fileName=QString();
  p->internal=false;
}
```

8.57.4.24 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.

```
{
  if(isOpen()) {
    qWarning("QuaZipFile::setZipName(): file is already open - can not set ZIP name");
    return;
  }
  if(p->zip!=NULL && p->internal)
    delete p->zip;
  p->zip=new QuaZip(zipName);
  p->internal=true;
}
```

8.57.4.25 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().

```
{
  if(!isOpen()) {
    qWarning("QuaZipFile::atEnd(): file is not open");
    return -1;
  }
  if(openMode()&ReadOnly)
    return p->raw?csize():usize();
  else
    return p->writePos;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.57.4.26 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().

```
{
  unz_file_info64 info_z;
  p->setZipError(UNZ_OK);
  if(p->zip==NULL||p->zip->getMode()!=QuaZip::mdUnzip) return -1;
  p->setZipError(unzGetCurrentFileInfo64(p->zip->getUnzFile(), &info_z, NULL, 0, NULL, 0, NULL, 0));
  if(p->zipError!=UNZ_OK)
    return -1;
  return info_z.uncompressed_size;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.57.4.27 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().

```
{
  p->setZipError(ZIP_OK);
  p->setZipError(zipWriteInFileInZip(p->zip->
  getZipFile(), data, (uint)maxSize));
  if(p->zipError!=ZIP_OK) return -1;
  else {
    p->writePos+=maxSize;
    return maxSize;
  }
}
```

Here is the call graph for this function:

8.57.5 Friends And Related Function Documentation

8.57.5.1 friend class QuaZipFilePrivate [friend]

Definition at line 75 of file quazipfile.h.

8.57.6 Field Documentation

8.57.6.1 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

8.58 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.

8.58.1 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.

8.58.2 Member Function Documentation

8.58.2.1 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().

```
{
    return permissionsFromExternalAttr(externalAttr);
}
```

Here is the call graph for this function:

8.58.3 Field Documentation

8.58.3.1 QString QuaZipFileInfo::comment

Definition at line 67 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.2 quint32 QuaZipFileInfo::compressedSize

Definition at line 57 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.3 quint32 QuaZipFileInfo::crc

Definition at line 55 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.4 QDateTime QuaZipFileInfo::dateTime

Definition at line 53 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.5 quint16 QuaZipFileInfo::diskNumberStart

Definition at line 61 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.6 quint32 QuaZipFileInfo::externalAttr

Definition at line 65 of file quazipfileinfo.h.

Referenced by getPermissions(), and QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.7 QByteArray QuaZipFileInfo::extra

Definition at line 69 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.8 quint16 QuaZipFileInfo::flags

Definition at line 49 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.9 quint16 QuaZipFileInfo::internalAttr

Definition at line 63 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.10 quint16 QuaZipFileInfo::method

Definition at line 51 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.11 QString QuaZipFileInfo::name

Definition at line 43 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.12 quint32 QuaZipFileInfo::uncompressedSize

Definition at line 59 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.13 quint16 QuaZipFileInfo::versionCreated

Definition at line 45 of file quazipfileinfo.h.

Referenced by QuaZipFileInfo64::toQuaZipFileInfo().

8.58.3.14 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

8.59 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.

8.59.1 Detailed Description

Call QuaZip::getCurrentFileInfo() or QuaZipFile::getFileInfo() to fill this structure.

Definition at line 81 of file quazipfileinfo.h.

8.59.2 Member Function Documentation

8.59.2.1 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().

```
{
    return getNTFSTime(extra, 8, fineTicks);
}
```

Here is the call graph for this function:

8.59.2.2 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().

```
{
    return getNTFSTime(extra, 16, fineTicks);
}
```

Here is the call graph for this function:

8.59.2.3 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:

8.59.2.4 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:

8.59.2.5 bool QuaZipFileInfo64::isEncrypted ( ) const [inline]

Definition at line 175 of file quazipfileinfo.h.

```
175 {return (flags & 1) != 0;}
```

8.59.2.6 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;
```

```
    info.externalAttr = externalAttr;
    info.comment = comment;
    info.extra = extra;
    return noOverflow;
}
```

Here is the caller graph for this function:

8.59.3 Field Documentation

8.59.3.1 QString QuaZipFileInfo64::comment

Definition at line 114 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), and toQuaZipFileInfo().

8.59.3.2 quint64 QuaZipFileInfo64::compressedSize

Definition at line 104 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.3 quint32 QuaZipFileInfo64::crc

Definition at line 102 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.4 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().

8.59.3.5 quint16 QuaZipFileInfo64::diskNumberStart

Definition at line 108 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.6 quint32 QuaZipFileInfo64::externalAttr

Definition at line 112 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), getPermissions(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.7 QByteArray QuaZipFileInfo64::extra

Definition at line 116 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), getNTFSaTime(), getNTFScTime(), getNTFSmTime(), and toQuaZipFileInfo().

8.59.3.8 quint16 QuaZipFileInfo64::flags

Definition at line 89 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.9 quint16 QuaZipFileInfo64::internalAttr

Definition at line 110 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.10 quint16 QuaZipFileInfo64::method

Definition at line 91 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.11 QString QuaZipFileInfo64::name

Definition at line 83 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), JlCompress::getFileList(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.12 quint64 QuaZipFileInfo64::uncompressedSize

Definition at line 106 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.13 quint16 QuaZipFileInfo64::versionCreated

Definition at line 85 of file quazipfileinfo.h.

Referenced by QuaZip::getCurrentFileInfo(), QuaZipDir_getFileInfo(), and toQuaZipFileInfo().

8.59.3.14 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

8.60 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

8.60.1 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.

8.60.2 Constructor & Destructor Documentation

8.60.2.1 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) {}
```

8.60.2.2 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        }
```

8.60.2.3 QuaZipFilePrivate::QuaZipFilePrivate (QuaZipFile * *q*, const QString & *zipName*, const QString & *fileName*, QuaZip::CaseSensitivity *cs*) [inline],[private]

Definition at line 87 of file quazipfile.cpp.

```
                                :
    q(q), internal(true), zipError(UNZ_OK)
    {
      zip=new QuaZip(zipName);
      this->fileName=fileName;
      if (this->fileName.startsWith('/'))
          this->fileName = this->fileName.mid(1);
      this->caseSensitivity=cs;
    }
```

8.60.2.4 QuaZipFilePrivate::QuaZipFilePrivate (QuaZipFile * *q*, QuaZip * *zip*) [inline],[private]

Definition at line 98 of file quazipfile.cpp.

```
                                                    :
    q(q), zip(zip), internal(false), zipError(UNZ_OK) {}
```

8.60.2.5 QuaZipFilePrivate::~QuaZipFilePrivate ( ) [inline],[private]

Definition at line 101 of file quazipfile.cpp.

```
  {
    if (internal)
      delete zip;
  }
```

8.60.3 Member Function Documentation

8.60.3.1 void QuaZipFilePrivate::resetZipError ( ) const [inline],[private]

Definition at line 69 of file quazipfile.cpp.

References UNZ_OK.

Referenced by QuaZipFile::close(), and QuaZipFile::open().

```
{setZipError(UNZ_OK);}
```

Here is the caller graph for this function:

8.60.3.2 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().

```
{
  QuaZipFilePrivate *fakeThis = const_cast<QuaZipFilePrivate*>(this); //
   non-const
  fakeThis->zipError=zipError;
  if(zipError==UNZ_OK)
    q->setErrorString(QString());
  else
    q->setErrorString(QuaZipFile::tr("ZIP/UNZIP API error %1").arg(zipError));
}
```

Here is the caller graph for this function:

8.60.4 Friends And Related Function Documentation

8.60.4.1 friend class QuaZipFile [friend]

Definition at line 38 of file quazipfile.cpp.

8.60.5 Field Documentation

8.60.5.1 QuaZip::CaseSensitivity QuaZipFilePrivate::caseSensitivity [private]

Definition at line 47 of file quazipfile.cpp.

Referenced by QuaZipFile::getCaseSensitivity(), QuaZipFile::open(), and QuaZipFile::setFileName().

8.60.5.2 quint32 QuaZipFilePrivate::crc [private]

Definition at line 59 of file quazipfile.cpp.

Referenced by QuaZipFile::close(), and QuaZipFile::open().

8.60.5.3 QString QuaZipFilePrivate::fileName [private]

Definition at line 45 of file quazipfile.cpp.

Referenced by QuaZipFile::getFileName(), QuaZipFile::open(), QuaZipFile::setFileName(), and QuaZipFile::setZip().

8.60.5.4 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().

8.60.5.5 QuaZipFile* QuaZipFilePrivate::q [private]

Definition at line 41 of file quazipfile.cpp.

Referenced by setZipError().

8.60.5.6 bool QuaZipFilePrivate::raw [private]

Definition at line 49 of file quazipfile.cpp.

Referenced by QuaZipFile::isRaw(), QuaZipFile::open(), and QuaZipFile::size().

8.60.5.7 quint64 QuaZipFilePrivate::uncompressedSize [private]

Definition at line 57 of file quazipfile.cpp.

Referenced by QuaZipFile::close(), and QuaZipFile::open().

8.60.5.8 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().

8.60.5.9 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().

8.60.5.10 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

8.61 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.

8.61.1 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.

8.61.2 Constructor & Destructor Documentation

8.61.2.1 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 }
```

8.61.2.2 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:

8.61.2.3 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:

8.61.2.4 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:

8.61.3 Member Function Documentation

8.61.3.1 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 }
```

8.61.3.2 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:

8.61.3.3 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:

8.61.3.4 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().

```
{
    setNTFSTime(extraLocal, mTime, 0, fineTicks);
    setNTFSTime(extraGlobal, mTime, 0, fineTicks);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.61.3.5 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:

8.61.3.6 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().

```
{
    QFileInfo info = QFileInfo(file);
    QFile::Permissions perm = info.permissions();
    QuaZipNewInfo_setPermissions(this, perm, info.isDir());
}
```

Here is the call graph for this function:

8.61.3.7 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().

```
{
    QuaZipNewInfo_setPermissions(this, permissions,
  name.endsWith('/'));
}
```

Here is the call graph for this function:

8.61.4 Field Documentation

8.61.4.1 QString QuaZipNewInfo::comment

Will be encoded using QuaZip::getCommentCodec().

Definition at line 75 of file quazipnewinfo.h.

Referenced by QuaZipFile::open().

8.61.4.2 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().

8.61.4.3 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().

8.61.4.4 QByteArray QuaZipNewInfo::extraGlobal

Definition at line 79 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), setFileNTFSaTime(), setFileNTFScTime(), and setFileNTFSmTime().

8.61.4.5 QByteArray QuaZipNewInfo::extraLocal

Definition at line 77 of file quazipnewinfo.h.

Referenced by QuaZipFile::open(), setFileNTFSaTime(), setFileNTFScTime(), and setFileNTFSmTime().

8.61.4.6 quint16 QuaZipNewInfo::internalAttr

Definition at line 64 of file quazipnewinfo.h.

Referenced by QuaZipFile::open().

8.61.4.7 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().

8.61.4.8 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

8.62 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

8.62.1 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.

8.62.2 Constructor & Destructor Documentation

8.62.2.1 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      }
```

8.62.2.2 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      }
```

8.62.2.3 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      }
```

8.62.3 Member Function Documentation

8.62.3.1 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().

```
{
    if (!hasCurrentFile_f || fileName.isEmpty()) {
        return;
    }
    // Adds current file to filename map as fileName
    unz64_file_pos fileDirectoryPos;
    unzGetFilePos64(unzFile_f, &fileDirectoryPos);
    directoryCaseSensitive.insert(fileName, fileDirectoryPos);
    // Only add lowercase to directory map if not already there
    // ensures only map the first one seen
    QString lower = fileName.toLower();
    if (!directoryCaseInsensitive.contains(lower))
        directoryCaseInsensitive.insert(lower, fileDirectoryPos);
    // Mark last one
    if (fileDirectoryPos.pos_in_zip_directory >
  lastMappedDirectoryEntry.pos_in_zip_directory)
        lastMappedDirectoryEntry = fileDirectoryPos;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.62.3.2 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().

```
{
    directoryCaseInsensitive.clear();
    directoryCaseSensitive.clear();
    lastMappedDirectoryEntry.num_of_file = 0;
    lastMappedDirectoryEntry.pos_in_zip_directory = 0;
}
```

Here is the caller graph for this function:

8.62.3.3 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     }
```

8.62.3.4 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;
```

```
  if (currentFile.isEmpty()) {
      if (!q->goToFirstFile())
          return false;
  } else {
      if (!q->setCurrentFile(currentFile))
          return false;
  }
  return true;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.62.3.5 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().

```
{
    zipError = UNZ_OK;
    if (mode != QuaZip::mdUnzip) {
        qWarning("QuaZipPrivate::goToNextUnmappedFile(): ZIP is not open in mdUnzip mode");
        return false;
    }
    // If not mapped anything, go to beginning
    if (lastMappedDirectoryEntry.pos_in_zip_directory == 0) {
        unzGoToFirstFile(unzFile_f);
    } else {
        // Goto the last one mapped, plus one
        unzGoToFilePos64(unzFile_f, &lastMappedDirectoryEntry);
        unzGoToNextFile(unzFile_f);
    }
```

```
    hasCurrentFile_f=zipError==UNZ_OK;
    if(zipError==UNZ_END_OF_LIST_OF_FILE)
      zipError=UNZ_OK;
    return hasCurrentFile_f;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.62.4 Friends And Related Function Documentation

8.62.4.1 friend class QuaZip [friend]

Definition at line 40 of file quazip.cpp.

8.62.5 Field Documentation

8.62.5.1 union { ... } [private]

8.62.5.2 bool QuaZipPrivate::autoClose [private]

Definition at line 71 of file quazip.cpp.

Referenced by QuaZip::isAutoClose(), QuaZip::open(), and QuaZip::setAutoClose().

8.62.5.3 QString QuaZipPrivate::comment [private]

Definition at line 53 of file quazip.cpp.

Referenced by QuaZip::close(), and QuaZip::setComment().

8.62.5.4 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().

8.62.5.5 bool QuaZipPrivate::dataDescriptorWritingEnabled [private]

Definition at line 67 of file quazip.cpp.

Referenced by QuaZip::isDataDescriptorWritingEnabled(), QuaZip::open(), and QuaZip::setDataDescriptorWritingEnabled().

8.62.5.6 QTextCodec * QuaZipPrivate::defaultFileNameCodec = NULL [static], [private]

Definition at line 140 of file quazip.cpp.

Referenced by getDefaultFileNameCodec(), and QuaZip::setDefaultFileNameCodec().

8.62.5.7 QHash<QString, unz64_file_pos> QuaZipPrivate::directoryCaseInsensitive [private]

Definition at line 138 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), clearDirectoryMap(), and QuaZip::setCurrentFile().

8.62.5.8 QHash<QString, unz64_file_pos> QuaZipPrivate::directoryCaseSensitive [private]

Definition at line 137 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), clearDirectoryMap(), and QuaZip::setCurrentFile().

8.62.5.9 QTextCodec* QuaZipPrivate::fileNameCodec [private]

Definition at line 45 of file quazip.cpp.

Referenced by QuaZip::getCurrentFileInfo(), QuaZip::getCurrentFileName(), QuaZip::getFileNameCodec(), and QuaZip::setFileNameCodec().

8.62.5.10 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().

8.62.5.11 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().

8.62.5.12 unz64_file_pos QuaZipPrivate::lastMappedDirectoryEntry [private]

Definition at line 139 of file quazip.cpp.

Referenced by addCurrentFileToDirectoryMap(), clearDirectoryMap(), goToFirstUnmappedFile(), and QuaZipPrivate().

8.62.5.13 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().

8.62.5.14 QuaZip* QuaZipPrivate::q [private]

Definition at line 43 of file quazip.cpp.

Referenced by getFileInfoList().

8.62.5.15 bool QuaZipPrivate::zip64 [private]

Definition at line 69 of file quazip.cpp.

Referenced by QuaZip::isZip64Enabled(), and QuaZip::setZip64Enabled().

8.62.5.16 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().

8.62.5.17 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

8.63 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

8.64 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 if 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.

Private Member Functions

- void setBox (QWidget *box)
- void convertMacroBox ()

  RebindWidget::convertMacroBox converts the macroBox content. The KB sends each keypress as "key [+|-]<keyname><newline>".
- void helpStatus (int status)

  RebindWidget::helpStatus shows a help line in the ui.

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

8.64.1 Detailed Description

Definition at line 15 of file rebindwidget.h.

8.64.2 Constructor & Destructor Documentation

8.64.2.1 RebindWidget::RebindWidget (QWidget * *parent*) [explicit]

Definition at line 9 of file rebindwidget.cpp.

References fnKeys, KeyMap::friendlyName(), mediaKeys, modKeys, mouseExtKeys, mouseKeys, numKeys, 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>");
```

```
#else
    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>");
#endif
}
```

Here is the call graph for this function:

8.64.2.2 RebindWidget::~RebindWidget ( )

Definition at line 59 of file rebindwidget.cpp.

References ui.

```
59                              {
60     delete ui;
61 }
```

8.64.3 Member Function Documentation

8.64.3.1 void RebindWidget::applyChanges (const QStringList & *keys*, bool *doUnbind*) [private],[slot]

Definition at line 327 of file rebindwidget.cpp.

References KeyAction::animAction(), KbLight::animList(), bind, DPI_OFFSET, KeyAction::dpiAction(), fnKeys, KbAnim::guid(), KbBind::light(), KeyAction::lightAction(), KeyAction::lockAction(), KeyAction::macroAction(), mediaKeys, KeyAction::modeAction(), modKeys, mouseExtKeys, mouseKeys, KbBind::noAction(), numKeys, KeyAction::PROGRAM_PR_MULTI, KeyAction::PROGRAM_RE_INDEF, KeyAction::PROGRAM_RE_KPSTOP, KeyAction::PROGRAM_RE_MULTI, KeyAction::programAction(), KbBind::setAction(), typingKeys, ui, and wheelKeys.

Referenced by on_applyButton_clicked(), and setSelection().

```
327                                                                 {
328     if(ui->typingBox->currentIndex() > 0)
329         bind->setAction(keys, typingKeys[ui->typingBox->currentIndex() - 1]);
330     else if(ui->modBox->currentIndex() > 0)
331         bind->setAction(keys, modKeys[ui->modBox->currentIndex() - 1]);
332     else if(ui->fnBox->currentIndex() > 0)
333         bind->setAction(keys, fnKeys[ui->fnBox->currentIndex() - 1]);
334     else if(ui->numBox->currentIndex() > 0)
335         bind->setAction(keys, numKeys[ui->numBox->currentIndex() - 1]);
336     else if(ui->mediaBox->currentIndex() > 0)
337         bind->setAction(keys, mediaKeys[ui->mediaBox->currentIndex() - 1]);
338     else if(ui->mbBox->currentIndex() > 0)
339         bind->setAction(keys, mouseKeys[ui->mbBox->currentIndex() - 1]);
340     else if(ui->mb2Box->currentIndex() > 0)
341         bind->setAction(keys, mouseExtKeys[ui->mb2Box->currentIndex() - 1]);
342     else if(ui->wheelBox->currentIndex() > 0)
343         bind->setAction(keys, wheelKeys[ui->wheelBox->currentIndex() - 1]);
344     else if(ui->dpiBox->currentIndex() > 0)
345         bind->setAction(keys, KeyAction::dpiAction(
    ui->dpiBox->currentIndex() - DPI_OFFSET, ui->dpiCustXBox->value(),
    ui->dpiCustYBox->value()));
```

```
    else if(ui->modeBox->currentIndex() > 0)
        bind->setAction(keys, KeyAction::modeAction(ui->modeBox->currentIndex() - 3 - (ui->modeWrapBox->isChecked() && ui->modeBox->currentIndex() < 3 ? 2 : 0)));
    else if(ui->lightBox->currentIndex() > 0)
        bind->setAction(keys, KeyAction::lightAction(ui->lightBox->currentIndex() - 1 + (ui->lightWrapBox->isChecked() ? 2 : 0)));
    else if(ui->lockBox->currentIndex() > 0)
        bind->setAction(keys, KeyAction::lockAction(ui->lockBox->currentIndex() - 1));
    else if(ui->animBox->currentIndex() > 0){
        KbAnim* anim = bind->light()->animList().at(ui->animBox->currentIndex() - 1);
        bind->setAction(keys, KeyAction::animAction(anim->guid(), !ui->animOnceBox->isChecked(), ui->animKrBox->isChecked()));
    } else if(!ui->programKpBox->text().isEmpty() || !ui->programKrBox->text().isEmpty()){
        int kpStop = 0, krStop = 0;
        if(!ui->programKpBox->text().isEmpty()){
            if(!ui->programKpSIBox->isChecked())
                kpStop = KeyAction::PROGRAM_PR_MULTI;
            else
                kpStop = ui->programKpModeBox->currentIndex();
        }
        if(!ui->programKrBox->text().isEmpty()){
            if(!ui->programKrSIBox->isChecked())
                krStop = KeyAction::PROGRAM_RE_MULTI;
            else if(ui->programKrModeBox->currentIndex() == 0)
                krStop = KeyAction::PROGRAM_RE_INDEF;
            else
                krStop = KeyAction::PROGRAM_RE_KPSTOP;
        }
        bind->setAction(keys, KeyAction::programAction(ui->programKpBox->text(), ui->programKrBox->text(), kpStop | krStop));
    } else if (ui->pteMacroBox->toPlainText().length() > 0) {
        // 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.
        QString mac;
        mac = ui->pteMacroComment->toPlainText().replace(":", "&das_IST_31N_col0n;");
        mac = ui->pteMacroText->toPlainText().replace(":", "&das_IST_31N_col0n;") + ":" + mac;
        mac = ui->pteMacroBox->toPlainText() + ":" + mac;
        bind->setAction(keys, KeyAction::macroAction(mac));
    } else if(doUnbind)
        bind->noAction(keys);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.64.3.2 void RebindWidget::convertMacroBox ( ) [private]

the ckb-daemon needs a shorter format, only " [+|-]<keyname>"

That function does the conversion.

Definition at line 776 of file rebindwidget.cpp.

References ui.

Referenced by on_btnStopMacro_clicked().

```
776                                     {
777     QString in;
778
779     in = ui->pteMacroBox->toPlainText();
780     in.replace (QRegExp("\n"), ",");
781     in.replace (QRegExp("key "), "");
782     ui->pteMacroBox->setPlainText(in);
783 }
```

Here is the caller graph for this function:

8.64.3.3 void RebindWidget::helpStatus (int *status*) [private]

Parameters

status	determines what to display.

Definition at line 752 of file rebindwidget.cpp.

References ui.

Referenced by on_btnClearMacro_clicked(), on_btnStartMacro_clicked(), on_btnStopMacro_clicked(), and setBox().

```
752                                         {
753     switch (status) {
754     case 1:
755         ui->lbl_macro->setText("Type in a macro name in the comment box and click start.");
756         break;
757     case 2:
758         ui->lbl_macro->setText("Type your macro and click stop when finished.");
```

```
        break;
    case 3:
        ui->lbl_macro->setText("Click Apply or change values in Macro Key Actions in advance.");
        break;
    default:
        ui->lbl_macro->setText(QString("Oops: Some magic in RebindWidget::helpStatus (%1)").arg(status));
    }
}
```

Here is the caller graph for this function:

8.64.3.4 void RebindWidget::on_animBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 592 of file rebindwidget.cpp.

References setBox(), and ui.

```
592                                                     {
```

```
    if(index == 0)
        ui->animButton->setChecked(false);
    else {
        ui->animButton->setChecked(true);
        setBox(ui->animBox);
    }
}
```

Here is the call graph for this function:

8.64.3.5 void RebindWidget::on_animButton_clicked (bool *checked*) [private],[slot]

Definition at line 689 of file rebindwidget.cpp.

References ui.

```
689                                                 {
690     if(checked && ui->animBox->currentIndex() == 0)
691         ui->animBox->setCurrentIndex(1);
692 }
```

8.64.3.6 void RebindWidget::on_applyButton_clicked ( ) [private],[slot]

Definition at line 386 of file rebindwidget.cpp.

References applyChanges(), on_btnStopMacro_clicked(), and selection.

```
386                                         {
387     // Normally, this should be done via signalling.
388     // Because there is no serarate thread, we have to call it directly
389     // (otherwise we could do Key char conversion step by step,
390     // but so it is more easy to change the key definition):
391     on_btnStopMacro_clicked();
392     applyChanges(selection, true);
393 }
```

Here is the call graph for this function:

8.64.3.7 void RebindWidget::on_btnClearMacro_clicked ( ) [private],[slot]

Definition at line 744 of file rebindwidget.cpp.

References helpStatus().

```
744                                         {
745     helpStatus(1);
746 }
```

Here is the call graph for this function:

8.64.3.8 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 703 of file rebindwidget.cpp.

References bind, KbBind::getMacroNumber(), KbBind::getMacroPath(), KbBind::handleNotificationChannel(), helpStatus(), macReader, and ui.

```
703                                         {
704     if (!macReader) {
705         bind->handleNotificationChannel(true);
706         macReader = new MacroReader(bind->getMacroNumber(),
      bind->getMacroPath(), ui->pteMacroBox, ui->pteMacroText);
707         // because of the second thread we need to disable three of the four bottom buttons.
708         // Clicking "Stop" will enable them again.
709         ui->applyButton->setEnabled(false);
710         ui->resetButton->setEnabled(false);
711         ui->unbindButton->setEnabled(false);
712         ui->btnStartMacro->setEnabled(false);
713         ui->btnStopMacro->setEnabled(true);
714         helpStatus(2);
715     }
716 }
```

Here is the call graph for this function:

8.64.3.9 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 725 of file rebindwidget.cpp.

References bind, convertMacroBox(), KbBind::handleNotificationChannel(), helpStatus(), macReader, and ui.

Referenced by on_applyButton_clicked().

```
725                                         {
726     if (macReader) {
727         bind->handleNotificationChannel(false);
728         delete macReader;
729         macReader = 0;
730         convertMacroBox();
731         ui->applyButton->setEnabled(true);
732         ui->resetButton->setEnabled(true);
733         ui->unbindButton->setEnabled(true);
734         ui->btnStartMacro->setEnabled(true);
735         ui->btnStopMacro->setEnabled(false);
736         helpStatus(3);
737     }
738 }
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.64.3.10 void RebindWidget::on_cancelButton_clicked ( ) [private],[slot]

Definition at line 395 of file rebindwidget.cpp.

References selection, and setSelection().

```
395                                         {
396     // Re-load selection
397     setSelection(selection);
398 }
```

Here is the call graph for this function:

8.64.3.11 void RebindWidget::on_dpiBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 518 of file rebindwidget.cpp.

References DPI_CUST_IDX, setBox(), and ui.

```
518                                                     {
519     if(index != DPI_CUST_IDX){
520         // Clear custom DPI
521         ui->dpiCustXBox->setEnabled(false);
522         ui->dpiCustYBox->setEnabled(false);
523         ui->dpiCustLabel->setEnabled(false);
524     } else {
525         ui->dpiCustXBox->setEnabled(true);
526         ui->dpiCustYBox->setEnabled(true);
527         ui->dpiCustLabel->setEnabled(true);
528     }
529     if(index == 0)
530         ui->dpiButton->setChecked(false);
531     else {
532         ui->dpiButton->setChecked(true);
533         setBox(ui->dpiBox);
534     }
535 }
```

Here is the call graph for this function:

8.64.3.12 void RebindWidget::on_dpiButton_clicked (bool *checked*) [private],[slot]

Definition at line 674 of file rebindwidget.cpp.

References ui.

```
674                                                 {
675     if(checked && ui->dpiBox->currentIndex() == 0)
676         ui->dpiBox->setCurrentIndex(1);
677 }
```

8.64.3.13 void RebindWidget::on_fnBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 464 of file rebindwidget.cpp.

References setBox(), and ui.

```
464                                                         {
465     if(index == 0)
466         ui->fnButton->setChecked(false);
467     else {
468         ui->fnButton->setChecked(true);
469         setBox(ui->fnBox);
470     }
471 }
```

Here is the call graph for this function:

8.64.3.14 void RebindWidget::on_fnButton_clicked (bool *checked*) [private],[slot]

Definition at line 611 of file rebindwidget.cpp.

References ui.

```
611                                                 {
612     if(checked && ui->fnBox->currentIndex() == 0)
613         ui->fnBox->setCurrentIndex(1);
614 }
```

8.64.3.15 void RebindWidget::on_lightBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 550 of file rebindwidget.cpp.

References setBox(), and ui.

```
550                                                         {
551     if(index == 0){
552         ui->lightButton->setChecked(false);
553         ui->lightWrapBox->hide();
554     } else {
555         ui->lightWrapBox->show();
556         ui->lightButton->setChecked(true);
557         setBox(ui->lightBox);
558     }
559 }
```

Here is the call graph for this function:

8.64.3.16 void RebindWidget::on_lightButton_clicked (bool *checked*) [private],[slot]

Definition at line 631 of file rebindwidget.cpp.

References ui.

```
631                                                 {
632     if(checked && ui->lightBox->currentIndex() == 0)
633         ui->lightBox->setCurrentIndex(1);
634 }
```

8.64.3.17 void RebindWidget::on_lockBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 561 of file rebindwidget.cpp.

References setBox(), and ui.

```
561                                                         {
562     if(index == 0)
563         ui->lockButton->setChecked(false);
564     else {
565         ui->lockButton->setChecked(true);
566         setBox(ui->lockBox);
567     }
568 }
```

Here is the call graph for this function:

8.64.3.18 void RebindWidget::on_lockButton_clicked (bool *checked*) [private],[slot]

Definition at line 636 of file rebindwidget.cpp.

References ui.

```
636                                                 {
637     if(checked && ui->lockBox->currentIndex() == 0)
638         ui->lockBox->setCurrentIndex(1);
639 }
```

8.64.3.19 void RebindWidget::on_mb2Box_currentIndexChanged (int *index*) [private],[slot]

Definition at line 500 of file rebindwidget.cpp.

References setBox(), and ui.

```
500                                                         {
501     if(index == 0)
502         ui->mb2Button->setChecked(false);
503     else {
504         ui->mb2Button->setChecked(true);
505         setBox(ui->mb2Box);
506     }
507 }
```

Here is the call graph for this function:

8.64.3.20 void RebindWidget::on_mb2Button_clicked (bool *checked*) [private],[slot]

Definition at line 664 of file rebindwidget.cpp.

References ui.

```
664                                                 {
665     if(checked && ui->mb2Box->currentIndex() == 0)
666         ui->mb2Box->setCurrentIndex(1);
667 }
```

8.64.3.21 void RebindWidget::on_mbBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 491 of file rebindwidget.cpp.

References setBox(), and ui.

```
491                                                         {
492     if(index == 0)
493         ui->mbButton->setChecked(false);
494     else {
495         ui->mbButton->setChecked(true);
496         setBox(ui->mbBox);
497     }
498 }
```

Here is the call graph for this function:

8.64.3.22 void RebindWidget::on_mbButton_clicked (bool *checked*) [private],[slot]

Definition at line 659 of file rebindwidget.cpp.

References ui.

```
659                                                 {
660     if(checked && ui->mbBox->currentIndex() == 0)
661         ui->mbBox->setCurrentIndex(1);
662 }
```

8.64.3.23 void RebindWidget::on_mediaBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 482 of file rebindwidget.cpp.

References setBox(), and ui.

```
482                                                         {
483     if(index == 0)
484         ui->mediaButton->setChecked(false);
485     else {
486         ui->mediaButton->setChecked(true);
487         setBox(ui->mediaBox);
488     }
489 }
```

Here is the call graph for this function:

8.64.3.24 void RebindWidget::on_mediaButton_clicked (bool *checked*) [private],[slot]

Definition at line 621 of file rebindwidget.cpp.

References ui.

```
621                                                 {
622     if(checked && ui->mediaBox->currentIndex() == 0)
623         ui->mediaBox->setCurrentIndex(1);
624 }
```

8.64.3.25 void RebindWidget::on_modBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 455 of file rebindwidget.cpp.

References setBox(), and ui.

```
455                                                     {
456     if(index == 0)
457         ui->modButton->setChecked(false);
458     else {
459         ui->modButton->setChecked(true);
460         setBox(ui->modBox);
461     }
462 }
```

Here is the call graph for this function:

8.64.3.26 void RebindWidget::on_modButton_clicked (bool *checked*) [private],[slot]

Definition at line 606 of file rebindwidget.cpp.

References ui.

```
606                                                 {
607     if(checked && ui->modBox->currentIndex() == 0)
608         ui->modBox->setCurrentIndex(1);
609 }
```

8.64.3.27 void RebindWidget::on_modeBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 537 of file rebindwidget.cpp.

References setBox(), and ui.

```
537                                                     {
538     if(index == 1 || index == 2)
539         ui->modeWrapBox->show();
540     else
541         ui->modeWrapBox->hide();
542     if(index == 0)
543         ui->modeButton->setChecked(false);
544     else {
545         ui->modeButton->setChecked(true);
546         setBox(ui->modeBox);
547     }
548 }
```

Here is the call graph for this function:

8.64.3.28 void RebindWidget::on_modeButton_clicked (bool *checked*) [private],[slot]

Definition at line 626 of file rebindwidget.cpp.

References ui.

```
626                                                 {
627     if(checked && ui->modeBox->currentIndex() == 0)
628         ui->modeBox->setCurrentIndex(1);
629 }
```

8.64.3.29 void RebindWidget::on_numBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 473 of file rebindwidget.cpp.

References setBox(), and ui.

```
473                                                         {
474     if(index == 0)
475         ui->numButton->setChecked(false);
476     else {
477         ui->numButton->setChecked(true);
478         setBox(ui->numBox);
479     }
480 }
```

Here is the call graph for this function:

8.64.3.30 void RebindWidget::on_numButton_clicked (bool *checked*) [private],[slot]

Definition at line 616 of file rebindwidget.cpp.

References ui.

```
616                                                 {
617     if(checked && ui->numBox->currentIndex() == 0)
618         ui->numBox->setCurrentIndex(1);
619 }
```

8.64.3.31 void RebindWidget::on_programKpBox_textChanged (const QString & *arg1*) [private],[slot]

Definition at line 570 of file rebindwidget.cpp.

References setBox(), and ui.

```
570                                                             {
571     if(arg1.isEmpty()){
572         ui->programKpButton->setChecked(false);
573         ui->programKpExtra->hide();
574     } else {
575         ui->programKpButton->setChecked(true);
576         setBox(ui->programKpBox);
577         ui->programKpExtra->show();
578     }
579 }
```

Here is the call graph for this function:

8.64.3.32 void RebindWidget::on_programKpButton_clicked (bool *checked*) [private],[slot]

Definition at line 641 of file rebindwidget.cpp.

References setBox(), and ui.

```
641                                                     {
642     if(!checked){
643         ui->programKpBox->setText("");
644     } else if(ui->programKpBox->text().isEmpty()){
645         ui->programKpBox->setFocus();
646         setBox(ui->programKpBox);
647     }
648 }
```

Here is the call graph for this function:

8.64.3.33 void RebindWidget::on_programKpSlBox_clicked (bool *checked*) [private],[slot]

Definition at line 679 of file rebindwidget.cpp.

References ui.

```
679                                                     {
680     ui->programKpModeBox->setCurrentIndex(0);
681     ui->programKpModeBox->setEnabled(checked);
682 }
```

8.64.3.34 void RebindWidget::on_programKrBox_textChanged (const QString & *arg1*) [private],[slot]

Definition at line 581 of file rebindwidget.cpp.

References setBox(), and ui.

```
581                                                         {
582     if(arg1.isEmpty()){
583         ui->programKrButton->setChecked(false);
584         ui->programKrExtra->hide();
585     } else {
586         ui->programKrButton->setChecked(true);
587         setBox(ui->programKrBox);
588         ui->programKrExtra->show();
589     }
590 }
```

Here is the call graph for this function:

8.64.3.35 void RebindWidget::on_programKrButton_clicked (bool *checked*) [private],[slot]

Definition at line 650 of file rebindwidget.cpp.

References setBox(), and ui.

```
650                                                         {
651     if(!checked){
652         ui->programKrBox->setText("");
653     } else if(ui->programKrBox->text().isEmpty()){
654         ui->programKrBox->setFocus();
655         setBox(ui->programKrBox);
656     }
657 }
```

Here is the call graph for this function:

8.64.3.36 void RebindWidget::on_programKrSlBox_clicked (bool *checked*) [private],[slot]

Definition at line 684 of file rebindwidget.cpp.

References ui.

```
684                                                         {
685     ui->programKrModeBox->setCurrentIndex(0);
686     ui->programKrModeBox->setEnabled(checked);
687 }
```

8.64.3.37 void RebindWidget::on_pteMacroBox_textChanged ( ) [private],[slot]

Definition at line 449 of file rebindwidget.cpp.

References setBox(), and ui.

```
449                                                 {
450     if (ui->pteMacroBox->toPlainText().length() > 0) {
451         setBox(ui->pteMacroBox);
452     }
453 }
```

Here is the call graph for this function:

8.64.3.38 void RebindWidget::on_resetButton_clicked ( ) [private],[slot]

Definition at line 400 of file rebindwidget.cpp.

References bind, KbBind::resetAction(), and selection.

```
400                                         {
401     bind->resetAction(selection);
402 }
```

Here is the call graph for this function:

8.64.3.39 void RebindWidget::on_typingBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 440 of file rebindwidget.cpp.

References setBox(), and ui.

```
440                                                         {
441     if(index == 0)
442         ui->typingButton->setChecked(false);
443     else {
444         ui->typingButton->setChecked(true);
445         setBox(ui->typingBox);
446     }
447 }
```

Here is the call graph for this function:

8.64.3.40 void RebindWidget::on_typingButton_clicked (bool *checked*) [private],[slot]

Definition at line 601 of file rebindwidget.cpp.

References ui.

```
601                                                 {
602     if(checked && ui->typingBox->currentIndex() == 0)
603         ui->typingBox->setCurrentIndex(1);
604 }
```

8.64.3.41 void RebindWidget::on_unbindButton_clicked ( ) [private],[slot]

Definition at line 404 of file rebindwidget.cpp.

References bind, KbBind::noAction(), and selection.

```
404                                         {
405     bind->noAction(selection);
406 }
```

Here is the call graph for this function:

8.64.3.42 void RebindWidget::on_wheelBox_currentIndexChanged (int *index*) [private],[slot]

Definition at line 509 of file rebindwidget.cpp.

References setBox(), and ui.

```
509                                                     {
510     if(index == 0)
511         ui->wheelButton->setChecked(false);
512     else {
513         ui->wheelButton->setChecked(true);
514         setBox(ui->wheelBox);
515     }
516 }
```

Here is the call graph for this function:

8.64.3.43 void RebindWidget::on_wheelButton_clicked (bool *checked*) [private],[slot]

Definition at line 669 of file rebindwidget.cpp.

References ui.

```
669                                                 {
670     if(checked && ui->wheelBox->currentIndex() == 0)
671         ui->wheelBox->setCurrentIndex(1);
672 }
```

8.64.3.44 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, fnKeys, Key::friendlyName(), KeyMap::isISO(), KbBind::isMouse(), KeyAction::isNormal(), KeyMap::K95, KeyMap::layout(), KbBind::map(), mediaKeys, KbProfile::modes(), modKeys, KbMode::name(), numKeys, KbBind::perf(), profile, setSelection(), typingKeys, and ui.

```
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(KbBind::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 + 3;
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 {
```

```
        ui->dpiButton->setEnabled(false);
        ui->dpiBox->setEnabled(false);
        ui->dpiWarning->show();
    }
    // Always disable custom DPI boxes until selected
    ui->dpiCustXBox->setEnabled(false);
    ui->dpiCustYBox->setEnabled(false);
    ui->dpiCustLabel->setEnabled(false);
}
```

Here is the call graph for this function:

8.64.3.45 void RebindWidget::setBox (QWidget * box) [private]

Definition at line 408 of file rebindwidget.cpp.

References helpStatus(), and ui.

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().

```
                                    {
    // Un-select every item except for the current one.
    // on_*_currentIndexChanged will take care of deselecting the checkbox.
    // Key
    if(box != ui->typingBox) ui->typingBox->setCurrentIndex(0);
    if(box != ui->modBox) ui->modBox->setCurrentIndex(0);
    if(box != ui->fnBox) ui->fnBox->setCurrentIndex(0);
    if(box != ui->numBox) ui->numBox->setCurrentIndex(0);
    if(box != ui->mediaBox) ui->mediaBox->setCurrentIndex(0);
    // Mouse
    if(box != ui->mbBox) ui->mbBox->setCurrentIndex(0);
    if(box != ui->mb2Box) ui->mb2Box->setCurrentIndex(0);
    if(box != ui->wheelBox) ui->wheelBox->setCurrentIndex(0);
    if(box != ui->dpiBox) ui->dpiBox->setCurrentIndex(0);
    // Anim
    if(box != ui->animBox) ui->animBox->setCurrentIndex(0);
    // Special
    if(box != ui->modeBox) ui->modeBox->setCurrentIndex(0);
    if(box != ui->lightBox) ui->lightBox->setCurrentIndex(0);
    if(box != ui->lockBox) ui->lockBox->setCurrentIndex(0);
    // Program
    if(box != ui->programKpBox && box != ui->programKrBox){
        ui->programKpButton->setChecked(false);
        ui->programKrButton->setChecked(false);
    }
    // Clear macro panel
    if (box != ui->pteMacroBox) {
        ui->pteMacroBox->setPlainText("");
        helpStatus(1);
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.64.3.46 void RebindWidget::setSelection (const QStringList & *newSelection,* bool *applyPrevious* = `false`)

Definition at line 128 of file rebindwidget.cpp.

References KbBind::action(), KeyAction::animInfo(), KbLight::animList(), applyChanges(), bind, KeyAction::DPI_CUSTOM, DPI_OFFSET, KeyAction::dpiInfo(), KbLight::findAnimIdx(), fnKeys, KeyAction::isAnim(), KeyAction::isDPI(), KeyAction::isMouse(), KeyAction::isNormal(), KeyAction::isProgram(), KeyAction::isSpecial(), KeyAction::is-

ValidMacro(), KbBind::light(), KeyAction::LIGHT_DOWN, KeyAction::LIGHT_DOWN_WRAP, KeyAction::LIGHT_UP, KeyAction::LIGHT_UP_WRAP, KeyAction::macroContent(), KeyAction::macroDisplay(), KeyAction::macroLine(), mediaKeys, KeyAction::MODE_NEXT, KeyAction::MODE_NEXT_WRAP, KeyAction::MODE_PREV, KeyAction::MODE_PREV_WRAP, modKeys, mouseExtKeys, mouseKeys, KbAnim::name(), numKeys, KeyAction::PROGRAM_PR_MULTI, KeyAction::PROGRAM_RE_INDEF, KeyAction::PROGRAM_RE_KPSTOP, KeyAction::PROGRAM_RE_MULTI, KeyAction::programInfo(), selection, setBox(), KeyAction::specialInfo(), TAB_ANIM, TAB_KB, TAB_MACRO, TAB_MOUSE, TAB_PROGRAM, TAB_SPECIAL, typingKeys, ui, and wheelKeys.

Referenced by on_cancelButton_clicked(), and setBind().

```
{
    if(bind){
        // Populate animation list
        // FIXME: There should be a trigger to update this whenever an animation is added/removed, even if the key selection hasn't changed
        ui->animBox->clear();
        ui->animBox->addItem(" ");
        foreach(KbAnim* anim, bind->light()->animList())
            ui->animBox->addItem(anim->name());
    }

    // Apply changes to previous selection (if any)
    if(!selection.isEmpty() && applyPrevious){
        QStringList previous = selection;
        selection = QStringList();
        applyChanges(previous, false);
    }

    selection = newSelection;
    if(newSelection.isEmpty()){
        hide();
        return;
    } else
        show();

    bool hasAction = false;
    QString action;
    if(selection.count() == 1){
        // Select action from the UI
        action = bind->action(selection.first());
        if(!action.isEmpty())
            hasAction = true;
    } else {
        // Select an action from the UI if all keys are the same.
        // Select nothing if any are unbound or different
        foreach(const QString& key, selection){
            QString keyAction = bind->action(key);
            if(keyAction.isEmpty() || (hasAction && keyAction != action)){
                hasAction = false;
                break;
            } else if(action.isEmpty()){
                action = keyAction;
                hasAction = true;
            }
        }
    }

    if(!hasAction)
        action = "";
    KeyAction act(action);
    // Clear everything
    setBox(0);
    ui->dpiCustXBox->setValue(400);
    ui->dpiCustYBox->setValue(400);
    ui->animOnceBox->setChecked(false);
    ui->animKrBox->setChecked(false);
    ui->programKpBox->setText("");
    ui->programKrBox->setText("");
    ui->programKpSIBox->setChecked(false);
    ui->programKrSIBox->setChecked(false);
    ui->programKpModeBox->setCurrentIndex(0);
    ui->programKrModeBox->setCurrentIndex(0);
    ui->programKpModeBox->setEnabled(false);
    ui->programKrModeBox->setEnabled(false);
    // Clear neu UI elements in MacroTab
    ui->pteMacroBox->setPlainText("");
    ui->pteMacroText->setPlainText("");
    ui->pteMacroComment->setPlainText("");
    // Fill in field and select tab according to action type
    bool mouse = act.isMouse();
    if(mouse){
        // Mouse buttons
        ui->tabWidget->setCurrentIndex(TAB_MOUSE);
        // Set mouse buttons (indexOf returns -1 if not found, index zero is blank)
```

```
        ui->mbBox->setCurrentIndex(mouseKeys.indexOf(action) + 1);
        ui->mb2Box->setCurrentIndex(mouseExtKeys.indexOf(action) + 1);
        ui->wheelBox->setCurrentIndex(wheelKeys.indexOf(action) + 1);
        if(act.isDPI()){
            QPoint custom;
            int value = act.dpiInfo(custom);
            if(value <= -DPI_OFFSET || value > KeyAction::DPI_CUSTOM)
                return;
            ui->dpiBox->setCurrentIndex(value + DPI_OFFSET);
            if(value == KeyAction::DPI_CUSTOM){
                ui->dpiCustXBox->setValue(custom.x());
                ui->dpiCustYBox->setValue(custom.y());
            }
        }
    } else if(act.isNormal()){
        // Standard key
        ui->tabWidget->setCurrentIndex(TAB_KB);
        ui->typingBox->setCurrentIndex(typingKeys.indexOf(action) + 1);
        ui->modBox->setCurrentIndex(modKeys.indexOf(action) + 1);
        ui->fnBox->setCurrentIndex(fnKeys.indexOf(action) + 1);
        ui->numBox->setCurrentIndex(numKeys.indexOf(action) + 1);
        ui->mediaBox->setCurrentIndex(mediaKeys.indexOf(action) + 1);
    } else if(act.isProgram()){
        // Program key
        ui->tabWidget->setCurrentIndex(TAB_PROGRAM);
        QString onPress, onRelease;
        int stop = act.programInfo(onPress, onRelease);
        ui->programKpBox->setText(onPress);
        ui->programKrBox->setText(onRelease);
        switch(stop & 0x0F){
        case KeyAction::PROGRAM_PR_MULTI:
            ui->programKpSIBox->setChecked(false);
            ui->programKpModeBox->setCurrentIndex(0);
            ui->programKpModeBox->setEnabled(false);
            break;
        default:
            ui->programKpSIBox->setChecked(true);
            ui->programKpModeBox->setCurrentIndex(stop & 0x0F);
            ui->programKpModeBox->setEnabled(true);
            break;
        }
        switch(stop & 0xF0){
        case KeyAction::PROGRAM_RE_MULTI:
            ui->programKrSIBox->setChecked(false);
            ui->programKrModeBox->setCurrentIndex(0);
            ui->programKrModeBox->setEnabled(false);
            break;
        case KeyAction::PROGRAM_RE_INDEF:
            ui->programKrSIBox->setChecked(true);
            ui->programKrModeBox->setCurrentIndex(0);
            ui->programKrModeBox->setEnabled(true);
            break;
        case KeyAction::PROGRAM_RE_KPSTOP:
            ui->programKrSIBox->setChecked(true);
            ui->programKrModeBox->setCurrentIndex(1);
            ui->programKrModeBox->setEnabled(true);
            break;
        }
    } else if(act.isAnim()){
        // Animation key
        ui->tabWidget->setCurrentIndex(TAB_ANIM);
        bool onlyOnce = false, stopOnRelease = false;
        QUuid id = act.animInfo(onlyOnce, stopOnRelease);
        // Find this animation in the list. If not found, -1 will be returned, resulting in the blank space being selected
        int index = bind->light()->findAnimIdx(id);
        ui->animBox->setCurrentIndex(index + 1);
        ui->animOnceBox->setChecked(!onlyOnce);
        ui->animKrBox->setChecked(stopOnRelease);
    } else if(act.isSpecial()){
        // Other special keys
        ui->tabWidget->setCurrentIndex(TAB_SPECIAL);
        int param;
        QString sAction = act.specialInfo(param);
        // Mode selection. Check wrap-around flag
        ui->modeWrapBox->setChecked(true);
        if(sAction == "mode"){
            ui->modeWrapBox->setChecked(true);
            if(param == KeyAction::MODE_PREV_WRAP)
                param = KeyAction::MODE_PREV;
            else if(param == KeyAction::MODE_NEXT_WRAP)
                param = KeyAction::MODE_NEXT;
            else if(param < 0)
                ui->modeWrapBox->setChecked(false);
            // Set mode box to current selection, or to mode 1 if invalid
            param += 3;
            if(param >= 0 && param < ui->modeBox->count())
```

```
                ui->modeBox->setCurrentIndex(param);
            else
                // 0 -> "", 1 -> Prev, 2 -> Next, 3 -> Mode 1
                ui->modeBox->setCurrentIndex(3);
        } else if (sAction == "macro") {
            ui->tabWidget->setCurrentIndex(TAB_MACRO);
            if (act.isValidMacro()) {
                ui->pteMacroBox->setPlainText(act.macroContent());
                ui->pteMacroText->setPlainText(act.macroLine()[1].replace("&das_IST_31N_col0n;", ":"));
                ui->pteMacroComment->setPlainText(act.macroLine()[2].replace("&das_IST_31N_col0n;", ":"))
    ;
            } else {
                qDebug("RebindWidget::setSelection found invalid macro definition.");
                act.macroDisplay();
            }
        } else
            ui->modeBox->setCurrentIndex(0);
        // Brightness control. Also check wrap
        if(sAction == "light"){
            ui->lightWrapBox->setChecked(true);
            if(param == KeyAction::LIGHT_DOWN_WRAP)
                param = KeyAction::LIGHT_DOWN;
            else if(param == KeyAction::LIGHT_UP_WRAP)
                param = KeyAction::LIGHT_UP;
            else
                ui->lightWrapBox->setChecked(false);
            if(param < 0 || param > 1)
                param = -1;
            ui->lightBox->setCurrentIndex(param + 1);
        } else
            ui->lightBox->setCurrentIndex(0);
        // Win lock
        if(sAction == "lock"){
            if(param < 0 || param > 2)
                param = -1;
            ui->lockBox->setCurrentIndex(param + 1);
        } else
            ui->lockBox->setCurrentIndex(0);
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

8.64.4 Field Documentation

8.64.4.1 KbBind* RebindWidget::bind [private]

Definition at line 86 of file rebindwidget.h.

Referenced by applyChanges(), on_btnStartMacro_clicked(), on_btnStopMacro_clicked(), on_resetButton_clicked(), on_unbindButton_clicked(), setBind(), and setSelection().

8.64.4.2 QStringList RebindWidget::fnKeys [private]

Definition at line 92 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

8.64.4.3 MacroReader* RebindWidget::macReader [private]

Definition at line 98 of file rebindwidget.h.

Referenced by on_btnStartMacro_clicked(), and on_btnStopMacro_clicked().

8.64.4.4 QStringList RebindWidget::mediaKeys [private]

Definition at line 94 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

8.64.4.5 QStringList RebindWidget::modKeys [private]

Definition at line 91 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

8.64.4.6 QStringList RebindWidget::mouseExtKeys [private]

Definition at line 96 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), and setSelection().

8.64.4.7 QStringList RebindWidget::mouseKeys [private]

Definition at line 95 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), and setSelection().

8.64.4.8 QStringList RebindWidget::numKeys [private]

Definition at line 93 of file rebindwidget.h.

Referenced by applyChanges(), RebindWidget(), setBind(), and setSelection().

8.64.4.9 KbProfile* RebindWidget::profile [private]

Definition at line 87 of file rebindwidget.h.

Referenced by setBind().

8.64.4.10 QStringList RebindWidget::selection [private]

Definition at line 88 of file rebindwidget.h.

Referenced by on_applyButton_clicked(), on_cancelButton_clicked(), on_resetButton_clicked(), on_unbindButton_clicked(), and setSelection().

8.64.4.11 const int RebindWidget::TAB_ANIM = 2 [static], [private]

Definition at line 77 of file rebindwidget.h.

Referenced by setSelection().

8.64.4.12 const int RebindWidget::TAB_KB = 0 [static], [private]

Definition at line 77 of file rebindwidget.h.

Referenced by setSelection().

8.64.4.13 const int RebindWidget::TAB_MACRO = 5 [static], [private]

Definition at line 77 of file rebindwidget.h.

Referenced by setSelection().

8.64.4.14 const int RebindWidget::TAB_MOUSE = 1 [static], [private]

Definition at line 77 of file rebindwidget.h.

Referenced by setSelection().

8.64.4.15 const int RebindWidget::TAB_PROGRAM = 4 [static], [private]

Definition at line 77 of file rebindwidget.h.

Referenced by setSelection().

8.64.4.16 const int RebindWidget::TAB_SPECIAL = 3 [static], [private]

Definition at line 77 of file rebindwidget.h.

Referenced by setSelection().

8.64.4.17 QStringList RebindWidget::typingKeys [private]

Definition at line 90 of file rebindwidget.h.

Referenced by applyChanges(), setBind(), and setSelection().

8.64.4.18 Ui::RebindWidget* RebindWidget::ui [private]

Definition at line 74 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(), setSelection(), and ~RebindWidget().

8.64.4.19 QStringList RebindWidget::wheelKeys [private]

Definition at line 97 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

8.65 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

8.65.1 Detailed Description

Definition at line 9 of file rlistwidget.h.

8.65.2 Constructor & Destructor Documentation

8.65.2.1 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:

8.65.3 Member Function Documentation

8.65.3.1 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:

8.65.3.2 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:

8.65.3.3 void RListWidget::orderChanged ( ) [signal]

Referenced by timerTick().

Here is the caller graph for this function:

8.65.3.4 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:

8.65.3.5 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:

8.65.4 Field Documentation

8.65.4.1 QVariant RListWidget::currentData [private]

Definition at line 24 of file rlistwidget.h.

8.65.4.2 const int RListWidget::DATA_ROLE = Qt::UserRole + 100 [static], [private]

Definition at line 29 of file rlistwidget.h.

Referenced by enter(), rescanItems(), and timerTick().

8.65.4.3 QVariant RListWidget::dragged [private]

Definition at line 26 of file rlistwidget.h.

Referenced by enter(), and timerTick().

8.65.4.4 QList<QVariant> RListWidget::previousItems [private]

Definition at line 25 of file rlistwidget.h.

Referenced by rescanItems(), and timerTick().

8.65.4.5 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/rlistwidget.cpp

8.66 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

8.66.1 Detailed Description

Definition at line 11 of file settingswidget.h.

8.66.2 Constructor & Destructor Documentation

8.66.2.1 SettingsWidget::SettingsWidget (QWidget * *parent* = 0) [explicit]

Definition at line 15 of file settingswidget.cpp.

References AutoRun::available(), AutoRun::enable(), extra, KeyMap::getLayout(), KbBind::globalRemap(), AutoRun::isEnabled(), Kb::layout(), KeyMap::layoutNames(), KbBind::loadGlobalRemap(), KeyMap::locale(), modKeys, modNames, KeyMap::NO_LAYOUT, AutoRun::once(), CkbSettings::remove(), CkbSettings::setValue(), showLayoutDialog(), ui, updateModifiers(), and CkbSettings::value().

```
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
```

```
    KbBind::loadGlobalRemap();
    if(modKeys.isEmpty()){
        modKeys << "caps" << "lshift" << "lctrl" << "lalt" << "lwin";
#ifdef Q_OS_MACX
        modNames << "Caps Lock" << "Shift" << "Control ()" << "Option ()" << "Command ()";
#else
        modNames << "Caps Lock" << "Shift" << "Control" << "Alt" << "Super";
#endif
    }
    foreach(const QString& name, modNames){
        ui->capsBox->addItem(name);
        ui->shiftBox->addItem(name);
        ui->ctrlBox->addItem(name);
        ui->altBox->addItem(name);
        ui->winBox->addItem(name);
    }

    int idx = modKeys.indexOf(KbBind::globalRemap("caps"));
    if(idx >= 0)
        ui->capsBox->setCurrentIndex(idx);
    idx = modKeys.indexOf(KbBind::globalRemap("lshift"));
    if(idx >= 0)
        ui->shiftBox->setCurrentIndex(idx);
    idx = modKeys.indexOf(KbBind::globalRemap("lctrl"));
    if(idx >= 0)
        ui->ctrlBox->setCurrentIndex(idx);
    idx = modKeys.indexOf(KbBind::globalRemap("lwin"));
    if(idx >= 0)
        ui->winBox->setCurrentIndex(idx);
    idx = modKeys.indexOf(KbBind::globalRemap("lalt"));
    if(idx >= 0)
        ui->altBox->setCurrentIndex(idx);
#ifdef Q_OS_MACX
    if(settings.value("osxCmdSwap").toBool()){
        // ckb <0.0.41
        settings.remove("osxCmdSwap");
        ui->ctrlBox->setCurrentIndex(modKeys.indexOf("lwin"));
        ui->winBox->setCurrentIndex(modKeys.indexOf("lctrl"));
        updateModifiers();
    }
#endif

    // Read auto update settings
    ui->autoFWBox->setChecked(!settings.value("DisableAutoFWCheck").toBool());

    // Read auto run settings
    if(!AutoRun::available())
        ui->loginItemBox->hide();
    else {
        if(!AutoRun::once())
            // If this is the first time running the app, enable auto run by default
            AutoRun::enable();
        ui->loginItemBox->setChecked(AutoRun::isEnabled());
    }

    // Prepare extra settings
    extra = new ExtraSettingsWidget(this);
}
```

Here is the call graph for this function:

8.66.2.2 SettingsWidget::~SettingsWidget ( )

Definition at line 94 of file settingswidget.cpp.

References ui.

```
94                                  {
95     delete ui;
96 }
```

8.66.3 Member Function Documentation

8.66.3.1 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:

8.66.3.2 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:

8.66.3.3 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:

8.66.3.4 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:

8.66.3.5 void SettingsWidget::on_extraButton_clicked ( ) [private],[slot]

Definition at line 180 of file settingswidget.cpp.

References extra.

```
180                                                 {
181     extra->exec();
182 }
```

8.66.3.6 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:

8.66.3.7 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:

8.66.3.8 void SettingsWidget::on_pushButton_clicked ( ) [private],[slot]

Definition at line 139 of file settingswidget.cpp.

```
139                                         {
140     qApp->quit();
141 }
```

8.66.3.9 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:

8.66.3.10 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:

8.66.3.11 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:

8.66.3.12 void SettingsWidget::setStatus (const QString & *text*)

Definition at line 110 of file settingswidget.cpp.

References ui.

Referenced by MainWindow::updateVersion().

```
110                                         {
111     ui->devicesLabel->setText(text);
112 }
```

Here is the caller graph for this function:

8.66.3.13 void SettingsWidget::setVersion (const QString & *version*)

Definition at line 114 of file settingswidget.cpp.

References ui.

Referenced by MainWindow::MainWindow().

```
114                                                 {
115     ui->versionLabel->setText(version);
116 }
```

Here is the caller graph for this function:

8.66.3.14 void SettingsWidget::showLayoutDialog ( ) [private], [slot]

Definition at line 98 of file settingswidget.cpp.

References 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:

8.66.3.15 void SettingsWidget::updateModifiers ( ) [private]

Definition at line 124 of file settingswidget.cpp.

References modKeys, right(), KbBind::saveGlobalRemap(), KbBind::setGlobalRemap(), and ui.

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:

8.66.4 Friends And Related Function Documentation

8.66.4.1 friend class MainWindow [friend]

Definition at line 44 of file settingswidget.h.

8.66.5 Field Documentation

8.66.5.1 ExtraSettingsWidget* SettingsWidget::extra [private]

Definition at line 46 of file settingswidget.h.

Referenced by on_extraButton_clicked(), pollUpdates(), and SettingsWidget().

8.66.5.2 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

8.67 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

8.67.1 Detailed Description

Definition at line 57 of file ckbsettings.h.

8.67.2 Constructor & Destructor Documentation

8.67.2.1 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:

8.67.2.2 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:

8.67.3 Field Documentation

8.67.3.1 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

8.68 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

8.68.1 Detailed Description

Definition at line 11 of file kbmode.h.

8.68.2 Constructor & Destructor Documentation

8.68.2.1 UsbId::UsbId (const QString & *_guid,* quint32 *_modified*) [inline]

Definition at line 16 of file kbmode.h.

```
16 : guid(_guid), modified(_modified), hwModified(_modified) {}
```

8.68.2.2 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) {}
```

8.68.2.3 UsbId::UsbId ( ) [inline]

Definition at line 18 of file kbmode.h.

```
18 : guid(QUuid::createUuid()),modified(0) {}
```

8.68.3 Member Function Documentation

8.68.3.1 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:

8.68.3.2 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:

8.68.3.3 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:

8.68.3.4 void UsbId::hwModifiedString (const QString & *newModified*) [inline]

Definition at line 25 of file kbmode.h.

References hwModified.

```
25 { hwModified = newModified.toUInt(0, 16); }
```

8.68.3.5 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:

8.68.3.6 void UsbId::modifiedString (const QString & *newModified*) [inline]

Definition at line 23 of file kbmode.h.

References modified.

```
23 { modified = newModified.toUInt(0, 16); }
```

8.68.3.7 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:

8.68.3.8 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:

8.68.4 Field Documentation

8.68.4.1 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().

8.68.4.2 quint32 UsbId::hwModified

Definition at line 14 of file kbmode.h.

Referenced by hwModifiedString(), Kb::hwSave(), KbMode::KbMode(), and KbProfile::KbProfile().

8.68.4.3 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 9

File Documentation

9.1 src/ckb/animadddialog.cpp File Reference

```
#include "animadddialog.h"
#include "ui_animadddialog.h"
```

Include dependency graph for animadddialog.cpp:

9.2 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

9.3 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.

9.3.1 Function Documentation

9.3.1.1 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:

9.3.2 Variable Documentation

9.3.2.1 const double ONE_DAY = 24. ∗ 60. ∗ 60. [static]

Definition at line 75 of file animscript.cpp.

Referenced by AnimScript::load().

9.4 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

9.4.1 Data Structure Documentation

9.4.1.1 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	

9.5 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)

9.5.1 Function Documentation

9.5.1.1 static int angleFlip (int *angle*) [inline], [static]

Definition at line 12 of file animsettingdialog.cpp.

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:

9.6 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

9.7 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"

9.7.1 Variable Documentation

9.7.1.1 const QString settingPath = "Program/DidLoginItem" [static]

Definition at line 18 of file autorun.cpp.

9.8 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

9.9 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:

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[])

9.9.1 Data Structure Documentation

9.9.1.1 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	

9.9.1.2 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	

9.9.1.3 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]	

9.9.2 Macro Definition Documentation

9.9.2.1 #define CKB_CONTAINER(*macro*) do { macro } while(0)

Definition at line 57 of file ckb-anim.h.

9.9.2.2 #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.

9.9.2.3 #define CKB_DESCRIPTION(*description*) CKB_CONTAINER(printf("description "); printurl(description); printf("\n");)

Definition at line 70 of file ckb-anim.h.

9.9.2.4 #define CKB_DOWN 1

Definition at line 52 of file ckb-anim.h.

9.9.2.5 #define CKB_GRAD_MAX 100

Definition at line 151 of file ckb-anim.h.

9.9.2.6 #define CKB_GUID(*guid*) CKB_CONTAINER(printf("guid "); printurl(guid); printf("\n");)

Definition at line 60 of file ckb-anim.h.

9.9.2.7 #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.

9.9.2.8 #define CKB_KEYNAME_MAX 12

Definition at line 130 of file ckb-anim.h.

Referenced by main().

9.9.2.9 #define CKB_KP_NAME "name"

Definition at line 91 of file ckb-anim.h.

9.9.2.10 #define CKB_KP_NONE "none"

Definition at line 90 of file ckb-anim.h.

9.9.2.11 #define CKB_KP_POSITION "position"

Definition at line 92 of file ckb-anim.h.

9.9.2.12 #define CKB_KPMODE(*mode*) CKB_CONTAINER(printf("kpmode %s\n", mode);)

Definition at line 93 of file ckb-anim.h.

9.9.2.13 #define CKB_LICENSE(*license*) CKB_CONTAINER(printf("license "); printurl(license); printf("\n");)

Definition at line 68 of file ckb-anim.h.

9.9.2.14 #define CKB_LIVEPARAMS(*enable*) CKB_CONTAINER(printf("parammode %s\n", (enable) ? "live" : "static");)

Definition at line 104 of file ckb-anim.h.

9.9.2.15 #define CKB_MAX_WORD (4 ∗ 1024)

Definition at line 238 of file ckb-anim.h.

Referenced by ckb_read_params(), and main().

9.9.2.16 #define CKB_NAME(*name*) CKB_CONTAINER(printf("name "); printurl(name); printf("\n");)

Definition at line 62 of file ckb-anim.h.

9.9.2.17 #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.

9.9.2.18 #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.

9.9.2.19 #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.

9.9.2.20 #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.

9.9.2.21 #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.

9.9.2.22 #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.

9.9.2.23 #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.

9.9.2.24 #define CKB_PARAM_LABEL(*name, text*) CKB_PARAM("label", name, text, "",)

Definition at line 83 of file ckb-anim.h.

9.9.2.25 #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.

9.9.2.26 #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.

9.9.2.27 #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.

9.9.2.28 #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.

9.9.2.29 #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.

9.9.2.30 #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.

9.9.2.31 #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.

9.9.2.32 #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.

9.9.2.33 #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.

9.9.2.34 #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.

9.9.2.35 #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.

9.9.2.36 #define CKB_PARSE_STRING(*param_name*) if(!strcmp(name, param_name))

Definition at line 123 of file ckb-anim.h.

9.9.2.37 #define CKB_PREEMPT(*enable*) CKB_CONTAINER(printf("preempt %s\n", (enable) ? "on" : "off");)

Definition at line 102 of file ckb-anim.h.

9.9.2.38 #define CKB_PRESET_END CKB_CONTAINER(printf("\n");)

Definition at line 87 of file ckb-anim.h.

9.9.2.39 #define CKB_PRESET_PARAM(*name, value*) CKB_CONTAINER(printf(" %s=", name); printurl(value);)

Definition at line 86 of file ckb-anim.h.

9.9.2.40 #define CKB_PRESET_START(*name*) CKB_CONTAINER(printf("preset "); printurl(name);)

Definition at line 85 of file ckb-anim.h.

9.9.2.41 #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.

9.9.2.42 #define CKB_REPEAT(*enable*) CKB_CONTAINER(printf("repeat %s\n", (enable) ? "on" : "off");)

Definition at line 99 of file ckb-anim.h.

9.9.2.43 #define CKB_TIME_ABSOLUTE "absolute"

Definition at line 96 of file ckb-anim.h.

9.9.2.44 #define CKB_TIME_DURATION "duration"

Definition at line 95 of file ckb-anim.h.

9.9.2.45 #define CKB_TIMEMODE(*mode*) CKB_CONTAINER(printf("time %s\n", mode);)

Definition at line 97 of file ckb-anim.h.

9.9.2.46 #define CKB_UP 0

Definition at line 53 of file ckb-anim.h.

9.9.2.47 #define CKB_VERSION(*version*) CKB_CONTAINER(printf("version "); printurl(version); printf("\n");)

Definition at line 64 of file ckb-anim.h.

9.9.2.48 #define FALSE 0

Definition at line 49 of file ckb-anim.h.

9.9.2.49 #define M_PI 3.14159265358979323846

Definition at line 42 of file ckb-anim.h.

9.9.2.50 #define TRUE 1

Definition at line 46 of file ckb-anim.h.

9.9.3 Function Documentation

9.9.3.1 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.

```
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 }
```

9.9.3.2 int ckb_frame (ckb_runctx *)

Referenced by main().

Here is the caller graph for this function:

9.9.3.3 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);
```

```
    if(res >= 2)
        urldecode(word2, word2);
    if(res >= 3)
        urldecode(word3, word3);
    line[strlen(line) - 1] = 0;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.9.3.4 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.

```
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                {
    // Find the points surrounding this position
    int count = grad->ptcount;
    if(count == 0){
        *a = *r = *g = *b = 0.f;
        return;
    }
    int i = 1;
    for(; i < count; i++){
        if(grad->pts[i] >= pos)
            break;
    }
    // Get color by linear interpolation. Premultiply the alpha value so that it returns the expected color
    // (i.e. stops with zero opacity won't contribute to color)
    float distance = grad->pts[i] - grad->pts[i - 1];
    float dx = (pos - grad->pts[i - 1]) / distance;
    float a1 = grad->a[i] / 255., a2 = grad->a[i - 1] / 255.;
    float a3 = *a = a1 * dx + a2 * (1.f - dx);
    if(a3 == 0.){
        *a = *r = *g = *b = 0.f;
        return;
    }
    *a *= 255.f;
    *r = (grad->r[i] * a1 * dx + grad->r[i - 1] * a2 * (1.f - dx)) / a3;
    *g = (grad->g[i] * a1 * dx + grad->g[i - 1] * a2 * (1.f - dx)) / a3;
    *b = (grad->b[i] * a1 * dx + grad->b[i - 1] * a2 * (1.f - dx)) / a3;
}
```

9.9.3.5 void ckb_info ( )

Referenced by main().

Here is the caller graph for this function:

9.9.3.6 void ckb_init (ckb_runctx * *context*)

Referenced by main().

Here is the caller graph for this function:

9.9.3.7 void ckb_keypress (ckb_runctx * , ckb_key * , int , int , int)

Referenced by main().

Here is the caller graph for this function:

9.9.3.8 void ckb_parameter (ckb_runctx * , const char * , const char *)

Referenced by ckb_read_params().

Here is the caller graph for this function:

9.9.3.9 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:

9.9.3.10 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 }
```

9.9.3.11 void ckb_start (ckb_runctx * , int)

Referenced by main().

Here is the caller graph for this function:

9.9.3.12 void ckb_time (ckb_runctx * , double)

Referenced by main().

Here is the caller graph for this function:

9.9.3.13 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, ckb_key::x, 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;
```

```
                }
            } while(strcmp(cmd, "end") || strcmp(param, "keymap"));
            // Run init function
            ckb_init(&ctx);
            // Skip anything else until "begin params"
            do {
                ckb_getline(cmd, param, value);
                if(!*cmd){
                    printf("Error [ckb-main]: Reached EOF looking for \"begin params\"");
                    return -2;
                }
            } while(strcmp(cmd, "begin") || strcmp(param, "params"));
            // Parse parameters
            ckb_read_params(&ctx);
            // Skip anything else until "begin run"
            do {
                ckb_getline(cmd, param, value);
                if(!*cmd){
                    printf("Error [ckb-main]: Reached EOF looking for \"begin run\"");
                    return -2;
                }
            } while(strcmp(cmd, "begin") || strcmp(param, "run"));
            // Run the main loop
            printf("begin run\n");
            fflush(stdout);
            while(1){
                ckb_getline(cmd, param, value);
                if(!*cmd || (!strcmp(cmd, "end") && !strcmp(param, "run")))
                    break;
                // Parse input
                if(!strcmp(cmd, "start"))
                    ckb_start(&ctx, 1);
                else if(!strcmp(cmd, "stop"))
                    ckb_start(&ctx, 0);
                else if(!strcmp(cmd, "begin") && !strcmp(param, "params"))
                    ckb_read_params(&ctx);
                else if(!strcmp(cmd, "key")){
                    int x, y;
                    if(sscanf(param, "%d,%d", &x, &y) == 2){
                        // Find a key with this position
                        ckb_key* key = 0;
                        for(i = 0; i < keycount; i++){
                            if(ctx.keys[i].x == x && ctx.keys[i].y == y){
                                key = ctx.keys + i;
                                break;
                            }
                        }
                        if(key)
                            ckb_keypress(&ctx, key, key->x, key->y, !strcmp(value, "down"));
                        else
                            ckb_keypress(&ctx, 0, x, y, !strcmp(value, "down"));
                    } else {
                        // Find a key with this name
                        ckb_key* key = 0;
                        for(i = 0; i < keycount; i++){
                            if(!strcmp(ctx.keys[i].name, param)){
                                key = ctx.keys + i;
                                break;
                            }
                        }
                        if(key)
                            ckb_keypress(&ctx, key, key->x, key->y, !strcmp(value, "down"));
                    }
                } else if(!strcmp(cmd, "frame")){
                    int end = ckb_frame(&ctx);
                    // Output the frame
                    printf("begin frame\n");
                    for(i = 0; i < ctx.keycount; i++){
                        ckb_key* key = ctx.keys + i;
                        printf("argb %s %02hhx%02hhx%02hhx%02hhx\n", key->name, key->
    a, key->r, key->g, key->b);
                    }
                    printf("end frame\n");
                    if(end)
                        break;
                    fflush(stdout);
                } else {
                    double delta = 0.;
                    if(!strcmp(cmd, "time") && sscanf(param, "%lf", &delta) == 1)
                        ckb_time(&ctx, delta);
                }
            }
            printf("end run\n");
            fflush(stdout);
            free(ctx.keys);
            return 0;
        }
```

```
    }
    printf("This program must be run from within ckb\n");
    return -1;
}
```

Here is the call graph for this function:

9.9.3.14 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 }
```

9.9.3.15 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:

9.10 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

9.10.1 Macro Definition Documentation

9.10.1.1 #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().

9.10.1.2 #define lockMutexCache QMutexLocker locker(&settingsCacheMutex)

Definition at line 18 of file ckbsettings.cpp.

Referenced by CkbSettings::get(), CkbSettings::set(), and CkbSettings::setValue().

9.10.1.3 #define lockMutexStatic QMutexLocker locker(&settingsMutex)

Definition at line 16 of file ckbsettings.cpp.

Referenced by globalSettings(), and CkbSettings::set().

9.10.1.4 #define lockMutexStatic2 QMutexLocker locker2(&settingsMutex)

Definition at line 17 of file ckbsettings.cpp.

Referenced by CkbSettings::get().

9.10.2 Function Documentation

9.10.2.1 QAtomicInt cacheWritesInProgress (0)

Referenced by CkbSettings::cleanUp(), and CkbSettings::isBusy().

Here is the caller graph for this function:

9.10.2.2 static QSettings* globalSettings ( ) [static]

Definition at line 20 of file ckbsettings.cpp.

References _globalSettings, and lockMutexStatic.

Referenced by CkbSettings::get(), and CkbSettings::set().

```
20                                  {
21     if(!_globalSettings){
22         lockMutexStatic;
23         if(!(volatile QSettings*)_globalSettings){   // Check again after locking mutex in
       case another thread created the object
24             // Put the settings object in a separate thread so that it won't lock up the GUI when it syncs
25             globalThread = new QThread;
26             globalThread->start();
27             _globalSettings = new QSettings;
28             _globalSettings->moveToThread(globalThread);
29         }
30     }
31     return _globalSettings;
32 }
```

Here is the caller graph for this function:

9.10.2.3 QMutex settingsCacheMutex (QMutex::Recursive)

9.10.2.4 QMutex settingsMutex (QMutex::Recursive)

9.10.3 Variable Documentation

9.10.3.1 QSettings* _globalSettings = 0 [static]

Definition at line 8 of file ckbsettings.cpp.

Referenced by CkbSettings::cleanUp(), and globalSettings().

9.10.3.2 QMap<QString, QVariant> globalCache [static]

Definition at line 12 of file ckbsettings.cpp.

Referenced by CkbSettings::get(), CkbSettings::set(), and CkbSettings::setValue().

9.10.3.3 QThread* globalThread = 0 [static]

Definition at line 9 of file ckbsettings.cpp.

Referenced by CkbSettings::cleanUp().

9.11 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

9.12 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

9.12.1 Macro Definition Documentation

9.12.1.1 #define lockMutex QMutexLocker locker(backing == _globalSettings ? &mutex : 0)

Definition at line 7 of file ckbsettingswriter.cpp.

9.12.1.2 #define lockMutexCache QMutexLocker locker(&settingsCacheMutex)

Definition at line 10 of file ckbsettingswriter.cpp.

9.12.1.3 #define lockMutexStatic QMutexLocker locker(&settingsMutex)

Definition at line 8 of file ckbsettingswriter.cpp.

Referenced by CkbSettingsWriter::run().

9.12.1.4 #define lockMutexStatic2 QMutexLocker locker2(&settingsMutex)

Definition at line 9 of file ckbsettingswriter.cpp.

9.12.2 Variable Documentation

9.12.2.1 QAtomicInt cacheWritesInProgress

Referenced by CkbSettingsWriter::CkbSettingsWriter(), and CkbSettingsWriter::~CkbSettingsWriter().

9.12.2.2 QMutex settingsCacheMutex

9.12.2.3 QMutex settingsMutex

9.13 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

9.14 src/ckb/colorbutton.cpp File Reference

```
#include "colorbutton.h"
#include <QColorDialog>
#include <QPainter>
```

Include dependency graph for colorbutton.cpp:

9.15 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

9.16 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)

9.16.1 Function Documentation

9.16.1.1 static int qs_strcmp (const void * *lhs*, const void * *rhs*) [static]

Definition at line 49 of file colormap.cpp.

Referenced by ColorMap::colorForName(), and ColorMap::init().

```
49                                                  {
50     return strcmp(*(const char**)lhs, *(const char**)rhs);
51 }
```

Here is the caller graph for this function:

9.17 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

9.17.1 Typedef Documentation

9.17.1.1 typedef QHash<QString, QRgb> QColorMap

Definition at line 9 of file colormap.h.

9.17.1.2 typedef QHashIterator<QString, QRgb> QColorMapIterator

Definition at line 10 of file colormap.h.

9.17.1.3 typedef QMutableHashIterator<QString, QRgb> QMutableColorMapIterator

Definition at line 11 of file colormap.h.

9.18 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

9.18.1 Variable Documentation

9.18.1.1 int lastSharedDimming = -2 [static]

Definition at line 8 of file extrasettingswidget.cpp.

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

9.19 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

9.20 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 KbId

Functions

- static const int DIALOG_HEIGHT_MAX (240)
- static float verifyFw (const QByteArray &blob, const QString &features)

Variables

- static KbId ids []
- static const int DIALOG_WIDTH = 420
- static const int DIALOG_HEIGHT_MIN = 200

9.20.1 Data Structure Documentation

9.20.1.1 struct KbId

Definition at line 8 of file fwupgradedialog.cpp.

Collaboration diagram for KbId:

Data Fields

const char *	feature	
short	product	
short	vendor	

9.20.2 Function Documentation

9.20.2.1 static const int DIALOG_HEIGHT_MAX (240) [static]

Referenced by FwUpgradeDialog::exec().

Here is the caller graph for this function:

9.20.2.2 static float verifyFw (const QByteArray & *blob*, const QString & *features*) [static]

Definition at line 85 of file fwupgradedialog.cpp.

References ids.

Referenced by FwUpgradeDialog::exec().

```
85                                                                  {
86     if(blob.length() < 0x0108)
87         return 0.f;
88     const char* bData = blob.data();
89     // Make sure it matches this device based on the vendor and product IDs embedded in the blob
90     bool match = false;
91     for(uint i = 0; i < sizeof(ids)/sizeof(KbId); i++){
92         if(!memcmp(&ids[i].vendor, bData + 0x102, 2) && !memcmp(&ids[i].product, bData + 0x104, 2)
93                 && features.startsWith(ids[i].feature, Qt::CaseInsensitive)){
94             match = true;
95             break;
96         }
97     }
98     if(!match)
99         return 0.f;
100     // Copy the version from the blob
101     short version;
102     memcpy(&version, bData + 0x106, 2);
103     // Un-hexify it
104     return QString::number(version, 16).toFloat() / 100.f;
105 }
```

Here is the caller graph for this function:

9.20.3 Variable Documentation

9.20.3.1 const int DIALOG_HEIGHT_MIN = 200 [static]

Definition at line 29 of file fwupgradedialog.cpp.

Referenced by FwUpgradeDialog::FwUpgradeDialog(), and FwUpgradeDialog::on_actionButton_clicked().

9.20.3.2 const int DIALOG_WIDTH = 420 [static]

Definition at line 28 of file fwupgradedialog.cpp.

Referenced by FwUpgradeDialog::exec(), FwUpgradeDialog::FwUpgradeDialog(), and FwUpgradeDialog::on_actionButton_clicked().

9.20.3.3 KbId ids[] [static]

Initial value:

```
= {

    { 0x1b1c, 0x1b17, "corsair k65 rgb" },
    { 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().

9.21 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

9.22 src/ckb/gradientbutton.cpp File Reference

```
#include <cmath>
#include <cstdio>
#include <QPainter>
#include "gradientbutton.h"
#include "gradientdialog.h"
```

Include dependency graph for gradientbutton.cpp:

9.23 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

9.24 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"

9.24.1 Variable Documentation

9.24.1.1 const QString prefsPath = "Stored Gradients" [static]

Definition at line 9 of file gradientdialog.cpp.

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

9.25 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

9.26 src/ckb/gradientdialogwidget.cpp File Reference

```
#include <cmath>
#include <QPainter>
#include <QPropertyAnimation>
#include "gradientdialogwidget.h"
```

Include dependency graph for gradientdialogwidget.cpp:

9.27 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

9.28 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

9.28.1 Variable Documentation

9.28.1.1 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().

9.28.1.2 QMutex notifyPathMutex [static]

Definition at line 13 of file kb.cpp.

Referenced by Kb::Kb(), and Kb::run().

9.28.1.3 QSet<QString> notifyPaths [static]

Definition at line 12 of file kb.cpp.

Referenced by Kb::Kb(), Kb::run(), and Kb::~Kb().

9.29 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

9.30 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 }

9.30.1 Typedef Documentation

9.30.1.1 typedef float(∗ blendFunc)(float, float)

Definition at line 362 of file kbanim.cpp.

9.30.2 Function Documentation

9.30.2.1 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 }
```

9.30.2.2 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 }
```

9.30.2.3 static float blendMultiply (float *bg*, float *fg*) [static]

Definition at line 351 of file kbanim.cpp.

```
351                                         {
352     return bg * fg;
353 }
```

9.30.2.4 static float blendNormal (float *bg*, float *fg*) [static]

Definition at line 333 of file kbanim.cpp.

```
333                                                 {
334     return fg;
335 }
```

9.30.2.5 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 }
```

9.30.3 Variable Documentation

9.30.3.1 blendFunc functions[5] = { blendNormal, blendAdd, blendSubtract, blendMultiply, blendDivide } [static]

Definition at line 363 of file kbanim.cpp.

Referenced by KbAnim::blend().

9.31 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

9.32 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:

9.33 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

9.34 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:

9.35 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

9.36 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:

9.37 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

9.38 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

9.38.1 Function Documentation

9.38.1.1 static QString tableName (const QString & *features*) [static]

Definition at line 153 of file kbfirmware.cpp.

Referenced by KbFirmware::_fileForBoard(), and KbFirmware::_latestForBoard().

```
153                                                 {
154     QStringList components = features.split(" ");
155     if(components.length() < 2)
156         return "";
157     // First two components are vendor and model
158     QString vendorModel = components[0].toUpper() + "-" + components[1].toUpper();
159     // Add "RGB" on RGB boards
160     if(features.contains("rgb") && !features.contains("monochrome"))
161         vendorModel += "RGB";
162     return vendorModel;
163 }
```

Here is the caller graph for this function:

9.38.2 Variable Documentation

9.38.2.1 const quint64 AUTO_CHECK_TIME = 60 ∗ 60 ∗ 1000 [static]

Definition at line 9 of file kbfirmware.cpp.

Referenced by KbFirmware::_checkUpdates().

9.39 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

9.40 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

9.40.1 Function Documentation

9.40.1.1 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:

9.40.1.2 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:

9.40.1.3 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:

9.40.2 Variable Documentation

9.40.2.1 int _shareDimming = -1 [static]

Definition at line 7 of file kblight.cpp.

Referenced by KbLight::dimming(), KbLight::open(), and KbLight::shareDimming().

9.40.2.2 QSet<KbLight*> activeLights [static]

Definition at line 8 of file kblight.cpp.

Referenced by KbLight::close(), KbLight::open(), KbLight::shareDimming(), and KbLight::~KbLight().

9.41 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

9.42 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:

9.43 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

9.44 src/ckb/kbmanager.cpp File Reference

```
#include "kbmanager.h"
```

Include dependency graph for kbmanager.cpp:

Variables

- QString devpath = "/dev/input/ckb%1"

9.44.1 Variable Documentation

9.44.1.1 QString devpath = "/dev/input/ckb%1"

Definition at line 4 of file kbmanager.cpp.

Referenced by KbManager::scanKeyboards().

9.45 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>"

9.45.1 Macro Definition Documentation

9.45.1.1 #define DAEMON_UNAVAILABLE_STR "<unavailable>"

Definition at line 12 of file kbmanager.h.

Referenced by KbManager::ckbDaemonVersionF(), KbManager::scanKeyboards(), and MainWindow::updateVersion().

9.46 src/ckb/kbmode.cpp File Reference

```
#include "kbmode.h"
#include "kb.h"
```

Include dependency graph for kbmode.cpp:

9.47 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

9.48 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())

9.48.1 Macro Definition Documentation

9.48.1.1 #define map_last(*map*) ((map).last())

Definition at line 8 of file kbperf.cpp.

Referenced by KbPerf::popDpi().

9.49 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

9.50 src/ckb/kbprofile.cpp File Reference

```
#include "kbprofile.h"
#include "kb.h"
```

Include dependency graph for kbprofile.cpp:

9.51 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

9.52 src/ckb/kbprofiledialog.cpp File Reference

```
#include <QMenu>
#include "kbprofiledialog.h"
#include "ui_kbprofiledialog.h"
```

Include dependency graph for kbprofiledialog.cpp:

9.53 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

9.54 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:

9.55 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

9.56 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

9.56.1 Variable Documentation

9.56.1.1 const int ANIM_KRSTOP = 0x2 [static]

Definition at line 173 of file keyaction.cpp.

Referenced by KeyAction::animAction(), and KeyAction::animInfo().

9.56.1.2 const int ANIM_ONCE = 0x1 [static]

Definition at line 173 of file keyaction.cpp.

Referenced by KeyAction::animAction(), and KeyAction::animInfo().

9.57 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

9.58 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 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 K70_X_START 38
- #define K70_WIDTH (K95_WIDTH - K70_X_START)
- #define K70_HEIGHT K95_HEIGHT
- #define K65_WIDTH 209
- #define K65_HEIGHT K70_HEIGHT
- #define K65_TOP_COUNT (sizeof(K65TopRow) / 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_SCIM (sizeof(ScimKeys) / sizeof(Key))
- #define SCIM_WIDTH M65_WIDTH
- #define SCIM_HEIGHT M65_HEIGHT
- #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 patchISO (QHash< QString, Key > &map)
- static void patchANSI (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 patchES []
- static const KeyPatch patchSE []
- static const KeyPatch patchDvorak []
- static const Key K65TopRow []
- static const Key KStrafeKeys []
- static const Key M65Keys []
- static const Key SabreKeys []
- static const Key ScimKeys []
- static QHash< QString, Key > K95BaseMap
- static QHash< QString, Key > standardMaps [KeyMap::_MODEL_MAX][KeyMap::_LAYOUT_MAX]

9.58.1 Data Structure Documentation

9.58.1.1 struct KeyPatch

Definition at line 30 of file keymap.cpp.

Collaboration diagram for KeyPatch:

Data Fields

const char *	_friendlyName	
const char *	_storageName	
const char *	name	

9.58.2 Macro Definition Documentation

9.58.2.1 #define ANSI_ENTER_H 12

Definition at line 25 of file keymap.cpp.

Referenced by patchANSI().

9.58.2.2 #define ANSI_ENTER_W 26

Definition at line 24 of file keymap.cpp.

Referenced by patchANSI().

9.58.2.3 #define ANSI_ENTER_X 199

Definition at line 22 of file keymap.cpp.

Referenced by patchANSI().

9.58.2.4 #define ANSI_ENTER_Y 51

Definition at line 23 of file keymap.cpp.

Referenced by patchANSI().

9.58.2.5 #define ANSI_LSHIFT_W 26

Definition at line 27 of file keymap.cpp.

Referenced by patchANSI().

9.58.2.6 #define ANSI_LSHIFT_X 45

Definition at line 26 of file keymap.cpp.

Referenced by patchANSI().

9.58.2.7 #define K65_HEIGHT K70_HEIGHT

Definition at line 143 of file keymap.cpp.

9.58.2.8 #define K65_TOP_COUNT (sizeof(K65TopRow) / sizeof(Key))

Definition at line 148 of file keymap.cpp.

Referenced by getMap().

9.58.2.9 #define K65_WIDTH 209

Definition at line 142 of file keymap.cpp.

Referenced by getMap(), and KeyMap::modelWidth().

9.58.2.10 #define K70_HEIGHT K95_HEIGHT

Definition at line 139 of file keymap.cpp.

9.58.2.11 #define K70_WIDTH (K95_WIDTH - K70_X_START)

Definition at line 138 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

9.58.2.12 #define K70_X_START 38

Definition at line 137 of file keymap.cpp.

Referenced by getMap().

9.58.2.13 #define K95_HEIGHT 76

Definition at line 134 of file keymap.cpp.

Referenced by KeyMap::modelHeight().

9.58.2.14 #define K95_WIDTH 298

Definition at line 133 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

9.58.2.15 #define KEYCOUNT_K95 (sizeof(K95Keys) / sizeof(Key))

Definition at line 19 of file keymap.cpp.

Referenced by getMap().

9.58.2.16 #define KEYCOUNT_M65 (sizeof(M65Keys) / sizeof(Key))

Definition at line 170 of file keymap.cpp.

Referenced by getMap().

9.58.2.17 #define KEYCOUNT_SABRE (sizeof(SabreKeys) / sizeof(Key))

Definition at line 183 of file keymap.cpp.

Referenced by getMap().

9.58.2.18 #define KEYCOUNT_SCIM (sizeof(ScimKeys) / sizeof(Key))

Definition at line 200 of file keymap.cpp.

Referenced by getMap().

9.58.2.19 #define KSTRAFE_HEIGHT K95_HEIGHT

Definition at line 153 of file keymap.cpp.

9.58.2.20 #define KSTRAFE_WIDTH (K70_WIDTH + (KSTRAFE_X_START * 2))

Definition at line 152 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

9.58.2.21 #define KSTRAFE_X_START 12

Definition at line 151 of file keymap.cpp.

Referenced by getMap().

9.58.2.22 #define M65_HEIGHT 67

Definition at line 173 of file keymap.cpp.

Referenced by KeyMap::modelHeight().

9.58.2.23 #define M65_WIDTH 52

Definition at line 172 of file keymap.cpp.

Referenced by KeyMap::modelWidth().

9.58.2.24 #define N_LAYOUTS KeyMap::_LAYOUT_MAX

Definition at line 207 of file keymap.cpp.

Referenced by getMap().

9.58.2.25 #define N_MODELS KeyMap::_MODEL_MAX

Definition at line 206 of file keymap.cpp.

Referenced by getMap().

9.58.2.26 #define NS 12, 12

Definition at line 6 of file keymap.cpp.

9.58.2.27 #define patch(*map, patches*) _patch(map, patches, PATCH_COUNT(patches))

Definition at line 106 of file keymap.cpp.

Referenced by getMap().

9.58.2.28 #define PATCH_COUNT(patch) (sizeof(patch) / sizeof(KeyPatch))

Definition at line 105 of file keymap.cpp.

9.58.2.29 #define SABRE_HEIGHT M65_HEIGHT

Definition at line 186 of file keymap.cpp.

9.58.2.30 #define SABRE_WIDTH M65_WIDTH

Definition at line 185 of file keymap.cpp.

9.58.2.31 #define SCIM_HEIGHT M65_HEIGHT

Definition at line 203 of file keymap.cpp.

9.58.2.32 #define SCIM_WIDTH M65_WIDTH

Definition at line 202 of file keymap.cpp.

9.58.3 Function Documentation

9.58.3.1 static void _patch (QHash< QString, Key > & *map,* const KeyPatch * *patches,* int *patchCount*) [static]

Definition at line 107 of file keymap.cpp.

References Key::_friendlyName, and Key::_storageName.

```
107                                                                                 {
108     for(const KeyPatch* p = patches; p < patches + patchCount; p++){
109         Key& key = map[p->name];
110         key._storageName = p->_storageName;
```

```
        key._friendlyName = p->_friendlyName;
    }
}
```

9.58.3.2 static QHash<QString, Key> getMap (KeyMap::Model *model*, KeyMap::Layout *layout*) [static]

Definition at line 210 of file keymap.cpp.

References KeyMap::DE, KeyMap::DK, KeyMap::ES, KeyMap::EU, KeyMap::EU_DVORAK, KeyMap::FR, KeyMap::GB_DVORAK, Key::height, KeyMap::isISO(), KeyMap::IT, KeyMap::K65, K65_TOP_COUNT, K65_WIDTH, KeyMap::K70, K70_X_START, KeyMap::K95, K95BaseMap, KEYCOUNT_K95, KEYCOUNT_M65, KEYCOUNT_SABRE, KEYCOUNT_SCIM, KSTRAFE_X_START, KeyMap::M65, KeyMap::MX, N_LAYOUTS, N_MODELS, KeyMap::NO, patch, patchANSI(), patchISO(), KeyMap::SABRE, KeyMap::SCIMITAR, KeyMap::SE, standardMaps, KeyMap::STRAFE, KeyMap::US_DVORAK, Key::width, Key::x, and Key::y.

```
210                                                                                 {
211     if(model < 0 || layout < 0 || model >= N_MODELS || layout >=
    N_LAYOUTS)
212         return QHash<QString, Key>();
213     // Return the map if it's already filled out
214     QHash<QString, Key>& map = standardMaps[model][layout];
215     if(!map.empty())
216         return map;
217     // Otherwise, create it
218     switch(model){
219     case KeyMap::K95:{
220         // The K95 maps serve as bases for all the other keyboards
221         // Fetch the master map, or create it if not yet done
222         if(K95BaseMap.empty()){
223             for(const Key* key = K95Keys; key < K95Keys +
    KEYCOUNT_K95; key++)
224                 K95BaseMap[key->name] = *key;
225         }
226         map = K95BaseMap;
227         // Patch the map for the layout
228         switch(layout){
229         case KeyMap::DK:
230             patch(map, patchDK);
231             break;
232         case KeyMap::EU_DVORAK:
233             patch(map, patchDvorak);    // fall through
234         case KeyMap::EU:
235             patch(map, patchEU);
236             break;
237         case KeyMap::GB_DVORAK:
238         case KeyMap::US_DVORAK:
239             patch(map, patchDvorak);
240             break;
241         case KeyMap::FR:
242             patch(map, patchFR);
243             break;
244         case KeyMap::DE:
245             patch(map, patchDE);
246             break;
247         case KeyMap::IT:
248             patch(map, patchIT);
249             break;
250         case KeyMap::NO:
251             patch(map, patchNO);
252             break;
253         case KeyMap::MX:
254             patch(map, patchMX);
255             break;
256         case KeyMap::ES:
257             patch(map, patchES);
258             break;
259         case KeyMap::SE:
260             patch(map, patchSE);
261             break;
262         default:;
263             // English QWERTY - no patch needed
264         }
265         if(KeyMap::isISO(layout))
266             patchISO(map);
267         else
268             patchANSI(map);
269         // Done! return the map
270         break;
271     }
```

```
    case KeyMap::K70:{
        // The K70 maps are based on the K95 maps. However all the keys are shifted left and the G keys are removed
        map = getMap(KeyMap::K95, layout);
        QMutableHashIterator<QString, Key> i(map);
        while(i.hasNext()){
            i.next();
            // Move key to left. Remove it if it fell off the edge
            if((i.value().x -= K70_X_START) < 0)
                i.remove();
        }
        // Remove the M buttons as well
        map.remove("mr");
        map.remove("m1");
        map.remove("m2");
        map.remove("m3");
        // Done!
        break;
    }
    case KeyMap::K65:{
        // The K65 maps additionally remove the numpad and have a modified top row
        map = getMap(KeyMap::K70, layout);
        QMutableHashIterator<QString, Key> i(map);
        while(i.hasNext()){
            i.next();
            if(i.value().x >= K65_WIDTH)
                i.remove();
        }
        for(const Key* key = K65TopRow; key < K65TopRow + K65_TOP_COUNT; key++)
            map[key->name] = *key;
        // Done!
        break;
    }
    case KeyMap::STRAFE:{
        // The Strafe RGB maps are based on the K70 map minus the media keys
        map = getMap(KeyMap::K70, layout);
        //move light and lock right
        map["light"].x=285 - K70_X_START;
        //map["light"].hasLed=false;
        map["lock"].x=297 - K70_X_START;
        //map["lock"].hasLed=false;
        // move everything right to make the space for the left sidelight
        QMutableHashIterator<QString, Key> i(map);
        while(i.hasNext()){
            i.next();
            i.value().x += KSTRAFE_X_START;
        }
        // Add Strafe lights and keys
        map["lsidel"] = KStrafeKeys[0];
        map["rsidel"] = KStrafeKeys[1];
        map["logo"] = KStrafeKeys[2];
        map["fn"] = KStrafeKeys[3];
        map.remove("rwin");
        // remove media controls
        map.remove("mute");
        map.remove("volup");
        map.remove("voldn");
        map.remove("stop");
        map.remove("prev");
        map.remove("play");
        map.remove("next");
        // Done!
        break;
    }
    case KeyMap::M65:{
        // M65 isn't a keyboard; all mouse maps are unique.
        for(const Key* key = M65Keys; key < M65Keys + KEYCOUNT_M65; key++){
            // Keyboard keys are written from the center because that's where the LEDs are, but the mouse buttons are odd shapes so they're
            // written from the upper left
            Key translatedKey = *key;
            translatedKey.x += translatedKey.width / 2;
            translatedKey.y += translatedKey.height / 2;
            map[key->name] = translatedKey;
        }
        // Mice also have no layout patches - no other changes necessary
        break;
    }
    case KeyMap::SABRE:{
        // Scimitar mouse
        for(const Key* key = SabreKeys; key < SabreKeys + KEYCOUNT_SABRE; key++){
            // Like the M65, the keys are upper-left justified
            Key translatedKey = *key;
            translatedKey.x += translatedKey.width / 2;
```

```
            translatedKey.y += translatedKey.height / 2;
            map[key->name] = translatedKey;
        }
        break;
    }
    case KeyMap::SCIMITAR:{
        // Scimitar mouse
        for(const Key* key = ScimKeys; key < ScimKeys +
    KEYCOUNT_SCIM; key++){
            Key translatedKey = *key;
            translatedKey.x += translatedKey.width / 2;
            translatedKey.y += translatedKey.height / 2;
            map[key->name] = translatedKey;
        }
        break;
    }
    default:;    // <- stop GCC from complaining
    }
    // Map is finished, return result
    return map;
}
```

Here is the call graph for this function:

9.58.3.3 static void patchANSI (QHash< QString, Key > & *map*) [static]

Definition at line 119 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, Key::width, Key::x, and Key::y.

Referenced by getMap().

```
{
    map.remove("bslash_iso");
    map.remove("hash");
    Key& enter = map["enter"];
    enter.x = ANSI_ENTER_X;
    enter.y = ANSI_ENTER_Y;
    enter.width = ANSI_ENTER_W;
    enter.height = ANSI_ENTER_H;
    Key& lshift = map["lshift"];
    lshift.x = ANSI_LSHIFT_X;
    lshift.width = ANSI_LSHIFT_W;
}
```

Here is the caller graph for this function:

9.58.3.4 static void patchISO (QHash< QString, Key > & *map*) [static]

Definition at line 116 of file keymap.cpp.

Referenced by getMap().

```
116                                                 {
117     map.remove("bslash");
118 }
```

Here is the caller graph for this function:

9.58.4 Variable Documentation

9.58.4.1 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 145 of file keymap.cpp.

9.58.4.2 QHash<QString, Key> K95BaseMap [static]

Definition at line 208 of file keymap.cpp.

Referenced by getMap().

9.58.4.3 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, "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, "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, "Space", "
      space", 116, 75, 84, 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 10 of file keymap.cpp.

9.58.4.4 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 155 of file keymap.cpp.

9.58.4.5 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 163 of file keymap.cpp.

9.58.4.6 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 55 of file keymap.cpp.

9.58.4.7 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 36 of file keymap.cpp.

9.58.4.8 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 97 of file keymap.cpp.

9.58.4.9 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 83 of file keymap.cpp.

9.58.4.10 const KeyPatch patchEU[] [static]

Initial value:

```
= {
    {0, "\\ (R)", "hash"},
    {0, "\\ (L)", "bslash_iso"}
}
```

Definition at line 43 of file keymap.cpp.

9.58.4.11 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 48 of file keymap.cpp.

9.58.4.12 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 62 of file keymap.cpp.

9.58.4.13 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 76 of file keymap.cpp.

9.58.4.14 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 69 of file keymap.cpp.

9.58.4.15 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 90 of file keymap.cpp.

9.58.4.16 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 176 of file keymap.cpp.

9.58.4.17 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 189 of file keymap.cpp.

9.58.4.18 QHash<QString, Key> standardMaps[KeyMap::_MODEL_MAX][KeyMap::_LAYOUT_MAX] [static]

Definition at line 209 of file keymap.cpp.

Referenced by getMap().

9.59 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

9.60 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

9.60.1 Function Documentation

9.60.1.1 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:

9.60.2 Variable Documentation

9.60.2.1 const int KEY_SIZE = 12 [static]

Definition at line 10 of file keywidget.cpp.

Referenced by KeyWidget::drawInfo(), and KeyWidget::map().

9.60.2.2 QImage* m65Overlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

9.60.2.3 QImage * sabOverlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

9.60.2.4 QImage * scimOverlay = 0 [static]

Definition at line 12 of file keywidget.cpp.

Referenced by KeyWidget::paintEvent().

9.61 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

9.62 src/ckb/kperfwidget.cpp File Reference

```
#include "kperfwidget.h"
#include "ui_kperfwidget.h"
#include "modeselectdialog.h"
#include <cmath>
```

Include dependency graph for kperfwidget.cpp:

9.63 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

9.63.1 Data Structure Documentation

9.63.1.1 struct KPerfWidget::IndicatorUi

Definition at line 48 of file kperfwidget.h.

Collaboration diagram for KPerfWidget::IndicatorUi:

Data Fields

ColorButton *	color1	
ColorButton *	color2	
ColorButton *	color3	
QCheckBox *	enable	
QComboBox *	hwEnable	

9.64 src/ckb/layoutdialog.cpp File Reference

```
#include "layoutdialog.h"
```

```
#include "kb.h"
#include "ui_layoutdialog.h"
```

Include dependency graph for layoutdialog.cpp:

9.65 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

9.66 src/ckb/macroreader.cpp File Reference

```
#include <qdebug.h>
#include "macroreader.h"
```

Include dependency graph for macroreader.cpp:

9.67 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.

9.68 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[])

9.68.1 Macro Definition Documentation

9.68.1.1 #define SHMEM_SIZE 128 ∗ 1024

Definition at line 20 of file main.cpp.

Referenced by isRunning().

9.68.1.2 #define SHMEM_SIZE_V015 16

Definition at line 21 of file main.cpp.

Referenced by isRunning().

9.68.2 Enumeration Type Documentation

9.68.2.1 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 };
```

9.68.3 Function Documentation

9.68.3.1 QSharedMemory appShare ("ckb")

Referenced by isRunning().

Here is the caller graph for this function:

9.68.3.2 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)
```

```
    if(!appShare.create(SHMEM_SIZE) || !appShare.lock()){
        // Lock existing shared memory
        if(!appShare.attach() || !appShare.lock())
            return true;
        bool running = false;
        if(appShare.size() == SHMEM_SIZE_V015){
            // Old shmem - no PID listed so assume the process is running, and print the command directly to the buffer
            if(command){
                void* data = appShare.data();
                snprintf((char*)data, SHMEM_SIZE_V015, "%s", command);
            }
            running = true;
        } else {
            // New shmem. Scan the contents of the shared memory for a PID
            QStringList lines = QString((const char*)appShare.constData()).split("\n");
            if(pidActive(lines)){
                running = true;
                // Append the command
                if(command){
                    lines.append(QString("Option ") + command);
                    QByteArray newMem = lines.join("\n").left(SHMEM_SIZE).toLatin1();
                    // Copy the NUL byte as well as the string
                    memcpy(appShare.data(), newMem.constData(), newMem.length() + 1);
                }
            }
        }
        if(running){
            appShare.unlock();
            return true;
        }
    }
    // Not already running. Initialize the shared memory with our PID
    snprintf((char*)appShare.data(), appShare.size(), "PID %ld", (long)getpid());
    appShare.unlock();
    return false;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.68.3.3 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().

```
                                {
    // Setup main application
    QApplication a(argc, argv);

    // Setup names and versions
    QCoreApplication::setOrganizationName("ckb");
    QCoreApplication::setApplicationVersion(CKB_VERSION_STR);
    QCoreApplication::setApplicationName("ckb");

    // Setup argument parser
    QCommandLineParser parser;
    QString errorMessage;
    parser.setApplicationDescription("Open Source Corsair Input Device Driver for Linux and OSX.");
    bool background = 0;

    // Although the daemon runs as root, the GUI needn't and shouldn't be, as it has the potential to corrupt settings data.
    if(getuid() == 0){
        printf("The ckb GUI should not be run as root.\n");
        return 0;
    }

    // Seed the RNG for UsbIds
    qsrand(QDateTime::currentMSecsSinceEpoch());
#ifdef Q_OS_MACX
    disableAppNap();
#endif
#if QT_VERSION >= QT_VERSION_CHECK(5, 3, 0)
    // Enable HiDPI support
    qApp->setAttribute(Qt::AA_UseHighDpiPixmaps);
#endif

    // Parse arguments
    switch (parseCommandLine(parser, &errorMessage)) {
    case CommandLineOK:
        // If launched with no argument
        break;
    case CommandLineError:
        fputs(qPrintable(errorMessage), stderr);
        fputs("\n\n", stderr);
        fputs(qPrintable(parser.helpText()), stderr);
        return 1;
    case CommandLineVersionRequested:
        // If launched with --version, print version info and then quit
        printf("%s %s\n", qPrintable(QCoreApplication::applicationName()),
               qPrintable(QCoreApplication::applicationVersion()));
        return 0;
    case CommandLineHelpRequested:
        // If launched with --help, print help and then quit
        parser.showHelp();
        return 0;
    case CommandLineClose:
        // If launched with --close, kill existing app
        if (isRunning("Close"))
            printf("Asking existing instance to close.\n");
        else
            printf("ckb is not running.\n");
        return 0;
    case CommandLineBackground:
        // If launched with --background, launch in background
        background = 1;
        break;
    }

    // Launch in background if requested, or if re-launching a previous session
    if(qApp->isSessionRestored())
            background = 1;
    if(isRunning(background ? 0 : "Open")){
        printf("ckb is already running. Exiting.\n");
        return 0;
    }
    MainWindow w;
    if(!background)
        w.show();

    return a.exec();
}
```

Here is the call graph for this function:

9.68.3.4 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)) {
```

```
        // print help and exit
        return CommandLineHelpRequested;
    }

    if (parser.isSet(backgroundOption)) {
        // open application in background
        return CommandLineBackground;
    }

    if (parser.isSet(closeOption)) {
        // close already running application instances, if any
        return CommandLineClose;
    }

    /* no explicit argument was passed */
    return CommandLineOK;
};
```

Here is the caller graph for this function:

9.68.3.5 static bool pidActive (const QStringList & *lines*) [static]

Definition at line 92 of file main.cpp.

Referenced by isRunning().

```
                                                {
    foreach(const QString& line, lines){
        if(line.startsWith("PID ")){
            bool ok;
            pid_t pid;
            // Valid PID found?
            if((pid = line.split(" ")[1].toLong(&ok)) > 0 && ok){
                // kill -0 does nothing to the application, but checks if it's running
                 return (kill(pid, 0) == 0 || errno != ESRCH);
             }
         }
     }
     // If the PID wasn't listed in the shmem, assume it is running
     return true;
}
```

Here is the caller graph for this function:

9.69 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 <QSharedMemory>
#include <QShortcut>
#include <QMessageBox>
#include <QMenuBar>
#include <unistd.h>
```

Include dependency graph for mainwindow.cpp:

Variables

- QSharedMemory appShare
- static const QString configLabel = "Settings"

9.69.1 Variable Documentation

9.69.1.1 QSharedMemory appShare

Referenced by MainWindow::timerTick().

9.69.1.2 const QString configLabel = "Settings" [static]

Definition at line 15 of file mainwindow.cpp.

Referenced by MainWindow::MainWindow().

9.70 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

9.71 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 ()

9.71.1 Macro Definition Documentation

9.71.1.1 #define ENUM_C(*name*) typedef enum

Definition at line 13 of file media.h.

9.71.1.2 #define ENUM_END_C(*name*) name

Definition at line 14 of file media.h.

9.71.1.3 #define EXTERN_C extern

Definition at line 12 of file media.h.

9.71.2 Enumeration Type Documentation

9.71.2.1 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);
```

9.71.3 Function Documentation

9.71.3.1 muteState getMuteState ( )

Definition at line 9 of file media_linux.cpp.

References MUTED, muteProcess, UNKNOWN, and UNMUTED.

Referenced by KbPerf::applyIndicators().

```
9                         {
10     // Get default sink mute state from pulseaudio
11     static muteState lastKnown = UNKNOWN;
12     static quint64 lastTime = 0;
13     // Instead of running a command to check the state and waiting for it to finish, run the command now
       but wait to check it until the next frame
14     // (locking up the GUI thread is bad)
15     if(lastTime > 0 && muteProcess.state() == QProcess::NotRunning){
16         if(muteProcess.exitCode() != 0)
17             lastKnown = UNKNOWN;
18         QString output = muteProcess.readLine().trimmed();
19         if(output == "yes")
20             lastKnown = MUTED;
21         else if(output == "no")
22             lastKnown = UNMUTED;
23     }
24     quint64 time = QDateTime::currentMSecsSinceEpoch();
25     if(time - lastTime < 33)
26         // Don't run it than 30 times per second
27         return lastKnown;
28     lastTime = time;
29     if(muteProcess.state() == QProcess::NotRunning)
30         // Shamelessly taken from pulseaudio-ctl
31         muteProcess.start("sh", QStringList() << "-c" << "pacmd list-sinks|grep -A 15 '*
       index'|awk '/muted:/{ print $2 }'");
32     return lastKnown;
33 }
```

Here is the caller graph for this function:

9.72 src/ckb/media_linux.cpp File Reference

```
#include <QDateTime>
#include <QProcess>
#include "media.h"
```

Include dependency graph for media_linux.cpp:

Functions

- muteState getMuteState ()

Variables

- static QProcess muteProcess

9.72.1 Function Documentation

9.72.1.1 muteState getMuteState ( )

Definition at line 9 of file media_linux.cpp.

References MUTED, muteProcess, UNKNOWN, and UNMUTED.

```
9                           {
10      // Get default sink mute state from pulseaudio
11      static muteState lastKnown = UNKNOWN;
12      static quint64 lastTime = 0;
13      // Instead of running a command to check the state and waiting for it to finish, run the command now
        but wait to check it until the next frame
14      // (locking up the GUI thread is bad)
15      if(lastTime > 0 && muteProcess.state() == QProcess::NotRunning){
16          if(muteProcess.exitCode() != 0)
17              lastKnown = UNKNOWN;
18          QString output = muteProcess.readLine().trimmed();
19          if(output == "yes")
20              lastKnown = MUTED;
21          else if(output == "no")
22              lastKnown = UNMUTED;
23      }
24      quint64 time = QDateTime::currentMSecsSinceEpoch();
25      if(time - lastTime < 33)
26          // Don't run it than 30 times per second
27          return lastKnown;
28      lastTime = time;
29      if(muteProcess.state() == QProcess::NotRunning)
30          // Shamelessly taken from pulseaudio-ctl
31          muteProcess.start("sh", QStringList() << "-c" << "pacmd list-sinks|grep -A 15 '*
        index'|awk '/muted:/{ print $2 }'");
32      return lastKnown;
33  }
```

9.72.2 Variable Documentation

9.72.2.1 QProcess muteProcess [static]

Definition at line 7 of file media_linux.cpp.

Referenced by getMuteState().

9.73 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 ()

9.73.1 Function Documentation

9.73.1.1 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:

9.73.1.2 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:

9.74 src/ckb/modeselectdialog.cpp File Reference

```
#include "modeselectdialog.h"
#include "ui_modeselectdialog.h"
```

Include dependency graph for modeselectdialog.cpp:

9.75 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

9.76 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

9.76.1 Macro Definition Documentation

9.76.1.1 #define SET_END isSetting = false

Definition at line 126 of file mperfwidget.cpp.

Referenced by MPerfWidget::boxXChanged(), MPerfWidget::boxYChanged(), MPerfWidget::sliderXMoved(), and MPerfWidget::sliderYMoved().

9.76.1.2 #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().

9.76.2 Function Documentation

9.76.2.1 int dpiExp (int *value*, int *min*, int *max*) [inline]

Definition at line 105 of file mperfwidget.cpp.

References DPI_BASE, and KbPerf::DPI_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 caller graph for this function:

9.76.2.2 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:

9.76.2.3 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:

9.76.3 Variable Documentation

9.76.3.1 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().

9.76.3.2 const QString xyLinkPath = "UI/DPI/UnlinkXY" [static]

Definition at line 6 of file mperfwidget.cpp.

Referenced by MPerfWidget::on_xyBox_clicked().

9.77 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

9.77.1 Data Structure Documentation

9.77.1.1 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	

9.78 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)

9.78.1 Macro Definition Documentation

9.78.1.1 #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().

9.78.1.2 #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().

9.78.1.3 #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().

9.78.2 Function Documentation

9.78.2.1 static int decrypt_byte (unsigned long * *pkeys*, const z_crc_t FAR * *pcrc_32_tab*) [static]

Definition at line 37 of file crypt.h.

```
{
    //(void) pcrc_32_tab; /* avoid "unused parameter" warning */
    unsigned temp;  /* POTENTIAL BUG:  temp*(temp^1) may overflow in an
                     * unpredictable manner on 16-bit systems; not a problem
                     * with any known compiler so far, though */

    temp = ((unsigned)(*(pkeys+2)) & 0xffff) | 2;
    return (int)(((temp * (temp ^ 1)) >> 8) & 0xff);
}
```

9.78.2.2 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().

```
{
    *(pkeys+0) = 305419896L;
    *(pkeys+1) = 591751049L;
    *(pkeys+2) = 878082192L;
    while (*passwd != '\0') {
        update_keys(pkeys,pcrc_32_tab,(int)*passwd);
        passwd++;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.78.2.3 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().

```
{
    (*(pkeys+0)) = CRC32((*(pkeys+0)), c);
    (*(pkeys+1)) += (*(pkeys+0)) & 0xff;
    (*(pkeys+1)) = (*(pkeys+1)) * 134775813L + 1;
    {
      register int keyshift = (int)((*(pkeys+1)) >> 24);
      (*(pkeys+2)) = CRC32((*(pkeys+2)), keyshift);
    }
    return c;
}
```

Here is the caller graph for this function:

9.79 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)

9.79.1 Data Structure Documentation

9.79.1.1 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	

9.79.1.2 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	

9.79.1.3 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	

9.79.2 Macro Definition Documentation

9.79.2.1 #define __USE_FILE_OFFSET64

Definition at line 32 of file ioapi.h.

9.79.2.2 #define __USE_LARGEFILE64

Definition at line 35 of file ioapi.h.

9.79.2.3 #define _FILE_OFFSET_BIT 64

Definition at line 41 of file ioapi.h.

9.79.2.4 #define _LARGEFILE64_SOURCE

Definition at line 38 of file ioapi.h.

9.79.2.5 #define OF _Z_OF

Definition at line 106 of file ioapi.h.

9.79.2.6 #define ZCALLBACK

Definition at line 125 of file ioapi.h.

9.79.2.7 #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().

9.79.2.8 #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().

9.79.2.9 #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().

9.79.2.10 #define ZLIB_FILEFUNC_MODE_CREATE (8)

Definition at line 118 of file ioapi.h.

Referenced by qiodevice_open_file_func(), and zipOpen3().

9.79.2.11 #define ZLIB_FILEFUNC_MODE_EXISTING (4)

Definition at line 117 of file ioapi.h.

Referenced by qiodevice_open_file_func(), unzOpenInternal(), and zipOpen3().

9.79.2.12 #define ZLIB_FILEFUNC_MODE_READ (1)

Definition at line 113 of file ioapi.h.

Referenced by qiodevice_open_file_func(), unzOpenInternal(), and zipOpen3().

9.79.2.13 #define ZLIB_FILEFUNC_MODE_READWRITEFILTER (3)

Definition at line 115 of file ioapi.h.

Referenced by qiodevice_open_file_func().

9.79.2.14 #define ZLIB_FILEFUNC_MODE_WRITE (2)

Definition at line 114 of file ioapi.h.

Referenced by zipOpen3().

9.79.2.15 #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().

9.79.2.16 #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().

9.79.2.17 #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().

9.79.2.18 #define ZOPEN64(*filefunc, filename, mode*) (call_zopen64((&(filefunc)),(filename),(mode)))

Definition at line 199 of file ioapi.h.

Referenced by unzOpenInternal(), and zipOpen3().

9.79.2.19 #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().

9.79.2.20 #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().

9.79.2.21 #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().

9.79.2.22 #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().

9.79.3 Typedef Documentation

9.79.3.1 zipFile ZEXPORT zipOpen64 _Z_OF ((voidpf opaque, voidpf file, int mode))

Addition for GDAL : END.

Definition at line 132 of file ioapi.h.

9.79.3.2 typedef struct zlib_filefunc64_32_def_s zlib_filefunc64_32_def

9.79.3.3 typedef struct zlib_filefunc64_def_s zlib_filefunc64_def

9.79.3.4 typedef struct zlib_filefunc_def_s zlib_filefunc_def

9.79.3.5 typedef unsigned long long int ZPOS64_T

Definition at line 94 of file ioapi.h.

9.79.4 Function Documentation

9.79.4.1 void fill_qiodevice64_filefunc _Z_OF ((zlib_filefunc64_def ∗pzlib_filefunc_def))

9.79.4.2 void fill_qiodevice_filefunc _Z_OF ((zlib_filefunc_def ∗pzlib_filefunc_def))

9.79.4.3 voidpf call_zopen64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf file, int mode))

9.79.4.4 int call_zseek64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream, ZPOS64_T offset, int origin))

9.79.4.5 ZPOS64_T call_ztell64 _Z_OF ((const zlib_filefunc64_32_def ∗pfilefunc, voidpf filestream))

9.79.4.6 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:

9.80 src/ckb/quazip/JlCompress.cpp File Reference

```
#include "JlCompress.h"
#include <QDebug>
```

Include dependency graph for JICompress.cpp:

Functions

- static bool copyData (QIODevice &inFile, QIODevice &outFile)

9.80.1 Function Documentation

9.80.1.1 static bool copyData (QIODevice & *inFile*, QIODevice & *outFile*) [static]

Definition at line 29 of file JICompress.cpp.

Referenced by JICompress::compressFile(), and JICompress::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:

9.81 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.

9.82 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)

9.82.1 Macro Definition Documentation

9.82.1.1 #define SEEK_CUR 1

Definition at line 29 of file qioapi.cpp.

9.82.1.2 #define SEEK_END 2

Definition at line 33 of file qioapi.cpp.

9.82.1.3 #define SEEK_SET 0

Definition at line 37 of file qioapi.cpp.

Referenced by unzOpenCurrentFile3().

9.82.2 Function Documentation

9.82.2.1 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.

```
{
    if (pfilefunc->zfile_func64.zopen64_file != NULL)
        return (*(pfilefunc->zfile_func64.zopen64_file)) (pfilefunc->zfile_func64.opaque,file,mode);
    else
    {
        return (*(pfilefunc->zopen32_file))(pfilefunc->zfile_func64.opaque,file,mode);
    }
}
```

9.82.2.2 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.

```
{
    if (pfilefunc->zfile_func64.zseek64_file != NULL)
        return (*(pfilefunc->zfile_func64.zseek64_file)) (pfilefunc->zfile_func64.opaque,filestream,offset,origin);
    else
    {
        uLong offsetTruncated = (uLong)offset;
        if (offsetTruncated != offset)
            return -1;
        else
            return (*(pfilefunc->zseek32_file))(pfilefunc->zfile_func64.opaque,filestream,offsetTruncated,origin);
    }
}
```

9.82.2.3 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.

```
{
    if (pfilefunc->zfile_func64.zseek64_file != NULL)
        return (*(pfilefunc->zfile_func64.ztell64_file)) (pfilefunc->zfile_func64.opaque,filestream);
    else
    {
        uLong tell_uLong = (*(pfilefunc->ztell32_file))(pfilefunc->zfile_func64.opaque,filestream);
        if ((tell_uLong) == ((uLong)-1))
            return (ZPOS64_T)-1;
        else
            return tell_uLong;
    }
}
```

9.82.2.4 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().

```
{
    // Open functions are the same for Qt.
    pzlib_filefunc_def->zopen64_file = qiodevice_open_file_func;
    pzlib_filefunc_def->zread_file = qiodevice_read_file_func;
    pzlib_filefunc_def->zwrite_file = qiodevice_write_file_func;
    pzlib_filefunc_def->ztell64_file = qiodevice64_tell_file_func;
    pzlib_filefunc_def->zseek64_file = qiodevice64_seek_file_func;
    pzlib_filefunc_def->zclose_file = qiodevice_close_file_func;
    pzlib_filefunc_def->zerror_file = qiodevice_error_file_func;
    pzlib_filefunc_def->opaque = new QIODevice_descriptor;
    pzlib_filefunc_def->zfakeclose_file = qiodevice_fakeclose_file_func;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.82.2.5 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 {
```

```
    pzlib_filefunc_def->zopen_file = qiodevice_open_file_func;
    pzlib_filefunc_def->zread_file = qiodevice_read_file_func;
    pzlib_filefunc_def->zwrite_file = qiodevice_write_file_func;
    pzlib_filefunc_def->ztell_file = qiodevice_tell_file_func;
    pzlib_filefunc_def->zseek_file = qiodevice_seek_file_func;
    pzlib_filefunc_def->zclose_file = qiodevice_close_file_func;
    pzlib_filefunc_def->zerror_file = qiodevice_error_file_func;
    pzlib_filefunc_def->opaque = new QIODevice_descriptor;
}
```

Here is the call graph for this function:

9.82.2.6 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().

```
{
    p_filefunc64_32->zfile_func64.zopen64_file = NULL;
    p_filefunc64_32->zopen32_file = p_filefunc32->zopen_file;
    p_filefunc64_32->zfile_func64.zerror_file = p_filefunc32->
  zerror_file;
    p_filefunc64_32->zfile_func64.zread_file = p_filefunc32->
```

```
zread_file;
  p_filefunc64_32->zfile_func64.zwrite_file = p_filefunc32->zwrite_file;
  p_filefunc64_32->zfile_func64.ztell64_file = NULL;
  p_filefunc64_32->zfile_func64.zseek64_file = NULL;
  p_filefunc64_32->zfile_func64.zclose_file = p_filefunc32->zclose_file;
  p_filefunc64_32->zfile_func64.zerror_file = p_filefunc32->zerror_file;
  p_filefunc64_32->zfile_func64.opaque = p_filefunc32->opaque;
  p_filefunc64_32->zfile_func64.zfakeclose_file = NULL;
  p_filefunc64_32->zseek32_file = p_filefunc32->zseek_file;
  p_filefunc64_32->ztell32_file = p_filefunc32->ztell_file;
}
```

Here is the caller graph for this function:

9.82.2.7 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().

```
{
    QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
    if (iodevice->isSequential()) {
        if (origin == ZLIB_FILEFUNC_SEEK_END
                && offset == 0) {
            // sequential devices are always at end (needed in mdAppend)
            return 0;
        } else {
            qWarning("qiodevice_seek_file_func() called for sequential device");
            return -1;
        }
    }
    qint64 qiodevice_seek_result=0;
    int ret;
    switch (origin)
    {
    case ZLIB_FILEFUNC_SEEK_CUR :
        qiodevice_seek_result = ((QIODevice*)stream)->pos() + offset;
        break;
    case ZLIB_FILEFUNC_SEEK_END :
        qiodevice_seek_result = ((QIODevice*)stream)->size() - offset;
        break;
    case ZLIB_FILEFUNC_SEEK_SET :
        qiodevice_seek_result = offset;
        break;
    default:
        return -1;
    }
    ret = !iodevice->seek(qiodevice_seek_result);
    return ret;
}
```

Here is the caller graph for this function:

9.82.2.8 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:

9.82.2.9 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     }
```

```
#endif
    device->close();
    return 0;
}
```

Here is the caller graph for this function:

9.82.2.10 int qiodevice_error_file_func (voidpf , voidpf)

Definition at line 310 of file qioapi.cpp.

Referenced by fill_qiodevice64_filefunc(), and fill_qiodevice_filefunc().

```
{
    // can't check for error due to the QIODevice API limitation
    return 0;
}
```

Here is the caller graph for this function:

9.82.2.11 int qiodevice_fakeclose_file_func (voidpf *opaque,* voidpf)

Definition at line 301 of file qioapi.cpp.

Referenced by fill_qiodevice64_filefunc().

```
{
    QIODevice_descriptor *d = reinterpret_cast<
  QIODevice_descriptor*>(opaque);
    delete d;
    return 0;
}
```

Here is the caller graph for this function:

9.82.2.12 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:

9.82.2.13 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().

```
{
    QIODevice_descriptor *d = reinterpret_cast<
  QIODevice_descriptor*>(opaque);
    QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
    qint64 ret64 = iodevice->read((char*)buf,size);
    uLong ret;
    ret = (uLong) ret64;
    if (ret64 != -1) {
        d->pos += ret64;
    }
    return ret;
}
```

Here is the caller graph for this function:

9.82.2.14 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().

```
{
    QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
    if (iodevice->isSequential()) {
        if (origin == ZLIB_FILEFUNC_SEEK_END
                && offset == 0) {
            // sequential devices are always at end (needed in mdAppend)
            return 0;
        } else {
            qWarning("qiodevice_seek_file_func() called for sequential device");
            return -1;
```

```
        }
    }
    uLong qiodevice_seek_result=0;
    int ret;
    switch (origin)
    {
    case ZLIB_FILEFUNC_SEEK_CUR :
        qiodevice_seek_result = ((QIODevice*)stream)->pos() + offset;
        break;
    case ZLIB_FILEFUNC_SEEK_END :
        qiodevice_seek_result = ((QIODevice*)stream)->size() - offset;
        break;
    case ZLIB_FILEFUNC_SEEK_SET :
        qiodevice_seek_result = offset;
        break;
    default:
        return -1;
    }
    ret = !iodevice->seek(qiodevice_seek_result);
    return ret;
}
```

Here is the caller graph for this function:

9.82.2.15 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:

9.82.2.16 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().

```
{
    QIODevice_descriptor *d = reinterpret_cast<QIODevice_descriptor*>(opaque);
    QIODevice *iodevice = reinterpret_cast<QIODevice*>(stream);
    uLong ret;
    qint64 ret64 = iodevice->write((char*)buf,size);
    if (ret64 != -1) {
        d->pos += ret64;
    }
    ret = (uLong) ret64;
    return ret;
}
```

Here is the caller graph for this function:

9.83 src/ckb/quazip/quaadler32.cpp File Reference

```
#include "quaadler32.h"
#include "zlib.h"
```

Include dependency graph for quaadler32.cpp:

9.84 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.

9.85 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.

9.86 src/ckb/quazip/quacrc32.cpp File Reference

```
#include "quacrc32.h"
#include "zlib.h"
```

Include dependency graph for quacrc32.cpp:

9.87 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.

9.88 src/ckb/quazip/quagzipfile.cpp File Reference

```
#include <QFile>
#include "quagzipfile.h"
```

Include dependency graph for quagzipfile.cpp:

9.89 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.

9.90 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

9.90.1 Macro Definition Documentation

9.90.1.1 #define QUAZIO_INBUFSIZE 4096

Definition at line 27 of file quaziodevice.cpp.

Referenced by QuaZIODevice::readData().

9.90.1.2 #define QUAZIO_OUTBUFSIZE 4096

Definition at line 28 of file quaziodevice.cpp.

Referenced by QuaZIODevice::flush(), and QuaZIODevice::writeData().

9.91 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.

9.92 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)

9.92.1 Function Documentation

9.92.1.1 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);
```

```
    return info;
}
```

Here is the call graph for this function:

9.92.1.2 template<> QuaZipFileInfo QuaZip_getFileInfo (QuaZip * *zip*, bool * *ok*)

Definition at line 663 of file quazip.cpp.

References QuaZip::getCurrentFileInfo().

```
{
    QuaZipFileInfo info;
    *ok = zip->getCurrentFileInfo(&info);
    return info;
}
```

Here is the call graph for this function:

9.92.1.3 template<> QuaZipFileInfo64 QuaZip_getFileInfo (QuaZip * *zip*, bool * *ok*)

Definition at line 671 of file quazip.cpp.

References QuaZip::getCurrentFileInfo().

```
{
    QuaZipFileInfo64 info;
    *ok = zip->getCurrentFileInfo(&info);
    return info;
}
```

Here is the call graph for this function:

9.92.1.4 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:

9.93 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

9.93.1 Macro Definition Documentation

9.93.1.1 #define UNZ_OPENERROR -1000

Definition at line 41 of file quazip.h.

Referenced by QuaZip::open().

9.94 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

9.94.1 Macro Definition Documentation

9.94.1.1 #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.

9.94.1.2 #define QUAZIP_EXTRA_NTFS_MAGIC 0x000Au

Definition at line 56 of file quazip_global.h.

Referenced by getNTFSTime(), and setNTFSTime().

9.94.1.3 #define QUAZIP_EXTRA_NTFS_TIME_MAGIC 0x0001u

Definition at line 57 of file quazip_global.h.

Referenced by getNTFSTime(), and setNTFSTime().

9.94.1.4 #define UNUSED

Definition at line 53 of file quazip_global.h.

9.95 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)

9.95.1 Function Documentation

9.95.1.1 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 }
```

9.95.1.2 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 }
```

9.95.1.3 static void QuaZipDir_convertInfoList (const QList< QuaZipFileInfo64 > & *from*, QList< QuaZipFileInfo > & *to*) [static]

Definition at line 205 of file quazipdir.cpp.

```
{
    to.clear();
    for (QList<QuaZipFileInfo64>::const_iterator i = from.constBegin();
            i != from.constEnd();
            ++i) {
        QuaZipFileInfo info32;
        i->toQuaZipFileInfo(info32);
        to.append(info32);
    }
}
```

9.95.1.4 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.

```
{
    QuaZipFileInfo64 info;
    if (isReal) {
        *ok = zip->getCurrentFileInfo(&info);
    } else {
        *ok = true;
        info.compressedSize = 0;
        info.crc = 0;
        info.diskNumberStart = 0;
        info.externalAttr = 0;
        info.flags = 0;
        info.internalAttr = 0;
        info.method = 0;
        info.uncompressedSize = 0;
        info.versionCreated = info.versionNeeded = 0;
    }
    info.name = relativeName;
    return info;
}
```

Here is the call graph for this function:

9.96 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.

9.97 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.

9.98 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.

9.99 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)

9.99.1 Function Documentation

9.99.1.1 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().

```
{
    QDateTime dateTime;
    for (int i = 0; i <= extra.size() - 4; ) {
        unsigned type = static_cast<unsigned>(static_cast<unsigned char>(
                                                extra.at(i)))
                | (static_cast<unsigned>(static_cast<unsigned char>(
                                                 extra.at(i + 1))) << 8);
        i += 2;
        unsigned length = static_cast<unsigned>(static_cast<unsigned char>(
                                                 extra.at(i)))
                | (static_cast<unsigned>(static_cast<unsigned char>(
                                                 extra.at(i + 1))) << 8);
        i += 2;
        if (type == QUAZIP_EXTRA_NTFS_MAGIC && length >= 32) {
            i += 4; // reserved
            while (i <= extra.size() - 4) {
                unsigned tag = static_cast<unsigned>(
                            static_cast<unsigned char>(extra.at(i)))
                        | (static_cast<unsigned>(
                                static_cast<unsigned char>(extra.at(i + 1)))
                            << 8);
                i += 2;
                int tagsize = static_cast<unsigned>(
                            static_cast<unsigned char>(extra.at(i)))
                        | (static_cast<unsigned>(
                                static_cast<unsigned char>(extra.at(i + 1)))
                            << 8);
                i += 2;
                if (tag == QUAZIP_EXTRA_NTFS_TIME_MAGIC
                        && tagsize >= position + 8) {
                    i += position;
                    quint64 mtime = static_cast<quint64>(
                                 static_cast<unsigned char>(extra.at(i)))
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 1))) << 8)
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 2))) << 16)
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 3))) << 24)
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 4))) << 32)
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 5))) << 40)
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 6))) << 48)
                        | (static_cast<quint64>(static_cast<unsigned char>(
                                                extra.at(i + 7))) << 56);
                    // the NTFS time is measured from 1601 for whatever reason
                    QDateTime base(QDate(1601, 1, 1), QTime(0, 0), Qt::UTC);
                    dateTime = base.addMSecs(mtime / 10000);
                    if (fineTicks != NULL) {
                        *fineTicks = static_cast<int>(mtime % 10000);
                    }
                    i += tagsize - position;
                } else {
                    i += tagsize;
                }

            }
        } else {
            i += length;
        }
    }
    if (fineTicks != NULL && dateTime.isNull()) {
        *fineTicks = 0;
    }
    return dateTime;
}
```

Here is the caller graph for this function:

9.99.1.2 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:

9.100 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).

9.101 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)

9.101.1 Function Documentation

9.101.1.1 template<typename FileInfo > void QuaZipNewInfo_init (QuaZipNewInfo & *self*, const FileInfo & *existing*)

Definition at line 57 of file quazipnewinfo.cpp.

Referenced by QuaZipNewInfo::QuaZipNewInfo().

```
{
    self.name = existing.name;
    self.dateTime = existing.dateTime;
    self.internalAttr = existing.internalAttr;
    self.externalAttr = existing.externalAttr;
    self.comment = existing.comment;
    self.extraLocal = existing.extra;
    self.extraGlobal = existing.extra;
    self.uncompressedSize = existing.uncompressedSize;
}
```

Here is the caller graph for this function:

9.101.1.2 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().

```
{
    quint32 uPerm = isDir ? 0040000 : 0100000;
    if ((perm & QFile::ReadOwner) != 0)
        uPerm |= 0400;
    if ((perm & QFile::WriteOwner) != 0)
        uPerm |= 0200;
    if ((perm & QFile::ExeOwner) != 0)
        uPerm |= 0100;
    if ((perm & QFile::ReadGroup) != 0)
        uPerm |= 0040;
    if ((perm & QFile::WriteGroup) != 0)
        uPerm |= 0020;
    if ((perm & QFile::ExeGroup) != 0)
        uPerm |= 0010;
    if ((perm & QFile::ReadOther) != 0)
        uPerm |= 0004;
    if ((perm & QFile::WriteOther) != 0)
        uPerm |= 0002;
    if ((perm & QFile::ExeOther) != 0)
        uPerm |= 0001;
    info->externalAttr = (info->externalAttr & ~0xFFFF0000u) | (uPerm << 16);
}
```

Here is the caller graph for this function:

9.101.1.3 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().

```
                                        {
    int ntfsPos = -1, timesPos = -1;
    unsigned ntfsLength = 0, ntfsTimesLength = 0;
    for (int i = 0; i <= extra.size() - 4; ) {
        unsigned type = static_cast<unsigned>(static_cast<unsigned char>(
                                                extra.at(i)))
                | (static_cast<unsigned>(static_cast<unsigned char>(
                                                extra.at(i + 1))) << 8);
        i += 2;
        unsigned length = static_cast<unsigned>(static_cast<unsigned char>(
                                                extra.at(i)))
                | (static_cast<unsigned>(static_cast<unsigned char>(
                                                extra.at(i + 1))) << 8);
        i += 2;
        if (type == QUAZIP_EXTRA_NTFS_MAGIC) {
            ntfsPos = i - 4; // the beginning of the NTFS record
            ntfsLength = length;
            if (length <= 4) {
                break; // no times in the NTFS record
            }
            i += 4; // reserved
            while (i <= extra.size() - 4) {
                unsigned tag = static_cast<unsigned>(
                            static_cast<unsigned char>(extra.at(i)))
                        | (static_cast<unsigned>(
                                static_cast<unsigned char>(extra.at(i + 1)))
                            << 8);
                i += 2;
                unsigned tagsize = static_cast<unsigned>(
                            static_cast<unsigned char>(extra.at(i)))
                        | (static_cast<unsigned>(
                                static_cast<unsigned char>(extra.at(i + 1)))
                            << 8);
                i += 2;
                if (tag == QUAZIP_EXTRA_NTFS_TIME_MAGIC) {
                    timesPos = i - 4; // the beginning of the NTFS times tag
                    ntfsTimesLength = tagsize;
                    break;
                } else {
                    i += tagsize;
                }
            }
            break; // I ain't going to search for yet another NTFS record!
        } else {
            i += length;
        }
    }
    if (ntfsPos == -1) {
        // No NTFS record, need to create one.
        ntfsPos = extra.size();
        ntfsLength = 32;
        extra.resize(extra.size() + 4 + ntfsLength);
        // the NTFS record header
        extra[ntfsPos] = static_cast<char>(QUAZIP_EXTRA_NTFS_MAGIC);
        extra[ntfsPos + 1] = static_cast<char>(QUAZIP_EXTRA_NTFS_MAGIC >> 8);
        extra[ntfsPos + 2] = 32; // the 2-byte size in LittleEndian
        extra[ntfsPos + 3] = 0;
        // zero the record
        memset(extra.data() + ntfsPos + 4, 0, 32);
        timesPos = ntfsPos + 8;
        // now set the tag data
        extra[timesPos] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC);
        extra[timesPos + 1] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC
                                                >> 8);
        // the size:
        extra[timesPos + 2] = 24;
        extra[timesPos + 3] = 0;
        ntfsTimesLength = 24;
    }
    if (timesPos == -1) {
        // No time tag in the NTFS record, need to add one.
        timesPos = ntfsPos + 4 + ntfsLength;
        extra.resize(extra.size() + 28);
        // Now we need to move the rest of the field
        // (possibly zero bytes, but memmove() is OK with that).
        // 0 ......... ntfsPos .. ntfsPos + 4   ... timesPos
        // <some data> <header>   <NTFS record>     <need-to-move data>    <end>
        memmove(extra.data() + timesPos + 28, extra.data() + timesPos,
                extra.size() - 28 - timesPos);
        ntfsLength += 28;
        // now set the tag data
```

```
        extra[timesPos] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC);
        extra[timesPos + 1] = static_cast<char>(QUAZIP_EXTRA_NTFS_TIME_MAGIC
                                                >> 8);
        // the size:
        extra[timesPos + 2] = 24;
        extra[timesPos + 3] = 0;
        // zero the record
        memset(extra.data() + timesPos + 4, 0, 24);
        ntfsTimesLength = 24;
    }
    if (ntfsTimesLength < 24) {
        // Broken times field. OK, this is really unlikely, but just in case...
        size_t timesEnd = timesPos + 4 + ntfsTimesLength;
        extra.resize(extra.size() + (24 - ntfsTimesLength));
        // Move it!
        // 0 ......... timesPos .... timesPos + 4 .. timesEnd
        // <some data> <time header> <broken times> <need-to-move data> <end>
        memmove(extra.data() + timesEnd + (24 - ntfsTimesLength),
                extra.data() + timesEnd,
                extra.size() - (24 - ntfsTimesLength) - timesEnd);
        // Now we have to increase the NTFS record and time tag lengths.
        ntfsLength += (24 - ntfsTimesLength);
        ntfsTimesLength = 24;
        extra[ntfsPos + 2] = static_cast<char>(ntfsLength);
        extra[ntfsPos + 3] = static_cast<char>(ntfsLength >> 8);
        extra[timesPos + 2] = static_cast<char>(ntfsTimesLength);
        extra[timesPos + 3] = static_cast<char>(ntfsTimesLength >> 8);
    }
    QDateTime base(QDate(1601, 1, 1), QTime(0, 0), Qt::UTC);
#if (QT_VERSION >= 0x040700)
    quint64 ticks = base.msecsTo(time) * 10000 + fineTicks;
#else
    QDateTime utc = time.toUTC();
    quint64 ticks = (static_cast<qint64>(base.date().daysTo(utc.date()))
            * Q_INT64_C(86400000)
            + static_cast<qint64>(base.time().msecsTo(utc.time())))
        * Q_INT64_C(10000) + fineTicks;
#endif
    extra[timesPos + 4 + position] = static_cast<char>(ticks);
    extra[timesPos + 5 + position] = static_cast<char>(ticks >> 8);
    extra[timesPos + 6 + position] = static_cast<char>(ticks >> 16);
    extra[timesPos + 7 + position] = static_cast<char>(ticks >> 24);
    extra[timesPos + 8 + position] = static_cast<char>(ticks >> 32);
    extra[timesPos + 9 + position] = static_cast<char>(ticks >> 40);
    extra[timesPos + 10 + position] = static_cast<char>(ticks >> 48);
    extra[timesPos + 11 + position] = static_cast<char>(ticks >> 56);
}
```

Here is the caller graph for this function:

9.102 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.

9.103 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 []

9.103.1 Data Structure Documentation

9.103.1.1 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	

9.103.1.2 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	

9.103.1.3 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	is64bitOpenFunction	
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	

9.103.2 Macro Definition Documentation

9.103.2.1 #define ALLOC(*size*) (malloc(size))

Definition at line 114 of file unzip.c.

Referenced by unz64local_SearchCentralDir(), unz64local_SearchCentralDir64(), unzOpenCurrentFile3(), and unzOpenInternal().

9.103.2.2 #define BUFREADCOMMENT (0x400)

Definition at line 408 of file unzip.c.

Referenced by unz64local_SearchCentralDir(), and unz64local_SearchCentralDir64().

9.103.2.3 #define CASESENSITIVITYDEFAULT_NO

Definition at line 100 of file unzip.c.

9.103.2.4 #define CASESENSITIVITYDEFAULTVALUE 2

Definition at line 375 of file unzip.c.

Referenced by unzStringFileNameCompare().

9.103.2.5 #define local static

Definition at line 93 of file unzip.c.

9.103.2.6 #define SIZECENTRALDIRITEM (0x2e)

Definition at line 120 of file unzip.c.

Referenced by unzGoToNextFile().

9.103.2.7 #define SIZEZIPLOCALHEADER (0x1e)

Definition at line 121 of file unzip.c.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), and unzOpenCurrentFile3().

9.103.2.8 #define STRCMPCASENOSENTIVEFUNCTION strcmpcasenosensitive_internal

Definition at line 381 of file unzip.c.

Referenced by unzStringFileNameCompare().

9.103.2.9 #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().

9.103.2.10 #define UNZ_BUFSIZE (16384)

Definition at line 106 of file unzip.c.

Referenced by unzOpenCurrentFile3(), and unzReadCurrentFile().

9.103.2.11 #define UNZ_MAXFILENAMEINZIP (256)

Definition at line 110 of file unzip.c.

Referenced by unzLocateFile().

9.103.3 Typedef Documentation

9.103.3.1 typedef struct unz_file_info64_internal_s unz_file_info64_internal

9.103.3.2 typedef uLongf z_crc_t

Definition at line 76 of file unzip.c.

9.103.4 Function Documentation

9.103.4.1 static int unz64local_getByte _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi)) [static]

9.103.4.2 static int unz64local_getShort _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)) [static]

9.103.4.3 static int unz64local_getLong64 _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX)) [static]

9.103.4.4 static ZPOS64_T unz64local_SearchCentralDir _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)) [static]

9.103.4.5 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]

9.103.4.6 static int strcmpcasenosensitive_internal (const char * *fileName1*, const char * *fileName2*) [static]

Definition at line 352 of file unzip.c.

```
{
    for (;;)
    {
        char c1=*(fileName1++);
        char c2=*(fileName2++);
        if ((c1>='a') && (c1<='z'))
            c1 -= 0x20;
        if ((c2>='a') && (c2<='z'))
            c2 -= 0x20;
        if (c1=='\0')
            return ((c2=='\0') ? 0 : -1);
        if (c2=='\0')
            return 1;
        if (c1<c2)
            return -1;
        if (c1>c2)
            return 1;
    }
}
```

9.103.4.7 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().

```
{
    uLong uMagic,uData,uFlags;
    uLong size_filename;
    uLong size_extra_field;
    int err=UNZ_OK;

    *piSizeVar = 0;
    *poffset_local_extrafield = 0;
    *psize_local_extrafield = 0;

    if (ZSEEK64(s->z_filefunc, s->filestream,s->cur_file_info_internal.offset_curfile +
                                    s->byte_before_the_zipfile, ZLIB_FILEFUNC_SEEK_SET)!=0)
        return UNZ_ERRNO;


    if (err==UNZ_OK)
    {
        if (unz64local_getLong(&s->z_filefunc, s->filestream,&uMagic) != UNZ_OK)
            err=UNZ_ERRNO;
        else if (uMagic!=0x04034b50)
            err=UNZ_BADZIPFILE;
    }

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&uData) != UNZ_OK)
        err=UNZ_ERRNO;
/*
    else if ((err==UNZ_OK) && (uData!=s->cur_file_info.wVersion))
        err=UNZ_BADZIPFILE;
*/
    if (unz64local_getShort(&s->z_filefunc, s->filestream,&uFlags) != UNZ_OK)
        err=UNZ_ERRNO;

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&uData) != UNZ_OK)
        err=UNZ_ERRNO;
    else if ((err==UNZ_OK) && (uData!=s->cur_file_info.compression_method))
        err=UNZ_BADZIPFILE;

    if ((err==UNZ_OK) && (s->cur_file_info.compression_method!=0) &&
/* #ifdef HAVE_BZIP2 */
                        (s->cur_file_info.compression_method!=Z_BZIP2ED) &&
/* #endif */
                        (s->cur_file_info.compression_method!=Z_DEFLATED))
        err=UNZ_BADZIPFILE;

    if (unz64local_getLong(&s->z_filefunc, s->filestream,&uData) != UNZ_OK) /* date/time */
        err=UNZ_ERRNO;

    if (unz64local_getLong(&s->z_filefunc, s->filestream,&uData) != UNZ_OK) /* crc */
        err=UNZ_ERRNO;
    else if ((err==UNZ_OK) && (uData!=s->cur_file_info.crc) && ((uFlags & 8)==0))
        err=UNZ_BADZIPFILE;

    if (unz64local_getLong(&s->z_filefunc, s->filestream,&uData) != UNZ_OK) /* size compr */
        err=UNZ_ERRNO;
    else if (uData != 0xFFFFFFFF && (err==UNZ_OK) && (uData!=s->cur_file_info.compressed_size) && ((uFlags & 8)==0))
        err=UNZ_BADZIPFILE;

    if (unz64local_getLong(&s->z_filefunc, s->
```

```
filestream,&uData) != UNZ_OK) /* size uncompr */
        err=UNZ_ERRNO;
    else if (uData != 0xFFFFFFFF && (err==UNZ_OK) && (uData!=s->cur_file_info.uncompressed_size) && ((uFlags & 8)==0))
        err=UNZ_BADZIPFILE;

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&size_filename) != UNZ_OK)
        err=UNZ_ERRNO;
    else if ((err==UNZ_OK) && (size_filename!=s->cur_file_info.size_filename))
        err=UNZ_BADZIPFILE;

    *piSizeVar += (uInt)size_filename;

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&size_extra_field) != UNZ_OK)
        err=UNZ_ERRNO;
    *poffset_local_extrafield= s->cur_file_info_internal.offset_curfile +
                                    SIZEZIPLOCALHEADER + size_filename;
    *psize_local_extrafield = (uInt)size_extra_field;

    *piSizeVar += (uInt)size_extra_field;

    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.8 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().

```
{
    ZPOS64_T uDate;
```

```
    uDate = (ZPOS64_T)(ulDosDate>>16);
    ptm->tm_mday = (uInt)(uDate&0x1f) ;
    ptm->tm_mon =  (uInt)((((uDate)&0x1E0)/0x20)-1) ;
    ptm->tm_year = (uInt)(((uDate&0x0FE00)/0x0200)+1980) ;

    ptm->tm_hour = (uInt) ((ulDosDate &0xF800)/0x800);
    ptm->tm_min =  (uInt) ((ulDosDate&0x7E0)/0x20) ;
    ptm->tm_sec =  (uInt) (2*(ulDosDate&0x1f)) ;
}
```

Here is the caller graph for this function:

9.103.4.9 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().

```
{
    unsigned char c;
    int err = (int)ZREAD64(*pzlib_filefunc_def,filestream,&c,1);
    if (err==1)
    {
        *pi = (int)c;
        return UNZ_OK;
    }
    else
    {
        if (ZERROR64(*pzlib_filefunc_def,filestream))
            return UNZ_ERRNO;
        else
            return UNZ_EOF;
    }
}
```

Here is the caller graph for this function:

9.103.4.10 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().

```
{
    unz64_s* s;
    unz_file_info64 file_info;
    unz_file_info64_internal file_info_internal;
    int err=UNZ_OK;
    uLong uMagic;
    ZPOS64_T llSeek=0;
    uLong uL;

    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    if (ZSEEK64(s->z_filefunc, s->filestream,
              s->pos_in_central_dir+s->byte_before_the_zipfile,
              ZLIB_FILEFUNC_SEEK_SET)!=0)
        err=UNZ_ERRNO;


    /* we check the magic */
    if (err==UNZ_OK)
    {
        if (unz64local_getLong(&s->z_filefunc, s->filestream,&uMagic) != UNZ_OK)
            err=UNZ_ERRNO;
        else if (uMagic!=0x02014b50)
            err=UNZ_BADZIPFILE;
    }

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.version) != UNZ_OK)
        err=UNZ_ERRNO;

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.version_needed) != UNZ_OK)
        err=UNZ_ERRNO;

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.flag) != UNZ_OK)
        err=UNZ_ERRNO;

    if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.compression_method) != UNZ_OK)
        err=UNZ_ERRNO;

    if (unz64local_getLong(&s->z_filefunc, s->filestream,&file_info.dosDate) != UNZ_OK)
        err=UNZ_ERRNO;

    unz64local_DosDateToTmuDate(file_info.dosDate,&file_info.tmu_date);

    if (unz64local_getLong(&s->z_filefunc, s->filestream,&file_info.crc) != UNZ_OK)
        err=UNZ_ERRNO;

    if (unz64local_getLong(&s->z_filefunc, s->filestream,&uL) != UNZ_OK)
        err=UNZ_ERRNO;
    file_info.compressed_size = uL;

    if (unz64local_getLong(&s->z_filefunc, s->
```

```
filestream,&uL) != UNZ_OK)
        err=UNZ_ERRNO;
  file_info.uncompressed_size = uL;

  if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.size_filename) != UNZ_OK)
        err=UNZ_ERRNO;

  if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.size_file_extra) != UNZ_OK)
        err=UNZ_ERRNO;

  if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.size_file_comment) != UNZ_OK)
        err=UNZ_ERRNO;

  if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.disk_num_start) != UNZ_OK)
        err=UNZ_ERRNO;

  if (unz64local_getShort(&s->z_filefunc, s->filestream,&file_info.internal_fa) != UNZ_OK)
        err=UNZ_ERRNO;

  if (unz64local_getLong(&s->z_filefunc, s->filestream,&file_info.external_fa) != UNZ_OK)
        err=UNZ_ERRNO;

                /* relative offset of local header */
  if (unz64local_getLong(&s->z_filefunc, s->filestream,&uL) != UNZ_OK)
        err=UNZ_ERRNO;
  file_info_internal.offset_curfile = uL;

  llSeek+=file_info.size_filename;
  if ((err==UNZ_OK) && (szFileName!=NULL))
  {
      uLong uSizeRead ;
      if (file_info.size_filename<fileNameBufferSize)
      {
          *(szFileName+file_info.size_filename)='\0';
          uSizeRead = file_info.size_filename;
      }
      else
          uSizeRead = fileNameBufferSize;

      if ((file_info.size_filename>0) && (fileNameBufferSize>0))
          if (ZREAD64(s->z_filefunc, s->filestream,szFileName,uSizeRead)!=uSizeRead)
              err=UNZ_ERRNO;
      llSeek -= uSizeRead;
  }

  /* Read extrafield */
  if ((err==UNZ_OK) && (extraField!=NULL))
  {
      ZPOS64_T uSizeRead ;
      if (file_info.size_file_extra<extraFieldBufferSize)
          uSizeRead = file_info.size_file_extra;
       else
          uSizeRead = extraFieldBufferSize;

       if (llSeek!=0)
       {
          if (ZSEEK64(s->z_filefunc, s->filestream,llSeek,ZLIB_FILEFUNC_SEEK_CUR)==0)
              llSeek=0;
          else
              err=UNZ_ERRNO;
       }

       if ((file_info.size_file_extra>0) && (extraFieldBufferSize>0))
          if (ZREAD64(s->z_filefunc, s->filestream,extraField,(uLong)uSizeRead)!=uSizeRead)
              err=UNZ_ERRNO;

       llSeek += file_info.size_file_extra - (uLong)uSizeRead;
    }
    else
       llSeek += file_info.size_file_extra;


    if ((err==UNZ_OK) && (file_info.size_file_extra != 0))
    {
                                uLong acc = 0;

       /* since lSeek now points to after the extra field we need to move back */
```

```
        llSeek -= file_info.size_file_extra;

        if (llSeek!=0)
        {
            if (ZSEEK64(s->z_filefunc, s->filestream,llSeek,ZLIB_FILEFUNC_SEEK_CUR)==0)
                llSeek=0;
            else
                err=UNZ_ERRNO;
        }

        while(acc < file_info.size_file_extra)
        {
            uLong headerId;
                                                uLong dataSize;

            if (unz64local_getShort(&s->z_filefunc, s->filestream,&headerId) != UNZ_OK)
                err=UNZ_ERRNO;

            if (unz64local_getShort(&s->z_filefunc, s->filestream,&dataSize) != UNZ_OK)
                err=UNZ_ERRNO;

            /* ZIP64 extra fields */
            if (headerId == 0x0001)
            {
                uLong uL;

                if(file_info.uncompressed_size == (ZPOS64_T)0xFFFFFFFFu)
                {
                    if (unz64local_getLong64(&s->z_filefunc, s->filestream,&file_info.uncompressed_size) != UNZ_OK)
                        err=UNZ_ERRNO;
                }

                if(file_info.compressed_size == (ZPOS64_T)0xFFFFFFFFu)
                {
                    if (unz64local_getLong64(&s->z_filefunc, s->filestream,&file_info.compressed_size) != UNZ_OK)
                        err=UNZ_ERRNO;
                }

                if(file_info_internal.offset_curfile == (ZPOS64_T)0xFFFFFFFFu)
                {
                    /* Relative Header offset */
                    if (unz64local_getLong64(&s->z_filefunc, s->filestream,&file_info_internal.offset_curfile) != UNZ_OK)
                        err=UNZ_ERRNO;
                }

                if(file_info.disk_num_start == 0xFFFFFFFFu)
                {
                    /* Disk Start Number */
                    if (unz64local_getLong(&s->z_filefunc, s->filestream,&uL) != UNZ_OK)
                        err=UNZ_ERRNO;
                }

            }
            else
            {
                if (ZSEEK64(s->z_filefunc, s->filestream,dataSize,ZLIB_FILEFUNC_SEEK_CUR)!=0)
                    err=UNZ_ERRNO;
            }

            acc += 2 + 2 + dataSize;
        }
    }

    if ((err==UNZ_OK) && (szComment!=NULL))
    {
        uLong uSizeRead ;
        if (file_info.size_file_comment<commentBufferSize)
        {
            *(szComment+file_info.size_file_comment)='\0';
            uSizeRead = file_info.size_file_comment;
        }
        else
            uSizeRead = commentBufferSize;

        if (llSeek!=0)
        {
            if (ZSEEK64(s->z_filefunc, s->filestream,llSeek,ZLIB_FILEFUNC_SEEK_CUR)==0)
                llSeek=0;
```

```
            else
                err=UNZ_ERRNO;
        }

        if ((file_info.size_file_comment>0) && (commentBufferSize>0))
            if (ZREAD64(s->z_filefunc, s->filestream,szComment,uSizeRead)!=
    uSizeRead)
                err=UNZ_ERRNO;
        llSeek+=file_info.size_file_comment - uSizeRead;
    }
    else
        llSeek+=file_info.size_file_comment;


    if ((err==UNZ_OK) && (pfile_info!=NULL))
        *pfile_info=file_info;

    if ((err==UNZ_OK) && (pfile_info_internal!=NULL))
        *pfile_info_internal=file_info_internal;

    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.11 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(), and UNZ_OK.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir64(), and unzOpenInternal().

```
{
    uLong x ;
```

```
    int i = 0;
    int err;

    err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x = (uLong)i;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((uLong)i)<<8;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((uLong)i)<<16;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x += ((uLong)i)<<24;

    if (err==UNZ_OK)
        *pX = x;
    else
        *pX = 0;
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.12 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(), and UNZ_OK.

Referenced by unz64local_GetCurrentFileInfoInternal(), unz64local_SearchCentralDir64(), and unzOpenInternal().

```
{
    ZPOS64_T x ;
    int i = 0;
    int err;

    err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x = (ZPOS64_T)i;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<8;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<16;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<24;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<32;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<40;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<48;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((ZPOS64_T)i)<<56;

    if (err==UNZ_OK)
        *pX = x;
    else
        *pX = 0;
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.13 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(), and UNZ_OK.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), and unzOpenInternal().

```
{
    uLong x ;
    int i = 0;
    int err;

    err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x = (uLong)i;

    if (err==UNZ_OK)
        err = unz64local_getByte(pzlib_filefunc_def,filestream,&i);
    x |= ((uLong)i)<<8;

    if (err==UNZ_OK)
        *pX = x;
    else
        *pX = 0;
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.14 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().

```
{
    unsigned char* buf;
    ZPOS64_T uSizeFile;
    ZPOS64_T uBackRead;
    ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
    ZPOS64_T uPosFound=0;

    if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
        return 0;


    uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);

    if (uMaxBack>uSizeFile)
        uMaxBack = uSizeFile;

    buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
    if (buf==NULL)
        return 0;

    uBackRead = 4;
    while (uBackRead<uMaxBack)
    {
        uLong uReadSize;
        ZPOS64_T uReadPos ;
        int i;
        if (uBackRead+BUFREADCOMMENT>uMaxBack)
            uBackRead = uMaxBack;
        else
            uBackRead+=BUFREADCOMMENT;
        uReadPos = uSizeFile-uBackRead ;

        uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
                     (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
        if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
            break;

        if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
            break;

        for (i=(int)uReadSize-3; (i--)>0;)
            if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) &&
                ((*(buf+i+2))==0x05) && ((*(buf+i+3))==0x06))
            {
                uPosFound = uReadPos+i;
                break;
            }

        if (uPosFound!=0)
            break;
    }
    TRYFREE(buf);
    return uPosFound;
}
```

Here is the caller graph for this function:

9.103.4.15 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().

```
{
    unsigned char* buf;
    ZPOS64_T uSizeFile;
    ZPOS64_T uBackRead;
    ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
    ZPOS64_T uPosFound=0;
    uLong uL;
                ZPOS64_T relativeOffset;

    if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
        return 0;


    uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);

    if (uMaxBack>uSizeFile)
        uMaxBack = uSizeFile;

    buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
    if (buf==NULL)
        return 0;

    uBackRead = 4;
    while (uBackRead<uMaxBack)
    {
        uLong uReadSize;
        ZPOS64_T uReadPos;
        int i;
        if (uBackRead+BUFREADCOMMENT>uMaxBack)
            uBackRead = uMaxBack;
        else
            uBackRead+=BUFREADCOMMENT;
        uReadPos = uSizeFile-uBackRead ;

        uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
                     (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
        if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
            break;

        if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
            break;

        for (i=(int)uReadSize-3; (i--)>0;)
            if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) &&
                ((*(buf+i+2))==0x06) && ((*(buf+i+3))==0x07))
            {
                uPosFound = uReadPos+i;
                break;
            }

        if (uPosFound!=0)
            break;
    }
    TRYFREE(buf);
    if (uPosFound == 0)
        return 0;

    /* Zip64 end of central directory locator */
    if (ZSEEK64(*pzlib_filefunc_def,filestream, uPosFound,
    ZLIB_FILEFUNC_SEEK_SET)!=0)
        return 0;

    /* the signature, already checked */
    if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    UNZ_OK)
        return 0;

    /* number of the disk with the start of the zip64 end of  central directory */
    if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
    UNZ_OK)
        return 0;
    if (uL != 0)
        return 0;
```

```
    /* relative offset of the zip64 end of central directory record */
    if (unz64local_getLong64(pzlib_filefunc_def,filestream,&relativeOffset)!=UNZ_OK)
        return 0;

    /* total number of disks */
    if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=UNZ_OK)
        return 0;
    if (uL != 1)
        return 0;

    /* Goto end of central directory record */
    if (ZSEEK64(*pzlib_filefunc_def,filestream, relativeOffset,ZLIB_FILEFUNC_SEEK_SET)!=0)
        return 0;

     /* the signature */
    if (unz64local_getLong(pzlib_filefunc_def,filestream,&uL)!=UNZ_OK)
        return 0;

    if (uL != 0x06064b50)
        return 0;

    return relativeOffset;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.16 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().

```
{
    unz64_s* s;
    if (file == NULL)
        return UNZ_PARAMERROR;
    s = (unz64_s*)file;
    s->flags &= ~flags;
    return UNZ_OK;
}
```

Here is the caller graph for this function:

9.103.4.17 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().

```
{
    unz64_s* s;
    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;

    if (s->pfile_in_zip_read!=NULL)
        unzCloseCurrentFile(file);

    if ((s->flags & UNZ_AUTO_CLOSE) != 0)
        ZCLOSE64(s->z_filefunc, s->filestream);
    else
        ZFAKECLOSE64(s->z_filefunc, s->filestream);
    TRYFREE(s);
    return UNZ_OK;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.18 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().

```
{
    int err=UNZ_OK;

    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    pfile_in_zip_read_info=s->pfile_in_zip_read;

    if (pfile_in_zip_read_info==NULL)
        return UNZ_PARAMERROR;


    if ((pfile_in_zip_read_info->rest_read_uncompressed == 0) &&
        (!pfile_in_zip_read_info->raw))
    {
        if (pfile_in_zip_read_info->crc32 != pfile_in_zip_read_info->crc32_wait)
            err=UNZ_CRCERROR;
    }


    TRYFREE(pfile_in_zip_read_info->read_buffer);
    pfile_in_zip_read_info->read_buffer = NULL;
    if (pfile_in_zip_read_info->stream_initialised == Z_DEFLATED)
        inflateEnd(&pfile_in_zip_read_info->stream);
#ifdef HAVE_BZIP2
    else if (pfile_in_zip_read_info->stream_initialised == Z_BZIP2ED)
        BZ2_bzDecompressEnd(&pfile_in_zip_read_info->bstream);
#endif


    pfile_in_zip_read_info->stream_initialised = 0;
    TRYFREE(pfile_in_zip_read_info);

    s->pfile_in_zip_read=NULL;

    return err;
}
```

Here is the caller graph for this function:

9.103.4.19 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:

9.103.4.20 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.

```
{
    int err;
    unz_file_info64 file_info64;
    err = unz64local_GetCurrentFileInfoInternal(file,&file_info64,NULL,
                                                szFileName,fileNameBufferSize,
                                                extraField,extraFieldBufferSize,
                                                szComment,commentBufferSize);
    if (err==UNZ_OK && pfile_info != NULL)
    {
        pfile_info->version = file_info64.version;
        pfile_info->version_needed = file_info64.version_needed;
        pfile_info->flag = file_info64.flag;
        pfile_info->compression_method = file_info64.compression_method;
        pfile_info->dosDate = file_info64.dosDate;
        pfile_info->crc = file_info64.crc;

        pfile_info->size_filename = file_info64.size_filename;
        pfile_info->size_file_extra = file_info64.size_file_extra;
        pfile_info->size_file_comment = file_info64.size_file_comment;

        pfile_info->disk_num_start = file_info64.disk_num_start;
        pfile_info->internal_fa = file_info64.internal_fa;
        pfile_info->external_fa = file_info64.external_fa;

        pfile_info->tmu_date = file_info64.tmu_date,


        pfile_info->compressed_size = (uLong)file_info64.compressed_size;
        pfile_info->uncompressed_size = (uLong)file_info64.uncompressed_size;

    }
    return err;
}
```

Here is the call graph for this function:

9.103.4.21 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().

```
{
    return unz64local_GetCurrentFileInfoInternal(file,pfile_info,NULL,
                                                 szFileName,fileNameBufferSize,
                                                 extraField,extraFieldBufferSize,
                                                 szComment,commentBufferSize);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.22 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.

```
{
    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    s=(unz64_s*)file;
    if (file==NULL)
        return 0; /*UNZ_PARAMERROR; */
    pfile_in_zip_read_info=s->pfile_in_zip_read;
    if (pfile_in_zip_read_info==NULL)
        return 0; /*UNZ_PARAMERROR; */
    return pfile_in_zip_read_info->pos_in_zipfile +
                         pfile_in_zip_read_info->byte_before_the_zipfile;
}
```

9.103.4.23 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().

```
{
    unz64_file_pos file_pos64;
    int err = unzGetFilePos64(file,&file_pos64);
    if (err==UNZ_OK)
    {
```

```
        file_pos->pos_in_zip_directory = (uLong)file_pos64.pos_in_zip_directory;
        file_pos->num_of_file = (uLong)file_pos64.num_of_file;
    }
    return err;
}
```

Here is the call graph for this function:

9.103.4.24 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().

```
{
    unz64_s* s;

    if (file==NULL || file_pos==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    if (!s->current_file_ok)
        return UNZ_END_OF_LIST_OF_FILE;

    file_pos->pos_in_zip_directory  = s->pos_in_central_dir;
    file_pos->num_of_file           = s->num_file;

    return UNZ_OK;
}
```

Here is the caller graph for this function:

9.103.4.25 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().

```
{
    unz64_s* s;
    uLong uReadThis ;
    if (file==NULL)
        return (int)UNZ_PARAMERROR;
    s=(unz64_s*)file;

    uReadThis = uSizeBuf;
    if (uReadThis>s->gi.size_comment)
        uReadThis = s->gi.size_comment;

    if (ZSEEK64(s->z_filefunc,s->filestream,s->central_pos+22,ZLIB_FILEFUNC_SEEK_SET)!=0)
        return UNZ_ERRNO;

    if (uReadThis>0)
    {
      *szComment='\0';
      if (ZREAD64(s->z_filefunc,s->filestream,szComment,uReadThis)!=uReadThis)
        return UNZ_ERRNO;
    }

    if ((szComment != NULL) && (uSizeBuf > s->gi.size_comment))
        *(szComment+s->gi.size_comment)='\0';
    return (int)uReadThis;
}
```

Here is the caller graph for this function:

9.103.4.26 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.

```
{
    unz64_s* s;
    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    /* to do : check if number_entry is not truncated */
    pglobal_info32->number_entry = (uLong)s->gi.number_entry;
    pglobal_info32->size_comment = s->gi.size_comment;
    return UNZ_OK;
}
```

9.103.4.27 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().

```
{
    unz64_s* s;
    if (file==NULL)
```

```
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    *pglobal_info=s->gi;
    return UNZ_OK;
}
```

Here is the caller graph for this function:

9.103.4.28 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.

```
{
    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    uInt read_now;
    ZPOS64_T size_to_read;

    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    pfile_in_zip_read_info=s->pfile_in_zip_read;

    if (pfile_in_zip_read_info==NULL)
        return UNZ_PARAMERROR;

    size_to_read = (pfile_in_zip_read_info->size_local_extrafield -
                pfile_in_zip_read_info->pos_local_extrafield);

    if (buf==NULL)
        return (int)size_to_read;

    if (len>size_to_read)
        read_now = (uInt)size_to_read;
    else
        read_now = (uInt)len ;

    if (read_now==0)
        return 0;

    if (ZSEEK64(pfile_in_zip_read_info->z_filefunc,
              pfile_in_zip_read_info->filestream,
              pfile_in_zip_read_info->offset_local_extrafield +
              pfile_in_zip_read_info->pos_local_extrafield,
              ZLIB_FILEFUNC_SEEK_SET)!=0)
        return UNZ_ERRNO;

    if (ZREAD64(pfile_in_zip_read_info->z_filefunc,
              pfile_in_zip_read_info->filestream,
              buf,read_now)!=read_now)
        return UNZ_ERRNO;

    return (int)read_now;
}
```

9.103.4.29 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:

9.103.4.30 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:

9.103.4.31 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().

```
{
    unz64_file_pos file_pos64;
    if (file_pos == NULL)
        return UNZ_PARAMERROR;

    file_pos64.pos_in_zip_directory = file_pos->pos_in_zip_directory;
    file_pos64.num_of_file = file_pos->num_of_file;
    return unzGoToFilePos64(file,&file_pos64);
}
```

Here is the call graph for this function:

9.103.4.32 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().

```
{
    unz64_s* s;
    int err;

    if (file==NULL || file_pos==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;

    /* jump to the right spot */
    s->pos_in_central_dir = file_pos->pos_in_zip_directory;
    s->num_file           = file_pos->num_of_file;

    /* set the current file */
    err = unz64local_GetCurrentFileInfoInternal(file,&s->cur_file_info,
                                                &s->cur_file_info_internal,
                                                NULL,0,NULL,0,NULL,0);
    /* return results */
    s->current_file_ok = (err == UNZ_OK);
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.33 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().

```
{
    int err=UNZ_OK;
    unz64_s* s;
    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    s->pos_in_central_dir=s->offset_central_dir;
    s->num_file=0;
    err=unz64local_GetCurrentFileInfoInternal(file,&s->cur_file_info,
                                              &s->cur_file_info_internal,
                                              NULL,0,NULL,0,NULL,0);
    s->current_file_ok = (err == UNZ_OK);
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.34 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().

```
{
    unz64_s* s;
    int err;

    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    if (!s->current_file_ok)
        return UNZ_END_OF_LIST_OF_FILE;
    if (s->gi.number_entry != 0xffff)    /* 2^16 files overflow hack */
      if (s->num_file+1==s->gi.number_entry)
        return UNZ_END_OF_LIST_OF_FILE;

    s->pos_in_central_dir += SIZECENTRALDIRITEM + s->cur_file_info.size_filename +
            s->cur_file_info.size_file_extra + s->cur_file_info.size_file_comment ;
    s->num_file++;
    err = unz64local_GetCurrentFileInfoInternal(file,&s->cur_file_info,
                                                &s->cur_file_info_internal,
                                                NULL,0,NULL,0,NULL,0);
    s->current_file_ok = (err == UNZ_OK);
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.35 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().

```
{
    unz64_s* s;
    int err;

    /* We remember the 'current' position in the file so that we can jump
     * back there if we fail.
     */
    unz_file_info64 cur_file_infoSaved;
    unz_file_info64_internal cur_file_info_internalSaved;
    ZPOS64_T num_fileSaved;
    ZPOS64_T pos_in_central_dirSaved;


    if (file==NULL)
        return UNZ_PARAMERROR;

    if (strlen(szFileName)>=UNZ_MAXFILENAMEINZIP)
        return UNZ_PARAMERROR;

    s=(unz64_s*)file;
    if (!s->current_file_ok)
        return UNZ_END_OF_LIST_OF_FILE;

    /* Save the current state */
    num_fileSaved = s->num_file;
    pos_in_central_dirSaved = s->pos_in_central_dir;
    cur_file_infoSaved = s->cur_file_info;
    cur_file_info_internalSaved = s->cur_file_info_internal;

    err = unzGoToFirstFile(file);

    while (err == UNZ_OK)
    {
        char szCurrentFileName[UNZ_MAXFILENAMEINZIP+1];
        err = unzGetCurrentFileInfo64(file,NULL,
                                      szCurrentFileName,sizeof(szCurrentFileName)-1,
                                      NULL,0,NULL,0);
        if (err == UNZ_OK)
        {
            if (unzStringFileNameCompare(szCurrentFileName,
                                         szFileName,iCaseSensitivity)==0)
                return UNZ_OK;
            err = unzGoToNextFile(file);
        }
    }

    /* We failed, so restore the state of the 'current file' to where we
     * were.
     */
    s->num_file = num_fileSaved ;
    s->pos_in_central_dir = pos_in_central_dirSaved ;
    s->cur_file_info = cur_file_infoSaved;
    s->cur_file_info_internal = cur_file_info_internalSaved;
    return err;
}
```

Here is the call graph for this function:

9.103.4.36 unzFile ZEXPORT unzOpen (voidpf *file*)

Definition at line 800 of file unzip.c.

References UNZ_DEFAULT_FLAGS, and unzOpenInternal().

```
{
    return unzOpenInternal(file, NULL, 0, UNZ_DEFAULT_FLAGS);
}
```

Here is the call graph for this function:

9.103.4.37 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().

```
{
    if (pzlib_filefunc32_def != NULL)
    {
        zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
        fill_zlib_filefunc64_32_def_from_filefunc32(&
```

```
zlib_filefunc64_32_def_fill,pzlib_filefunc32_def);
        return unzOpenInternal(file, &zlib_filefunc64_32_def_fill, 0,
UNZ_DEFAULT_FLAGS);
    }
    else
        return unzOpenInternal(file, NULL, 0, UNZ_DEFAULT_FLAGS);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.38 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.

```
{
    if (pzlib_filefunc_def != NULL)
    {
        zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
        zlib_filefunc64_32_def_fill.zfile_func64 = *pzlib_filefunc_def;
        zlib_filefunc64_32_def_fill.ztell32_file = NULL;
        zlib_filefunc64_32_def_fill.zseek32_file = NULL;
        return unzOpenInternal(file, &zlib_filefunc64_32_def_fill, 1,
UNZ_DEFAULT_FLAGS);
    }
    else
```

```
        return unzOpenInternal(file, NULL, 1, UNZ_DEFAULT_FLAGS);
}
```

Here is the call graph for this function:

9.103.4.39 unzFile ZEXPORT unzOpen64 (voidpf *file*)

Definition at line 805 of file unzip.c.

References UNZ_DEFAULT_FLAGS, and unzOpenInternal().

```
{
    return unzOpenInternal(file, NULL, 1, UNZ_DEFAULT_FLAGS);
}
```

Here is the call graph for this function:

9.103.4.40 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:

9.103.4.41 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:

9.103.4.42 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().

```
{
    int err=UNZ_OK;
    uInt iSizeVar;
    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    ZPOS64_T offset_local_extrafield;  /* offset of the local extra field */
    uInt  size_local_extrafield;    /* size of the local extra field */
#    ifndef NOUNCRYPT
    char source[12];
#    else
    if (password != NULL)
        return UNZ_PARAMERROR;
#    endif

    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    if (!s->current_file_ok)
        return UNZ_PARAMERROR;

    if (s->pfile_in_zip_read != NULL)
        unzCloseCurrentFile(file);

    if (unz64local_CheckCurrentFileCoherencyHeader(s,&iSizeVar, &offset_local_extrafield,&size_local_extrafield)!=UNZ_OK)
        return UNZ_BADZIPFILE;

    pfile_in_zip_read_info = (file_in_zip64_read_info_s*)ALLOC(sizeof(file_in_zip64_read_info_s));
    if (pfile_in_zip_read_info==NULL)
        return UNZ_INTERNALERROR;

    pfile_in_zip_read_info->read_buffer=(char*)ALLOC(UNZ_BUFSIZE);
    pfile_in_zip_read_info->offset_local_extrafield = offset_local_extrafield;
    pfile_in_zip_read_info->size_local_extrafield = size_local_extrafield;
    pfile_in_zip_read_info->pos_local_extrafield=0;
    pfile_in_zip_read_info->raw=raw;

    if (pfile_in_zip_read_info->read_buffer==NULL)
    {
        TRYFREE(pfile_in_zip_read_info);
        return UNZ_INTERNALERROR;
    }

    pfile_in_zip_read_info->stream_initialised=0;

    if (method!=NULL)
        *method = (int)s->cur_file_info.compression_method;

    if (level!=NULL)
    {
        *level = 6;
        switch (s->cur_file_info.flag & 0x06)
        {
          case 6 : *level = 1; break;
          case 4 : *level = 2; break;
          case 2 : *level = 9; break;
        }
    }

    if ((s->cur_file_info.compression_method!=0) &&
/* #ifdef HAVE_BZIP2 */
        (s->cur_file_info.compression_method!=Z_BZIP2ED) &&
/* #endif */
        (s->cur_file_info.compression_method!=Z_DEFLATED))

        err=UNZ_BADZIPFILE;

    pfile_in_zip_read_info->crc32_wait=s->cur_file_info.crc;
    pfile_in_zip_read_info->crc32=0;
    pfile_in_zip_read_info->total_out_64=0;
    pfile_in_zip_read_info->compression_method = s->cur_file_info.compression_method;
```

```
    pfile_in_zip_read_info->filestream=s->filestream;
    pfile_in_zip_read_info->z_filefunc=s->z_filefunc;
    pfile_in_zip_read_info->byte_before_the_zipfile=s->byte_before_the_zipfile;

    pfile_in_zip_read_info->stream.total_out = 0;

    if ((s->cur_file_info.compression_method==Z_BZIP2ED) && (!raw))
    {
#ifdef HAVE_BZIP2
      pfile_in_zip_read_info->bstream.bzalloc = (void *(*) (void *, int, int))0;
      pfile_in_zip_read_info->bstream.bzfree = (free_func)0;
      pfile_in_zip_read_info->bstream.opaque = (voidpf)0;
      pfile_in_zip_read_info->bstream.state = (voidpf)0;

      pfile_in_zip_read_info->stream.zalloc = (alloc_func)0;
      pfile_in_zip_read_info->stream.zfree = (free_func)0;
      pfile_in_zip_read_info->stream.opaque = (voidpf)0;
      pfile_in_zip_read_info->stream.next_in = (voidpf)0;
      pfile_in_zip_read_info->stream.avail_in = 0;

      err=BZ2_bzDecompressInit(&pfile_in_zip_read_info->bstream, 0, 0);
      if (err == Z_OK)
        pfile_in_zip_read_info->stream_initialised=Z_BZIP2ED;
      else
      {
        TRYFREE(pfile_in_zip_read_info);
        return err;
      }
#else
      pfile_in_zip_read_info->raw=1;
#endif
    }
    else if ((s->cur_file_info.compression_method==Z_DEFLATED) && (!raw))
    {
      pfile_in_zip_read_info->stream.zalloc = (alloc_func)0;
      pfile_in_zip_read_info->stream.zfree = (free_func)0;
      pfile_in_zip_read_info->stream.opaque = (voidpf)0;
      pfile_in_zip_read_info->stream.next_in = 0;
      pfile_in_zip_read_info->stream.avail_in = 0;

      err=inflateInit2(&pfile_in_zip_read_info->stream, -MAX_WBITS);
      if (err == Z_OK)
        pfile_in_zip_read_info->stream_initialised=Z_DEFLATED;
      else
      {
        TRYFREE(pfile_in_zip_read_info);
        return err;
      }
        /* windowBits is passed < 0 to tell that there is no zlib header.
         * Note that in this case inflate *requires* an extra "dummy" byte
         * after the compressed stream in order to complete decompression and
         * return Z_STREAM_END.
         * In unzip, i don't wait absolutely Z_STREAM_END because I known the
         * size of both compressed and uncompressed data
         */
    }
    pfile_in_zip_read_info->rest_read_compressed =
            s->cur_file_info.compressed_size ;
    pfile_in_zip_read_info->rest_read_uncompressed =
            s->cur_file_info.uncompressed_size ;


    pfile_in_zip_read_info->pos_in_zipfile =
            s->cur_file_info_internal.offset_curfile + SIZEZIPLOCALHEADER +
              iSizeVar;

    pfile_in_zip_read_info->stream.avail_in = (uInt)0;

    s->pfile_in_zip_read = pfile_in_zip_read_info;
                s->encrypted = 0;

#    ifndef NOUNCRYPT
    if (password != NULL)
    {
        int i;
        s->pcrc_32_tab = get_crc_table();
        init_keys(password,s->keys,s->pcrc_32_tab);
        if (ZSEEK64(s->z_filefunc, s->filestream,
                  s->pfile_in_zip_read->pos_in_zipfile +
                     s->pfile_in_zip_read->byte_before_the_zipfile,
                  SEEK_SET)!=0)
            return UNZ_INTERNALERROR;
        if(ZREAD64(s->z_filefunc, s->filestream,source, 12)<12)
```

```
            return UNZ_INTERNALERROR;

        for (i = 0; i<12; i++)
            zdecode(s->keys,s->pcrc_32_tab,source[i]);

        s->pfile_in_zip_read->pos_in_zipfile+=12;
        s->encrypted=1;
    }
#    endif


    return UNZ_OK;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.43 int ZEXPORT unzOpenCurrentFilePassword (unzFile *file*, const char * *password*)

Definition at line 1654 of file unzip.c.

References unzOpenCurrentFile3().

```
{
    return unzOpenCurrentFile3(file, NULL, NULL, 0, password);
}
```

Here is the call graph for this function:

9.103.4.44 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().

```
{
    unz64_s us;
    unz64_s *s;
    ZPOS64_T central_pos;
    uLong   uL;

    uLong number_disk;          /* number of the current dist, used for
                                   spaning ZIP, unsupported, always 0*/
    uLong number_disk_with_CD;  /* number the the disk with central dir, used
                                   for spaning ZIP, unsupported, always 0*/
    ZPOS64_T number_entry_CD;      /* total number of entries in
                                   the central dir
                                   (same than number_entry on nospan) */

    int err=UNZ_OK;

    if (unz_copyright[0]!=' ')
        return NULL;

    us.flags = flags;
    us.z_filefunc.zseek32_file = NULL;
    us.z_filefunc.ztell32_file = NULL;
    if (pzlib_filefunc64_32_def==NULL)
        fill_qiodevice64_filefunc(&us.z_filefunc.
zfile_func64);
    else
        us.z_filefunc = *pzlib_filefunc64_32_def;
    us.is64bitOpenFunction = is64bitOpenFunction;



    us.filestream = ZOPEN64(us.z_filefunc,
                                                file,
                                                ZLIB_FILEFUNC_MODE_READ |
                                                ZLIB_FILEFUNC_MODE_EXISTING);
    if (us.filestream==NULL)
        return NULL;

    central_pos = unz64local_SearchCentralDir64(&us.
z_filefunc,us.filestream);
    if (central_pos)
    {
        uLong uS;
        ZPOS64_T uL64;

```

```
        us.isZip64 = 1;

        if (ZSEEK64(us.z_filefunc, us.filestream,
                                    central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
        err=UNZ_ERRNO;

        /* the signature, already checked */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* size of zip64 end of central directory record */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&uL64)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* version made by */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uS)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* version needed to extract */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uS)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* number of this disk */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&number_disk)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* number of the disk with the start of the central directory */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&number_disk_with_CD)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* total number of entries in the central directory on this disk */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&us.gi.number_entry)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* total number of entries in the central directory */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&number_entry_CD)!=UNZ_OK)
            err=UNZ_ERRNO;

        if ((number_entry_CD!=us.gi.number_entry) ||
            (number_disk_with_CD!=0) ||
            (number_disk!=0))
            err=UNZ_BADZIPFILE;

        /* size of the central directory */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&us.size_central_dir)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* offset of start of central directory with respect to the
          starting disk number */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&us.offset_central_dir)!=UNZ_OK)
            err=UNZ_ERRNO;

        us.gi.size_comment = 0;
    }
    else
    {
        central_pos = unz64local_SearchCentralDir(&us.z_filefunc,us.filestream);
        if (central_pos==0)
            err=UNZ_ERRNO;

        us.isZip64 = 0;

        if (ZSEEK64(us.z_filefunc, us.filestream,
                                    central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
            err=UNZ_ERRNO;

        /* the signature, already checked */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* number of this disk */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&number_disk)!=UNZ_OK)
            err=UNZ_ERRNO;

```

```
        /* number of the disk with the start of the central directory */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&number_disk_with_CD)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* total number of entries in the central dir on this disk */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        us.gi.number_entry = uL;

        /* total number of entries in the central dir */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        number_entry_CD = uL;

        if ((number_entry_CD!=us.gi.number_entry) ||
            (number_disk_with_CD!=0) ||
            (number_disk!=0))
            err=UNZ_BADZIPFILE;

        /* size of the central directory */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        us.size_central_dir = uL;

        /* offset of start of central directory with respect to the
            starting disk number */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        us.offset_central_dir = uL;

        /* zipfile comment length */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&us.gi.size_comment)!=UNZ_OK)
            err=UNZ_ERRNO;
    }

    if ((central_pos<us.offset_central_dir+us.size_central_dir) &&
        (err==UNZ_OK))
        err=UNZ_BADZIPFILE;

    if (err!=UNZ_OK)
    {
        if ((us.flags & UNZ_AUTO_CLOSE) != 0)
            ZCLOSE64(us.z_filefunc, us.filestream);
        else
            ZFAKECLOSE64(us.z_filefunc, us.filestream);
        return NULL;
    }

    us.byte_before_the_zipfile = central_pos -
                            (us.offset_central_dir+us.size_central_dir);
    us.central_pos = central_pos;
    us.pfile_in_zip_read = NULL;
    us.encrypted = 0;


    s=(unz64_s*)ALLOC(sizeof(unz64_s));
    if( s != NULL)
    {
        *s=us;
        unzGoToFirstFile((unzFile)s);
    }
    return (unzFile)s;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.103.4.45 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().

```
{
    int err=UNZ_OK;
    uInt iRead = 0;
    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    if (file==NULL)
```

```
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    pfile_in_zip_read_info=s->pfile_in_zip_read;

    if (pfile_in_zip_read_info==NULL)
        return UNZ_PARAMERROR;


    if (pfile_in_zip_read_info->read_buffer == NULL)
        return UNZ_END_OF_LIST_OF_FILE;
    if (len==0)
        return 0;

    pfile_in_zip_read_info->stream.next_out = (Bytef*)buf;

    pfile_in_zip_read_info->stream.avail_out = (uInt)len;

    if ((len>pfile_in_zip_read_info->rest_read_uncompressed) &&
        (!(pfile_in_zip_read_info->raw)))
        pfile_in_zip_read_info->stream.avail_out =
            (uInt)pfile_in_zip_read_info->rest_read_uncompressed;

    if ((len>pfile_in_zip_read_info->rest_read_compressed+
           pfile_in_zip_read_info->stream.avail_in) &&
         (pfile_in_zip_read_info->raw))
        pfile_in_zip_read_info->stream.avail_out =
            (uInt)pfile_in_zip_read_info->rest_read_compressed+
            pfile_in_zip_read_info->stream.avail_in;

    while (pfile_in_zip_read_info->stream.avail_out>0)
    {
        if ((pfile_in_zip_read_info->stream.avail_in==0) &&
            (pfile_in_zip_read_info->rest_read_compressed>0))
        {
            uInt uReadThis = UNZ_BUFSIZE;
            if (pfile_in_zip_read_info->rest_read_compressed<uReadThis)
                uReadThis = (uInt)pfile_in_zip_read_info->rest_read_compressed;
            if (uReadThis == 0)
                return UNZ_EOF;
            if (ZSEEK64(pfile_in_zip_read_info->z_filefunc,
                      pfile_in_zip_read_info->filestream,
                      pfile_in_zip_read_info->pos_in_zipfile +
                         pfile_in_zip_read_info->byte_before_the_zipfile,
                         ZLIB_FILEFUNC_SEEK_SET)!=0)
                return UNZ_ERRNO;
            if (ZREAD64(pfile_in_zip_read_info->z_filefunc,
                      pfile_in_zip_read_info->filestream,
                      pfile_in_zip_read_info->read_buffer,
                      uReadThis)!=uReadThis)
                return UNZ_ERRNO;


#            ifndef NOUNCRYPT
            if(s->encrypted)
            {
                uInt i;
                for(i=0;i<uReadThis;i++)
                  pfile_in_zip_read_info->read_buffer[i] =
                      zdecode(s->keys,s->pcrc_32_tab,
                              pfile_in_zip_read_info->read_buffer[i]);
            }
#            endif


            pfile_in_zip_read_info->pos_in_zipfile += uReadThis;

            pfile_in_zip_read_info->rest_read_compressed-=uReadThis;

            pfile_in_zip_read_info->stream.next_in =
                (Bytef*)pfile_in_zip_read_info->read_buffer;
            pfile_in_zip_read_info->stream.avail_in = (uInt)uReadThis;
        }

        if ((pfile_in_zip_read_info->compression_method==0) || (pfile_in_zip_read_info->
raw))
        {
            uInt uDoCopy,i ;

            if ((pfile_in_zip_read_info->stream.avail_in == 0) &&
                (pfile_in_zip_read_info->rest_read_compressed == 0))
                return (iRead==0) ? UNZ_EOF : iRead;

            if (pfile_in_zip_read_info->stream.avail_out <
                            pfile_in_zip_read_info->stream.avail_in)
                uDoCopy = pfile_in_zip_read_info->stream.avail_out ;
            else
                uDoCopy = pfile_in_zip_read_info->stream.avail_in ;
```

```

            for (i=0;i<uDoCopy;i++)
                *(pfile_in_zip_read_info->stream.next_out+i) =
                        *(pfile_in_zip_read_info->stream.next_in+i);

            pfile_in_zip_read_info->total_out_64 = pfile_in_zip_read_info->total_out_64 + uDoCopy;

            pfile_in_zip_read_info->crc32 = crc32(pfile_in_zip_read_info->crc32,
                                  pfile_in_zip_read_info->stream.next_out,
                                  uDoCopy);
            pfile_in_zip_read_info->rest_read_uncompressed-=uDoCopy;
            pfile_in_zip_read_info->stream.avail_in -= uDoCopy;
            pfile_in_zip_read_info->stream.avail_out -= uDoCopy;
            pfile_in_zip_read_info->stream.next_out += uDoCopy;
            pfile_in_zip_read_info->stream.next_in += uDoCopy;
            pfile_in_zip_read_info->stream.total_out += uDoCopy;
            iRead += uDoCopy;
        }
        else if (pfile_in_zip_read_info->compression_method==Z_BZIP2ED)
        {
#ifdef HAVE_BZIP2
            uLong uTotalOutBefore,uTotalOutAfter;
            const Bytef *bufBefore;
            uLong uOutThis;

            pfile_in_zip_read_info->bstream.next_in        = (char*)pfile_in_zip_read_info->stream.next_in;
            pfile_in_zip_read_info->bstream.avail_in       = pfile_in_zip_read_info->stream.avail_in;
            pfile_in_zip_read_info->bstream.total_in_lo32  = pfile_in_zip_read_info->stream.total_in;
            pfile_in_zip_read_info->bstream.total_in_hi32  = 0;
            pfile_in_zip_read_info->bstream.next_out       = (char*)pfile_in_zip_read_info->stream.next_out;
            pfile_in_zip_read_info->bstream.avail_out      = pfile_in_zip_read_info->stream.avail_out;
            pfile_in_zip_read_info->bstream.total_out_lo32 = pfile_in_zip_read_info->stream.total_out;
            pfile_in_zip_read_info->bstream.total_out_hi32 = 0;

            uTotalOutBefore = pfile_in_zip_read_info->bstream.total_out_lo32;
            bufBefore = (const Bytef *)pfile_in_zip_read_info->bstream.next_out;

            err=BZ2_bzDecompress(&pfile_in_zip_read_info->bstream);

            uTotalOutAfter = pfile_in_zip_read_info->bstream.total_out_lo32;
            uOutThis = uTotalOutAfter-uTotalOutBefore;

            pfile_in_zip_read_info->total_out_64 = pfile_in_zip_read_info->total_out_64 + uOutThis;

            pfile_in_zip_read_info->crc32 = crc32(pfile_in_zip_read_info->crc32,bufBefore, (uInt)(uOutThis));
            pfile_in_zip_read_info->rest_read_uncompressed -= uOutThis;
            iRead += (uInt)(uTotalOutAfter - uTotalOutBefore);

            pfile_in_zip_read_info->stream.next_in   = (Bytef*)pfile_in_zip_read_info->bstream.next_in;
            pfile_in_zip_read_info->stream.avail_in  = pfile_in_zip_read_info->bstream.avail_in;
            pfile_in_zip_read_info->stream.total_in  = pfile_in_zip_read_info->bstream.total_in_lo32;
            pfile_in_zip_read_info->stream.next_out  = (Bytef*)pfile_in_zip_read_info->bstream.next_out;
            pfile_in_zip_read_info->stream.avail_out = pfile_in_zip_read_info->bstream.avail_out;
            pfile_in_zip_read_info->stream.total_out = pfile_in_zip_read_info->bstream.total_out_lo32;

            if (err==BZ_STREAM_END)
              return (iRead==0) ? UNZ_EOF : iRead;
            if (err!=BZ_OK)
              break;
#endif
        } /* end Z_BZIP2ED */
        else
        {
            ZPOS64_T uTotalOutBefore,uTotalOutAfter;
            const Bytef *bufBefore;
            ZPOS64_T uOutThis;
            int flush=Z_SYNC_FLUSH;

            uTotalOutBefore = pfile_in_zip_read_info->stream.total_out;
            bufBefore = pfile_in_zip_read_info->stream.next_out;

            /*
            if ((pfile_in_zip_read_info->rest_read_uncompressed ==
```

```
                    pfile_in_zip_read_info->stream.avail_out) &&
                (pfile_in_zip_read_info->rest_read_compressed == 0))
                flush = Z_FINISH;
            */
            err=inflate(&pfile_in_zip_read_info->stream,flush);

            if ((err>=0) && (pfile_in_zip_read_info->stream.msg!=NULL))
              err = Z_DATA_ERROR;

            uTotalOutAfter = pfile_in_zip_read_info->stream.total_out;
            uOutThis = uTotalOutAfter-uTotalOutBefore;

            pfile_in_zip_read_info->total_out_64 = pfile_in_zip_read_info->total_out_64 + uOutThis;

            pfile_in_zip_read_info->crc32 =
                crc32(pfile_in_zip_read_info->crc32,bufBefore,
                        (uInt)(uOutThis));

            pfile_in_zip_read_info->rest_read_uncompressed -=
                uOutThis;

            iRead += (uInt)(uTotalOutAfter - uTotalOutBefore);

            if (err==Z_STREAM_END)
                return (iRead==0) ? UNZ_EOF : iRead;
            if (err!=Z_OK)
                break;
        }
    }

    if (err==Z_OK)
        return iRead;
    return err;
}
```

Here is the caller graph for this function:

9.103.4.46 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().

```
{
    unz64_s* s;
    if (file == NULL)
        return UNZ_PARAMERROR;
    s = (unz64_s*)file;
    s->flags |= flags;
    return UNZ_OK;
}
```

Here is the caller graph for this function:

9.103.4.47 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:

9.103.4.48 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:

9.103.4.49 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:

9.103.4.50 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().

```
{
    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    if (file==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    pfile_in_zip_read_info=s->pfile_in_zip_read;

    if (pfile_in_zip_read_info==NULL)
        return UNZ_PARAMERROR;

    return (z_off_t)pfile_in_zip_read_info->stream.total_out;
}
```

Here is the caller graph for this function:

9.103.4.51 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.

```
{

    unz64_s* s;
    file_in_zip64_read_info_s* pfile_in_zip_read_info;
    if (file==NULL)
        return (ZPOS64_T)-1;
    s=(unz64_s*)file;
    pfile_in_zip_read_info=s->pfile_in_zip_read;

    if (pfile_in_zip_read_info==NULL)
        return (ZPOS64_T)-1;

    return pfile_in_zip_read_info->total_out_64;
}
```

9.103.5 Variable Documentation

9.103.5.1 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().

9.104 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)

9.104.1 Data Structure Documentation

9.104.1.1 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	

9.104.1.2 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	

9.104.1.3 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	

9.104.1.4 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	

9.104.1.5 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	

9.104.1.6 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	

9.104.1.7 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	

9.104.2 Macro Definition Documentation

9.104.2.1 #define UNZ_AUTO_CLOSE 0x01u

Definition at line 88 of file unzip.h.

Referenced by QuaZip::open(), unzClose(), and unzOpenInternal().

9.104.2.2 #define UNZ_BADZIPFILE (-103)

Definition at line 84 of file unzip.h.

Referenced by unz64local_CheckCurrentFileCoherencyHeader(), unz64local_GetCurrentFileInfoInternal(), unzOpenCurrentFile3(), and unzOpenInternal().

9.104.2.3 #define UNZ_CRCERROR (-105)

Definition at line 86 of file unzip.h.

Referenced by unzCloseCurrentFile().

9.104.2.4 #define UNZ_DEFAULT_FLAGS UNZ_AUTO_CLOSE

Definition at line 89 of file unzip.h.

Referenced by unzOpen(), unzOpen2(), unzOpen2_64(), and unzOpen64().

9.104.2.5 #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().

9.104.2.6 #define UNZ_EOF (0)

Definition at line 82 of file unzip.h.

Referenced by unz64local_getByte(), and unzReadCurrentFile().

9.104.2.7 #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().

9.104.2.8 #define UNZ_INTERNALERROR (-104)

Definition at line 85 of file unzip.h.

Referenced by unzOpenCurrentFile3().

9.104.2.9 #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().

9.104.2.10 #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().

9.104.2.11 #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().

9.104.3 Typedef Documentation

9.104.3.1 typedef struct tm_unz_s tm_unz

9.104.3.2 typedef struct unz64_file_pos_s unz64_file_pos

9.104.3.3 typedef struct unz_file_info_s unz_file_info

9.104.3.4 typedef struct unz_file_info64_s unz_file_info64

9.104.3.5 typedef struct unz_file_pos_s unz_file_pos

9.104.3.6 typedef struct unz_global_info_s unz_global_info

9.104.3.7 typedef struct unz_global_info64_s unz_global_info64

9.104.3.8 typedef voidp unzFile

Definition at line 75 of file unzip.h.

9.104.4 Function Documentation

9.104.4.1 int ZEXPORT unzStringFileNameCompare _Z_OF ((const char *fileName1, const char *fileName2, int iCaseSensitivity))

9.104.4.2 unzFile ZEXPORT unzOpen _Z_OF ((voidpf file))

9.104.4.3 unzFile ZEXPORT unzOpen2 _Z_OF ((voidpf file, zlib_filefunc_def *pzlib_filefunc_def))

9.104.4.4 unzFile ZEXPORT unzOpen2_64 _Z_OF ((voidpf file, zlib_filefunc64_def *pzlib_filefunc_def))

9.104.4.5 int ZEXPORT unzClose _Z_OF ((unzFile file))

Addition for GDAL : END.

9.104.4.6 int ZEXPORT unzGetGlobalInfo _Z_OF ((unzFile file, unz_global_info *pglobal_info))

9.104.4.7 int ZEXPORT unzGetGlobalInfo64 _Z_OF ((unzFile file, unz_global_info64 *pglobal_info))

9.104.4.8 int ZEXPORT unzGetGlobalComment _Z_OF ((unzFile file, char *szComment, uLong uSizeBuf))

9.104.4.9 int ZEXPORT unzLocateFile _Z_OF ((unzFile file, const char *szFileName, int iCaseSensitivity))

9.104.4.10 int ZEXPORT unzGetCurrentFileInfo64 _Z_OF ((unzFile file, unz_file_info64 *pfile_info, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize))

9.104.4.11 int ZEXPORT unzGetCurrentFileInfo _Z_OF ((unzFile file, unz_file_info *pfile_info, char *szFileName, uLong fileNameBufferSize, void *extraField, uLong extraFieldBufferSize, char *szComment, uLong commentBufferSize))

9.104.4.12 int ZEXPORT unzOpenCurrentFilePassword _Z_OF ((unzFile file, const char *password))

9.104.4.13 int ZEXPORT unzOpenCurrentFile2 _Z_OF ((unzFile file, int *method, int *level, int raw))

9.104.4.14 int ZEXPORT unzOpenCurrentFile3 _Z_OF ((unzFile file, int *method, int *level, int raw, const char *password))

9.104.4.15 int ZEXPORT unzReadCurrentFile _Z_OF ((unzFile file, voidp buf, unsigned len))

9.104.4.16 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:

9.104.4.17 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:

9.104.4.18 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().

```
{
    unz64_s* s;

    if (file==NULL || file_pos==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;
    if (!s->current_file_ok)
        return UNZ_END_OF_LIST_OF_FILE;

    file_pos->pos_in_zip_directory  = s->pos_in_central_dir;
    file_pos->num_of_file           = s->num_file;

    return UNZ_OK;
}
```

Here is the caller graph for this function:

9.104.4.19 uLong ZEXPORT unzGetOffset (unzFile *file*)

Definition at line 2102 of file unzip.c.

References unzGetOffset64().

```
{
    ZPOS64_T offset64;

    if (file==NULL)
          return 0; /*UNZ_PARAMERROR; */
    offset64 = unzGetOffset64(file);
    return (uLong)offset64;
}
```

Here is the call graph for this function:

9.104.4.20 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().

```
{
    unz64_s* s;

    if (file==NULL)
          return 0; /*UNZ_PARAMERROR; */
    s=(unz64_s*)file;
    if (!s->current_file_ok)
      return 0;
    if (s->gi.number_entry != 0 && s->gi.number_entry != 0xffff)
      if (s->num_file==s->gi.number_entry)
         return 0;
    return s->pos_in_central_dir;
}
```

Here is the caller graph for this function:

9.104.4.21 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().

```
{
    unz64_file_pos file_pos64;
    if (file_pos == NULL)
        return UNZ_PARAMERROR;

    file_pos64.pos_in_zip_directory = file_pos->
  pos_in_zip_directory;
    file_pos64.num_of_file = file_pos->num_of_file;
    return unzGoToFilePos64(file,&file_pos64);
}
```

Here is the call graph for this function:

9.104.4.22 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().

```
{
    unz64_s* s;
    int err;

    if (file==NULL || file_pos==NULL)
        return UNZ_PARAMERROR;
    s=(unz64_s*)file;

    /* jump to the right spot */
    s->pos_in_central_dir = file_pos->pos_in_zip_directory;
    s->num_file           = file_pos->num_of_file;

    /* set the current file */
    err = unz64local_GetCurrentFileInfoInternal(file,&s->cur_file_info,
                                                &s->cur_file_info_internal,
                                                NULL,0,NULL,0,NULL,0);
    /* return results */
    s->current_file_ok = (err == UNZ_OK);
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.104.4.23 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().

```
{
    unz64_s us;
    unz64_s *s;
    ZPOS64_T central_pos;
    uLong   uL;

    uLong number_disk;          /* number of the current dist, used for
                                   spaning ZIP, unsupported, always 0*/
    uLong number_disk_with_CD;  /* number the the disk with central dir, used
                                   for spaning ZIP, unsupported, always 0*/
    ZPOS64_T number_entry_CD;      /* total number of entries in
                                   the central dir
                                   (same than number_entry on nospan) */

    int err=UNZ_OK;

    if (unz_copyright[0]!=' ')
        return NULL;

    us.flags = flags;
    us.z_filefunc.zseek32_file = NULL;
    us.z_filefunc.ztell32_file = NULL;
    if (pzlib_filefunc64_32_def==NULL)
        fill_qiodevice64_filefunc(&us.z_filefunc.zfile_func64);
    else
        us.z_filefunc = *pzlib_filefunc64_32_def;
    us.is64bitOpenFunction = is64bitOpenFunction;



    us.filestream = ZOPEN64(us.z_filefunc,
                                                file,
                                                ZLIB_FILEFUNC_MODE_READ |
                                                ZLIB_FILEFUNC_MODE_EXISTING);
    if (us.filestream==NULL)
        return NULL;

    central_pos = unz64local_SearchCentralDir64(&us.z_filefunc,us.filestream);
    if (central_pos)
    {
        uLong uS;
        ZPOS64_T uL64;

```

```
        us.isZip64 = 1;

        if (ZSEEK64(us.z_filefunc, us.filestream,
                                    central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
        err=UNZ_ERRNO;

        /* the signature, already checked */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* size of zip64 end of central directory record */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&uL64)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* version made by */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uS)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* version needed to extract */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uS)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* number of this disk */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&number_disk)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* number of the disk with the start of the central directory */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&number_disk_with_CD)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* total number of entries in the central directory on this disk */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&us.gi.number_entry)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* total number of entries in the central directory */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&number_entry_CD)!=UNZ_OK)
            err=UNZ_ERRNO;

        if ((number_entry_CD!=us.gi.number_entry) ||
            (number_disk_with_CD!=0) ||
            (number_disk!=0))
            err=UNZ_BADZIPFILE;

        /* size of the central directory */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&us.size_central_dir)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* offset of start of central directory with respect to the
          starting disk number */
        if (unz64local_getLong64(&us.z_filefunc, us.filestream,&us.offset_central_dir)!=UNZ_OK)
            err=UNZ_ERRNO;

        us.gi.size_comment = 0;
    }
    else
    {
        central_pos = unz64local_SearchCentralDir(&us.z_filefunc,us.filestream);
        if (central_pos==0)
            err=UNZ_ERRNO;

        us.isZip64 = 0;

        if (ZSEEK64(us.z_filefunc, us.filestream,
                                    central_pos,ZLIB_FILEFUNC_SEEK_SET)!=0)
            err=UNZ_ERRNO;

        /* the signature, already checked */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* number of this disk */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&number_disk)!=UNZ_OK)
            err=UNZ_ERRNO;

```

```
        /* number of the disk with the start of the central directory */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&number_disk_with_CD)!=UNZ_OK)
            err=UNZ_ERRNO;

        /* total number of entries in the central dir on this disk */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        us.gi.number_entry = uL;

        /* total number of entries in the central dir */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        number_entry_CD = uL;

        if ((number_entry_CD!=us.gi.number_entry) ||
            (number_disk_with_CD!=0) ||
            (number_disk!=0))
            err=UNZ_BADZIPFILE;

        /* size of the central directory */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        us.size_central_dir = uL;

        /* offset of start of central directory with respect to the
            starting disk number */
        if (unz64local_getLong(&us.z_filefunc, us.filestream,&uL)!=UNZ_OK)
            err=UNZ_ERRNO;
        us.offset_central_dir = uL;

        /* zipfile comment length */
        if (unz64local_getShort(&us.z_filefunc, us.filestream,&us.gi.size_comment)!=UNZ_OK)
            err=UNZ_ERRNO;
    }

    if ((central_pos<us.offset_central_dir+us.size_central_dir) &&
        (err==UNZ_OK))
        err=UNZ_BADZIPFILE;

    if (err!=UNZ_OK)
    {
        if ((us.flags & UNZ_AUTO_CLOSE) != 0)
            ZCLOSE64(us.z_filefunc, us.filestream);
        else
            ZFAKECLOSE64(us.z_filefunc, us.filestream);
        return NULL;
    }

    us.byte_before_the_zipfile = central_pos -
                            (us.offset_central_dir+us.size_central_dir);
    us.central_pos = central_pos;
    us.pfile_in_zip_read = NULL;
    us.encrypted = 0;


    s=(unz64_s*)ALLOC(sizeof(unz64_s));
    if( s != NULL)
    {
        *s=us;
        unzGoToFirstFile((unzFile)s);
    }
    return (unzFile)s;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.104.4.24 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().

```
{
    unz64_s* s;
    if (file == NULL)
        return UNZ_PARAMERROR;
    s = (unz64_s*)file;
    s->flags |= flags;
    return UNZ_OK;
}
```

Here is the caller graph for this function:

9.104.4.25 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:

9.104.4.26 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:

9.105 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"

9.105.1 Data Structure Documentation

9.105.1.1 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	

9.105.1.2 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	

9.105.1.3 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	

9.105.1.4 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	

9.105.2 Macro Definition Documentation

9.105.2.1 #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().

9.105.2.2 #define BUFREADCOMMENT (0x400)

Definition at line 481 of file zip.c.

Referenced by zip64local_SearchCentralDir(), and zip64local_SearchCentralDir64().

9.105.2.3 #define CENTRALHEADERMAGIC (0x02014b50)

Definition at line 111 of file zip.c.

Referenced by zipOpenNewFileInZip4_64().

9.105.2.4 #define CRC_LOCALHEADER_OFFSET (0x0e)

Definition at line 117 of file zip.c.

9.105.2.5 #define DESCRIPTORHEADERMAGIC (0x08074b50)

Definition at line 110 of file zip.c.

Referenced by zipCloseFileInZipRaw64().

9.105.2.6 #define ENDHEADERMAGIC (0x06054b50)

Definition at line 112 of file zip.c.

Referenced by Write_EndOfCentralDirectoryRecord().

9.105.2.7 #define FLAG_LOCALHEADER_OFFSET (0x06)

Definition at line 116 of file zip.c.

9.105.2.8 #define INCLUDECRYPTINGCODE_IFCRYPTALLOWED

Definition at line 194 of file zip.c.

9.105.2.9 #define local static

Definition at line 51 of file zip.c.

9.105.2.10 #define LOCALHEADERMAGIC (0x04034b50)

Definition at line 109 of file zip.c.

Referenced by Write_LocalFileHeader().

9.105.2.11 #define MAKEULONG64(a, b) ((ZPOS64_T)(((unsigned long)(a)) | ((ZPOS64_T)((unsigned long)(b))) << 32))

Definition at line 83 of file zip.c.

9.105.2.12 #define SEEK_CUR 1

Definition at line 86 of file zip.c.

9.105.2.13 #define SEEK_END 2

Definition at line 90 of file zip.c.

Referenced by zipOpen3().

9.105.2.14 #define SEEK_SET 0

Definition at line 94 of file zip.c.

9.105.2.15 #define SIZECENTRALHEADER (0x2e) /* 46 */

Definition at line 119 of file zip.c.

Referenced by zipOpenNewFileInZip4_64().

9.105.2.16 #define SIZEDATA_INDATABLOCK (4096-(4*4))

Definition at line 107 of file zip.c.

Referenced by allocate_new_datablock(), and LoadCentralDirectoryRecord().

9.105.2.17 #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().

9.105.2.18 #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().

9.105.2.19 #define Z_BUFSIZE (64*1024) /* (16384) */

Definition at line 60 of file zip.c.

Referenced by zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), and zipWriteInFileInZip().

9.105.2.20 #define Z_MAXFILENAMEINZIP (256)

Definition at line 64 of file zip.c.

9.105.2.21 #define ZIP64ENDHEADERMAGIC (0x6064b50)

Definition at line 113 of file zip.c.

Referenced by Write_Zip64EndOfCentralDirectoryRecord().

9.105.2.22 #define ZIP64ENDLOCHEADERMAGIC (0x7064b50)

Definition at line 114 of file zip.c.

Referenced by Write_Zip64EndOfCentralDirectoryLocator().

9.105.3 Typedef Documentation

9.105.3.1 typedef struct linkedlist_data_s linkedlist_data

9.105.3.2 typedef struct linkedlist_datablock_internal_s linkedlist_datablock_internal

9.105.3.3 typedef uLongf z_crc_t

Definition at line 34 of file zip.c.

9.105.4 Function Documentation

9.105.4.1 static int zip64local_putValue _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T x, int nbByte)) [static]

9.105.4.2 static void zip64local_putValue_inmemory _Z_OF ((void *dest, ZPOS64_T x, int nbByte)) [static]

9.105.4.3 static int zip64local_getByte _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, int *pi)) [static]

9.105.4.4 static int zip64local_getShort _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, uLong *pX)) [static]

9.105.4.5 static int zip64local_getLong64 _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream, ZPOS64_T *pX)) [static]

9.105.4.6 static ZPOS64_T zip64local_SearchCentralDir _Z_OF ((const zlib_filefunc64_32_def *pzlib_filefunc_def, voidpf filestream)) [static]

9.105.4.7 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().

```
{
    linkedlist_datablock_internal* ldi;
    const unsigned char* from_copy;

    if (ll==NULL)
        return ZIP_INTERNALERROR;

    if (ll->last_block == NULL)
    {
```

```
        ll->first_block = ll->last_block = allocate_new_datablock();
        if (ll->first_block == NULL)
            return ZIP_INTERNALERROR;
    }

    ldi = ll->last_block;
    from_copy = (unsigned char*)buf;

    while (len>0)
    {
        uInt copy_this;
        uInt i;
        unsigned char* to_copy;

        if (ldi->avail_in_this_block==0)
        {
            ldi->next_datablock = allocate_new_datablock();
            if (ldi->next_datablock == NULL)
                return ZIP_INTERNALERROR;
            ldi = ldi->next_datablock ;
            ll->last_block = ldi;
        }

        if (ldi->avail_in_this_block < len)
            copy_this = (uInt)ldi->avail_in_this_block;
        else
            copy_this = (uInt)len;

        to_copy = &(ldi->data[ldi->filled_in_this_block]);

        for (i=0;i<copy_this;i++)
            *(to_copy+i)=*(from_copy+i);

        ldi->filled_in_this_block += copy_this;
        ldi->avail_in_this_block -= copy_this;
        from_copy += copy_this ;
        len -= copy_this;
    }
    return ZIP_OK;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.8 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().

```
{
    linkedlist_datablock_internal* ldi;
    ldi = (linkedlist_datablock_internal*)
                ALLOC(sizeof(linkedlist_datablock_internal));
    if (ldi!=NULL)
    {
        ldi->next_datablock = NULL ;
        ldi->filled_in_this_block = 0 ;
        ldi->avail_in_this_block = SIZEDATA_INDATABLOCK ;
    }
    return ldi;
}
```

Here is the caller graph for this function:

9.105.4.9 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().

```
{
    while (ldi!=NULL)
    {
        linkedlist_datablock_internal* ldinext = ldi->
next_datablock;
        TRYFREE(ldi);
        ldi = ldinext;
    }
}
```

Here is the caller graph for this function:

9.105.4.10 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:

9.105.4.11 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:

9.105.4.12 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 */
```

```
    if (zip64local_getLong(&pziinit->z_filefunc, pziinit->filestream,&uL)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* size of zip64 end of central directory record */
    if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->filestream, &sizeEndOfCentralDirectory)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* version made by */
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream, &VersionMadeBy)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* version needed to extract */
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream, &VersionNeeded)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* number of this disk */
    if (zip64local_getLong(&pziinit->z_filefunc, pziinit->filestream,&number_disk)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* number of the disk with the start of the central directory */
    if (zip64local_getLong(&pziinit->z_filefunc, pziinit->filestream,&number_disk_with_CD)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* total number of entries in the central directory on this disk */
    if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->filestream, &number_entry)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* total number of entries in the central directory */
    if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->filestream,&number_entry_CD)!=ZIP_OK)
      err=ZIP_ERRNO;

    if ((number_entry_CD!=number_entry) || (number_disk_with_CD!=0) || (number_disk!=0))
      err=ZIP_BADZIPFILE;

    /* size of the central directory */
    if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->filestream,&size_central_dir)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* offset of start of central directory with respect to the
    starting disk number */
    if (zip64local_getLong64(&pziinit->z_filefunc, pziinit->filestream,&offset_central_dir)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* TODO.. */
    /* read the comment from the standard central header. */
    size_comment = 0;
  }
  else
  {
    /* Read End of central Directory info */
    if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, central_pos, ZLIB_FILEFUNC_SEEK_SET)!=0)
      err=ZIP_ERRNO;

    /* the signature, already checked */
    if (zip64local_getLong(&pziinit->z_filefunc, pziinit->filestream,&uL)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* number of this disk */
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream,&number_disk)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* number of the disk with the start of the central directory */
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream,&number_disk_with_CD)!=ZIP_OK)
      err=ZIP_ERRNO;

    /* total number of entries in the central dir on this disk */
    number_entry = 0;
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream, &uL)!=ZIP_OK)
      err=ZIP_ERRNO;
    else
      number_entry = uL;

```

```
    /* total number of entries in the central dir */
    number_entry_CD = 0;
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream, &uL)!=ZIP_OK)
      err=ZIP_ERRNO;
    else
      number_entry_CD = uL;

    if ((number_entry_CD!=number_entry) || (number_disk_with_CD!=0) || (number_disk!=0))
      err=ZIP_BADZIPFILE;

    /* size of the central directory */
    size_central_dir = 0;
    if (zip64local_getLong(&pziinit->z_filefunc, pziinit->filestream, &uL)!=ZIP_OK)
      err=ZIP_ERRNO;
    else
      size_central_dir = uL;

    /* offset of start of central directory with respect to the starting disk number */
    offset_central_dir = 0;
    if (zip64local_getLong(&pziinit->z_filefunc, pziinit->filestream, &uL)!=ZIP_OK)
      err=ZIP_ERRNO;
    else
      offset_central_dir = uL;


    /* zipfile global comment length */
    if (zip64local_getShort(&pziinit->z_filefunc, pziinit->filestream, &size_comment)!=ZIP_OK)
      err=ZIP_ERRNO;
  }

  if ((central_pos<offset_central_dir+size_central_dir) &&
    (err==ZIP_OK))
    err=ZIP_BADZIPFILE;

  if (err!=ZIP_OK)
  {
      if ((pziinit->flags & ZIP_AUTO_CLOSE) != 0) {
        ZCLOSE64(pziinit->z_filefunc, pziinit->filestream);
      } else {
        ZFAKECLOSE64(pziinit->z_filefunc, pziinit->filestream);
      }
    return ZIP_ERRNO;
  }

  if (size_comment>0)
  {
    pziinit->globalcomment = (char*)ALLOC(size_comment+1);
    if (pziinit->globalcomment)
    {
      size_comment = ZREAD64(pziinit->z_filefunc, pziinit->filestream, pziinit->globalcomment,size_comment);
      pziinit->globalcomment[size_comment]=0;
    }
  }

  byte_before_the_zipfile = central_pos - (offset_central_dir+size_central_dir);
  pziinit->add_position_when_writting_offset = byte_before_the_zipfile;

  {
    ZPOS64_T size_central_dir_to_read = size_central_dir;
    size_t buf_size = SIZEDATA_INDATABLOCK;
    void* buf_read = (void*)ALLOC(buf_size);
    if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, offset_central_dir + byte_before_the_zipfile, ZLIB_FILEFUNC_SEEK_SET) != 0)
      err=ZIP_ERRNO;

    while ((size_central_dir_to_read>0) && (err==ZIP_OK))
    {
      ZPOS64_T read_this = SIZEDATA_INDATABLOCK;
      if (read_this > size_central_dir_to_read)
        read_this = size_central_dir_to_read;

      if (ZREAD64(pziinit->z_filefunc, pziinit->filestream,buf_read,(uLong)read_this) != read_this)
        err=ZIP_ERRNO;

      if (err==ZIP_OK)
        err = add_data_in_datablock(&pziinit->central_dir,buf_read, (uLong)read_this);

      size_central_dir_to_read-=read_this;
    }
```

```
    TRYFREE(buf_read);
  }
  pziinit->begin_pos = byte_before_the_zipfile;
  pziinit->number_entry = number_entry_CD;

  if (ZSEEK64(pziinit->z_filefunc, pziinit->filestream, offset_central_dir+
  byte_before_the_zipfile,ZLIB_FILEFUNC_SEEK_SET) != 0)
    err=ZIP_ERRNO;

  return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.13 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().

```
{
  int err = ZIP_OK;

  /*signature*/
  err = zip64local_putValue(&zi->z_filefunc,zi->
filestream,(uLong)ENDHEADERMAGIC,4);

  if (err==ZIP_OK) /* number of this disk */
    err = zip64local_putValue(&zi->z_filefunc,zi->
filestream,(uLong)0,2);

  if (err==ZIP_OK) /* number of the disk with the start of the central directory */
    err = zip64local_putValue(&zi->z_filefunc,zi->
filestream,(uLong)0,2);

  if (err==ZIP_OK) /* total number of entries in the central dir on this disk */
```

```
  {
    {
      if(zi->number_entry >= 0xFFFF)
        err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0xffff,2); /* use value in ZIP64 record */
      else
        err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)zi->number_entry,2);
    }
  }

  if (err==ZIP_OK) /* total number of entries in the central dir */
  {
    if(zi->number_entry >= 0xFFFF)
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0xffff,2); /* use value in ZIP64 record */
    else
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)zi->number_entry,2);
  }

  if (err==ZIP_OK) /* size of the central directory */
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)size_centraldir,4);

  if (err==ZIP_OK) /* offset of start of central directory with respect to the starting disk number */
  {
    ZPOS64_T pos = centraldir_pos_inzip - zi->add_position_when_writting_offset;
    if(pos >= 0xffffffff)
    {
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream, (uLong)0xffffffff,4);
    }
    else
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream, (uLong)(centraldir_pos_inzip - zi->add_position_when_writting_offset),4);
  }

   return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.14 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().

```
{
  int err = ZIP_OK;
  uInt size_global_comment = 0;

  if(global_comment != NULL)
    size_global_comment = (uInt)strlen(global_comment);

  err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)size_global_comment,2);

  if (err == ZIP_OK && size_global_comment > 0)
  {
    if (ZWRITE64(zi->z_filefunc,zi->filestream, global_comment, size_global_comment) != size_global_comment)
      err = ZIP_ERRNO;
  }
  return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.15 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().

```
{
  /* write the local header */
  int err;
  uInt size_filename = (uInt)strlen(filename);
  uInt size_extrafield = size_extrafield_local;

  err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)LOCALHEADERMAGIC, 4);

  if (err==ZIP_OK)
  {
    if(zi->ci.zip64)
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)45,2);/* version needed to extract */
    else
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)version_to_extract,2);
  }

  if (err==ZIP_OK)
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)zi->ci.flag,2);

  if (err==ZIP_OK)
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)zi->ci.method,2);

  if (err==ZIP_OK)
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)zi->ci.dosDate,4);

  /* CRC / Compressed size / Uncompressed size will be filled in later and rewritten later */
  if (err==ZIP_OK)
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0,4); /* crc 32, unknown */
  if (err==ZIP_OK)
  {
    if(zi->ci.zip64)
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0xFFFFFFFF,4); /* compressed size, unknown */
    else
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0,4); /* compressed size, unknown */
  }
  if (err==ZIP_OK)
  {
    if(zi->ci.zip64)
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0xFFFFFFFF,4); /* uncompressed size, unknown */
    else
      err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0,4); /* uncompressed size, unknown */
  }

  if (err==ZIP_OK)
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)size_filename,2);

  if(zi->ci.zip64)
  {
    size_extrafield += 20;
  }

  if (err==ZIP_OK)
    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)size_extrafield,2);

  if ((err==ZIP_OK) && (size_filename > 0))
  {
    if (ZWRITE64(zi->z_filefunc,zi->filestream,filename,size_filename)!=size_filename)
      err = ZIP_ERRNO;
  }

  if ((err==ZIP_OK) && (size_extrafield_local > 0))
  {
    if (ZWRITE64(zi->z_filefunc, zi->filestream, extrafield_local, size_extrafield_local) != size_extrafield_local)
      err = ZIP_ERRNO;
  }


  if ((err==ZIP_OK) && (zi->ci.zip64))
  {
      /* write the Zip64 extended info */
      short HeaderID = 1;
      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:

9.105.4.16 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;
```

```
 ZPOS64_T pos = zip64eocd_pos_inzip - zi->add_position_when_writting_offset;

 err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)ZIP64ENDLOCHEADERMAGIC,4);

 /*num disks*/
   if (err==ZIP_OK) /* number of the disk with the start of the central directory */
     err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0,4);

 /*relative offset*/
   if (err==ZIP_OK) /* Relative offset to the Zip64EndOfCentralDirectory */
     err = zip64local_putValue(&zi->z_filefunc,zi->filestream, pos,8);

 /*total disks*/ /* Do not support spawning of disk so always say 1 here*/
   if (err==ZIP_OK) /* number of the disk with the start of the central directory */
     err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)1,4);

   return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.17 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().

```
{
  int err = ZIP_OK;

  uLong Zip64DataSize = 44;

  err = zip64local_putValue(&zi->z_filefunc,zi->
```

```
    filestream,(uLong)ZIP64ENDHEADERMAGIC,4);

 if (err==ZIP_OK) /* size of this 'zip64 end of central directory' */
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(ZPOS64_T)Zip64DataSize,8); /* why ZPOS64_T of this ? */

 if (err==ZIP_OK) /* version made by */
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)45,2);

 if (err==ZIP_OK) /* version needed */
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)45,2);

 if (err==ZIP_OK) /* number of this disk */
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0,4);

 if (err==ZIP_OK) /* number of the disk with the start of the central directory */
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)0,4);

 if (err==ZIP_OK) /* total number of entries in the central dir on this disk */
   err = zip64local_putValue(&zi->z_filefunc, zi->filestream, zi->number_entry, 8);

 if (err==ZIP_OK) /* total number of entries in the central dir */
   err = zip64local_putValue(&zi->z_filefunc, zi->filestream, zi->number_entry, 8);

 if (err==ZIP_OK) /* size of the central directory */
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(ZPOS64_T)size_centraldir,8);

 if (err==ZIP_OK) /* offset of start of central directory with respect to the starting disk number */
 {
   ZPOS64_T pos = centraldir_pos_inzip - zi->add_position_when_writting_offset;
   err = zip64local_putValue(&zi->z_filefunc,zi->filestream, (ZPOS64_T)pos,8);
 }
 return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.18 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().

```
{
    int err=ZIP_OK;

    if (zi->ci.encrypt != 0)
    {
#ifndef NOCRYPT
        uInt i;
        int t;
        for (i=0;i<zi->ci.pos_in_buffered_data;i++)
            zi->ci.buffered_data[i] = zencode(zi->ci.keys, zi->ci.pcrc_32_tab, zi->ci.buffered_data[i],t);
#endif
    }

    if (ZWRITE64(zi->z_filefunc,zi->filestream,zi->ci.buffered_data,zi->ci.pos_in_buffered_data) != zi->ci.pos_in_buffered_data)
      err = ZIP_ERRNO;

    zi->ci.totalCompressedData += zi->ci.pos_in_buffered_data;

#ifdef HAVE_BZIP2
    if(zi->ci.method == Z_BZIP2ED)
    {
      zi->ci.totalUncompressedData += zi->ci.bstream.total_in_lo32;
      zi->ci.bstream.total_in_lo32 = 0;
      zi->ci.bstream.total_in_hi32 = 0;
    }
    else
#endif
    {
      zi->ci.totalUncompressedData += zi->ci.stream.total_in;
      zi->ci.stream.total_in = 0;
    }


    zi->ci.pos_in_buffered_data = 0;

    return err;
}
```

Here is the caller graph for this function:

9.105.4.19 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().

```
{
    unsigned char c;
    int err = (int)ZREAD64(*pzlib_filefunc_def,filestream,&c,1);
    if (err==1)
    {
        *pi = (int)c;
        return ZIP_OK;
    }
    else
    {
        if (ZERROR64(*pzlib_filefunc_def,filestream))
            return ZIP_ERRNO;
        else
            return ZIP_EOF;
    }
}
```

Here is the caller graph for this function:

9.105.4.20 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 zip64local_getByte(), and ZIP_OK.

Referenced by LoadCentralDirectoryRecord(), and zip64local_SearchCentralDir64().

```
{
    uLong x ;
    int i = 0;
    int err;

    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
    x = (uLong)i;

    if (err==ZIP_OK)
        err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
    x += ((uLong)i)<<8;

    if (err==ZIP_OK)
        err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
    x += ((uLong)i)<<16;

    if (err==ZIP_OK)
        err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
    x += ((uLong)i)<<24;

    if (err==ZIP_OK)
        *pX = x;
    else
        *pX = 0;
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.21 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 zip64local_getByte(), and ZIP_OK.

Referenced by LoadCentralDirectoryRecord(), and zip64local_SearchCentralDir64().

```
{
  ZPOS64_T x;
  int i = 0;
  int err;

  err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x = (ZPOS64_T)i;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<8;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<16;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<24;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<32;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<40;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<48;

  if (err==ZIP_OK)
    err = zip64local_getByte(pzlib_filefunc_def,filestream,&i);
  x += ((ZPOS64_T)i)<<56;

  if (err==ZIP_OK)
    *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:

9.105.4.22 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 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:

9.105.4.23 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().

```
{
    unsigned char buf[8];
    int n;
    for (n = 0; n < nbByte; n++)
    {
        buf[n] = (unsigned char)(x & 0xff);
        x >>= 8;
    }
    if (x != 0)
      {     /* data overflow - hack for ZIP64 (X Roche) */
      for (n = 0; n < nbByte; n++)
        {
          buf[n] = 0xff;
        }
      }

    if (ZWRITE64(*pzlib_filefunc_def,filestream,buf,nbByte)!=(uLong)nbByte)
        return ZIP_ERRNO;
    else
        return ZIP_OK;
}
```

Here is the caller graph for this function:

9.105.4.24 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().

```
{
    unsigned char* buf=(unsigned char*)dest;
    int n;
    for (n = 0; n < nbByte; n++) {
        buf[n] = (unsigned char)(x & 0xff);
        x >>= 8;
    }

    if (x != 0)
    {     /* data overflow - hack for ZIP64 */
       for (n = 0; n < nbByte; n++)
       {
          buf[n] = 0xff;
       }
    }
}
```

Here is the caller graph for this function:

9.105.4.25 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().

```
{
  unsigned char* buf;
  ZPOS64_T uSizeFile;
```

```
  ZPOS64_T uBackRead;
  ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
  ZPOS64_T uPosFound=0;

  if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
    return 0;


  uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);

  if (uMaxBack>uSizeFile)
    uMaxBack = uSizeFile;

  buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
  if (buf==NULL)
    return 0;

  uBackRead = 4;
  while (uBackRead<uMaxBack)
  {
    uLong uReadSize;
    ZPOS64_T uReadPos ;
    int i;
    if (uBackRead+BUFREADCOMMENT>uMaxBack)
      uBackRead = uMaxBack;
    else
      uBackRead+=BUFREADCOMMENT;
    uReadPos = uSizeFile-uBackRead ;

    uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
      (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
    if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
  ZLIB_FILEFUNC_SEEK_SET)!=0)
      break;

    if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
      break;

    for (i=(int)uReadSize-3; (i--)>0;)
      if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) &&
        ((*(buf+i+2))==0x05) && ((*(buf+i+3))==0x06))
      {
        uPosFound = uReadPos+i;
        break;
      }

    if (uPosFound!=0)
      break;
  }
  TRYFREE(buf);
  return uPosFound;
}
```

Here is the caller graph for this function:

9.105.4.26 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 {
```

```
unsigned char* buf;
ZPOS64_T uSizeFile;
ZPOS64_T uBackRead;
ZPOS64_T uMaxBack=0xffff; /* maximum size of global comment */
ZPOS64_T uPosFound=0;
uLong uL;
ZPOS64_T relativeOffset;

if (ZSEEK64(*pzlib_filefunc_def,filestream,0,ZLIB_FILEFUNC_SEEK_END) != 0)
  return 0;

uSizeFile = ZTELL64(*pzlib_filefunc_def,filestream);

if (uMaxBack>uSizeFile)
  uMaxBack = uSizeFile;

buf = (unsigned char*)ALLOC(BUFREADCOMMENT+4);
if (buf==NULL)
  return 0;

uBackRead = 4;
while (uBackRead<uMaxBack)
{
  uLong uReadSize;
  ZPOS64_T uReadPos;
  int i;
  if (uBackRead+BUFREADCOMMENT>uMaxBack)
    uBackRead = uMaxBack;
  else
    uBackRead+=BUFREADCOMMENT;
  uReadPos = uSizeFile-uBackRead ;

  uReadSize = ((BUFREADCOMMENT+4) < (uSizeFile-uReadPos)) ?
    (BUFREADCOMMENT+4) : (uLong)(uSizeFile-uReadPos);
  if (ZSEEK64(*pzlib_filefunc_def,filestream,uReadPos,
ZLIB_FILEFUNC_SEEK_SET)!=0)
    break;

  if (ZREAD64(*pzlib_filefunc_def,filestream,buf,uReadSize)!=uReadSize)
    break;

  for (i=(int)uReadSize-3; (i--)>0;)
  {
    /* Signature "0x07064b50" Zip64 end of central directory locater */
    if (((*(buf+i))==0x50) && ((*(buf+i+1))==0x4b) && ((*(buf+i+2))==0x06) && ((*(buf+i+3))==0x07))
    {
      uPosFound = uReadPos+i;
      break;
    }
  }

    if (uPosFound!=0)
      break;
}

TRYFREE(buf);
if (uPosFound == 0)
  return 0;

/* Zip64 end of central directory locator */
if (ZSEEK64(*pzlib_filefunc_def,filestream, uPosFound,
ZLIB_FILEFUNC_SEEK_SET)!=0)
  return 0;

/* the signature, already checked */
if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
ZIP_OK)
  return 0;

/* number of the disk with the start of the zip64 end of  central directory */
if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
ZIP_OK)
  return 0;
if (uL != 0)
  return 0;

/* relative offset of the zip64 end of central directory record */
if (zip64local_getLong64(pzlib_filefunc_def,filestream,&relativeOffset)!=
ZIP_OK)
  return 0;

/* total number of disks */
if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
ZIP_OK)
  return 0;
if (uL != 1)
  return 0;
```

```
  /* Goto Zip64 end of central directory record */
  if (ZSEEK64(*pzlib_filefunc_def,filestream, relativeOffset,
  ZLIB_FILEFUNC_SEEK_SET)!=0)
    return 0;

  /* the signature */
  if (zip64local_getLong(pzlib_filefunc_def,filestream,&uL)!=
  ZIP_OK)
    return 0;

  if (uL != 0x06064b50) /* signature of 'Zip64 end of central directory' */
    return 0;

  return relativeOffset;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.27 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().

```
{
    uLong year = (uLong)ptm->tm_year;
    if (year>=1980)
        year-=1980;
    else if (year>=80)
        year-=80;
    return
      (uLong) (((ptm->tm_mday) + (32 * (ptm->tm_mon+1)) + (512 * year)) << 16) |
        ((ptm->tm_sec/2) + (32* ptm->tm_min) + (2048 * (uLong)ptm->
  tm_hour));
}
```

Here is the caller graph for this function:

9.105.4.28 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().

```
{
    zip64_internal* zi;
    if (file == NULL)
        return ZIP_PARAMERROR;
    zi = (zip64_internal*)file;
    zi->flags &= ~flags;
    // If the data descriptor is not written, we can't use a non-seekable output.
    if ((zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) == 0) {
        zi->flags &= ~ZIP_SEQUENTIAL;
    }
    return ZIP_OK;
}
```

Here is the caller graph for this function:

9.105.4.29 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().

```
{
    zip64_internal* zi;
    int err = 0;
    uLong size_centraldir = 0;
    ZPOS64_T centraldir_pos_inzip;
    ZPOS64_T pos;

    if (file == NULL)
        return ZIP_PARAMERROR;

    zi = (zip64_internal*)file;

    if (zi->in_opened_file_inzip == 1)
    {
        err = zipCloseFileInZip (file);
    }

#ifndef NO_ADDFILEINEXISTINGZIP
    if (global_comment==NULL)
        global_comment = zi->globalcomment;
#endif

    centraldir_pos_inzip = ZTELL64(zi->z_filefunc,zi->filestream);

    if (err==ZIP_OK)
    {
        linkedlist_datablock_internal* ldi = zi->central_dir.first_block;
        while (ldi!=NULL)
        {
            if ((err==ZIP_OK) && (ldi->filled_in_this_block>0))
            {
                if (ZWRITE64(zi->z_filefunc,zi->filestream, ldi->data, ldi->filled_in_this_block) != ldi->filled_in_this_block)
                    err = ZIP_ERRNO;
            }

            size_centraldir += ldi->filled_in_this_block;
            ldi = ldi->next_datablock;
        }
    }
    free_linkedlist(&(zi->central_dir));

    pos = centraldir_pos_inzip - zi->add_position_when_writting_offset;
    if(pos >= 0xffffffff || zi->number_entry > 0xFFFF)
    {
      ZPOS64_T Zip64EOCDpos = ZTELL64(zi->z_filefunc,zi->filestream);
      Write_Zip64EndOfCentralDirectoryRecord(zi, size_centraldir, centraldir_pos_inzip);

      Write_Zip64EndOfCentralDirectoryLocator(zi, Zip64EOCDpos);
    }

    if (err==ZIP_OK)
      err = Write_EndOfCentralDirectoryRecord(zi, size_centraldir, centraldir_pos_inzip);

    if(err == ZIP_OK)
      err = Write_GlobalComment(zi, global_comment);

    if ((zi->flags & ZIP_AUTO_CLOSE) != 0) {
        if (ZCLOSE64(zi->z_filefunc,zi->filestream) != 0) {
            if (err == ZIP_OK)
                err = ZIP_ERRNO;
        }
    } else {
        if (ZFAKECLOSE64(zi->z_filefunc,zi->filestream) != 0) {
            if (err == ZIP_OK)
                err = ZIP_ERRNO;
        }
    }

#ifndef NO_ADDFILEINEXISTINGZIP
    TRYFREE(zi->globalcomment);
#endif
    TRYFREE(zi);

```

```
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.30 int ZEXPORT zipCloseFileInZip (zipFile *file*)

Definition at line 1810 of file zip.c.

References zipCloseFileInZipRaw().

Referenced by QuaZipFile::close(), zipClose(), and zipOpenNewFileInZip4_64().

```
{
    return zipCloseFileInZipRaw (file,0,0);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.31 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:

9.105.4.32 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().

```
{
    zip64_internal* zi;
    ZPOS64_T compressed_size;
    uLong invalidValue = 0xffffffff;
    short datasize = 0;
    int err=ZIP_OK;

    if (file == NULL)
        return ZIP_PARAMERROR;
    zi = (zip64_internal*)file;

    if (zi->in_opened_file_inzip == 0)
        return ZIP_PARAMERROR;
    zi->ci.stream.avail_in = 0;

    if ((zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
                {
                        while (err==ZIP_OK)
                        {
                                uLong uTotalOutBefore;
                                if (zi->ci.stream.avail_out == 0)
                                {
                                        if (zip64FlushWriteBuffer(zi) == ZIP_ERRNO)
                                                err = ZIP_ERRNO;
                                        zi->ci.stream.avail_out = (uInt)Z_BUFSIZE;
                                        zi->ci.stream.next_out = zi->ci.buffered_data;
                                }
                                uTotalOutBefore = zi->ci.stream.total_out;
                                err=deflate(&zi->ci.stream,  Z_FINISH);
                                zi->ci.pos_in_buffered_data += (uInt)(zi->ci.stream.total_out - uTotalOutBefore) ;
                        }
                }
    else if ((zi->ci.method == Z_BZIP2ED) && (!zi->ci.raw))
    {
#ifdef HAVE_BZIP2
      err = BZ_FINISH_OK;
      while (err==BZ_FINISH_OK)
      {
        uLong uTotalOutBefore;
        if (zi->ci.bstream.avail_out == 0)
        {
          if (zip64FlushWriteBuffer(zi) == ZIP_ERRNO)
            err = ZIP_ERRNO;
          zi->ci.bstream.avail_out = (uInt)Z_BUFSIZE;
          zi->ci.bstream.next_out = (char*)zi->ci.buffered_data;
        }
        uTotalOutBefore = zi->ci.bstream.total_out_lo32;
        err=BZ2_bzCompress(&zi->ci.bstream,  BZ_FINISH);
        if(err == BZ_STREAM_END)
          err = Z_STREAM_END;

        zi->ci.pos_in_buffered_data += (uInt)(zi->ci.bstream.total_out_lo32 - uTotalOutBefore);
      }

      if(err == BZ_FINISH_OK)
        err = ZIP_OK;
#endif
    }

```

```
    if (err==Z_STREAM_END)
        err=ZIP_OK; /* this is normal */

    if ((zi->ci.pos_in_buffered_data>0) && (err==ZIP_OK))
                {
        if (zip64FlushWriteBuffer(zi)==ZIP_ERRNO)
            err = ZIP_ERRNO;
                }

    if ((zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
    {
        int tmp_err = deflateEnd(&zi->ci.stream);
        if (err == ZIP_OK)
            err = tmp_err;
        zi->ci.stream_initialised = 0;
    }
#ifdef HAVE_BZIP2
    else if((zi->ci.method == Z_BZIP2ED) && (!zi->ci.raw))
    {
      int tmperr = BZ2_bzCompressEnd(&zi->ci.bstream);
                        if (err==ZIP_OK)
                                err = tmperr;
                        zi->ci.stream_initialised = 0;
    }
#endif

    if (!zi->ci.raw)
    {
        crc32 = (uLong)zi->ci.crc32;
        uncompressed_size = zi->ci.totalUncompressedData;
    }
    compressed_size = zi->ci.totalCompressedData;

#    ifndef NOCRYPT
    compressed_size += zi->ci.crypt_header_size;
#    endif

    /* update Current Item crc and sizes, */
    if(compressed_size >= 0xffffffff || uncompressed_size >= 0xffffffff || zi->ci.pos_local_header >= 0xffffffff)
    {
      /*version Made by*/
      zip64local_putValue_inmemory(zi->ci.central_header+4,(uLong)45,2);
      /*version needed*/
      zip64local_putValue_inmemory(zi->ci.central_header+6,(uLong)45,2);

    }

    zip64local_putValue_inmemory(zi->ci.central_header+16,crc32,4); /*crc*/


    if(compressed_size >= 0xffffffff)
      zip64local_putValue_inmemory(zi->ci.central_header+20, invalidValue,4); /*compr size*/
    else
      zip64local_putValue_inmemory(zi->ci.central_header+20, compressed_size,4); /*compr size*/

    /* set internal file attributes field */
    if (zi->ci.stream.data_type == Z_ASCII)
        zip64local_putValue_inmemory(zi->ci.central_header+36,(uLong)Z_ASCII,2);

    if(uncompressed_size >= 0xffffffff)
      zip64local_putValue_inmemory(zi->ci.central_header+24, invalidValue,4); /*uncompr size*/
    else
      zip64local_putValue_inmemory(zi->ci.central_header+24, uncompressed_size,4); /*uncompr size*/

    /* Add ZIP64 extra info field for uncompressed size */
    if(uncompressed_size >= 0xffffffff)
      datasize += 8;

    /* Add ZIP64 extra info field for compressed size */
    if(compressed_size >= 0xffffffff)
      datasize += 8;

    /* Add ZIP64 extra info field for relative offset to local file header of current file */
    if(zi->ci.pos_local_header >= 0xffffffff)
      datasize += 8;

    if(datasize > 0)
    {
```

```
      char* p = NULL;

      if((uLong)(datasize + 4) > zi->ci.size_centralExtraFree)
      {
        /* we can not write more data to the buffer that we have room for. */
        return ZIP_BADZIPFILE;
      }

      p = zi->ci.central_header + zi->ci.size_centralheader;

      /* Add Extra Information Header for 'ZIP64 information' */
      zip64local_putValue_inmemory(p, 0x0001, 2); /* HeaderID */
      p += 2;
      zip64local_putValue_inmemory(p, datasize, 2); /* DataSize */
      p += 2;

      if(uncompressed_size >= 0xffffffff)
      {
        zip64local_putValue_inmemory(p, uncompressed_size, 8);
        p += 8;
      }

      if(compressed_size >= 0xffffffff)
      {
        zip64local_putValue_inmemory(p, compressed_size, 8);
        p += 8;
      }

      if(zi->ci.pos_local_header >= 0xffffffff)
      {
        zip64local_putValue_inmemory(p, zi->ci.pos_local_header, 8);
        p += 8;
      }

      /* Update how much extra free space we got in the memory buffer */
      /* and increase the centralheader size so the new ZIP64 fields are included */
      /* ( 4 below is the size of HeaderID and DataSize field ) */
      zi->ci.size_centralExtraFree -= datasize + 4;
      zi->ci.size_centralheader += datasize + 4;

      /* Update the extra info size field */
      zi->ci.size_centralExtra += datasize + 4;
      zip64local_putValue_inmemory(zi->ci.central_header+30,(uLong)zi->ci.size_centralExtra,2);
    }

    if (err==ZIP_OK)
        err = add_data_in_datablock(&zi->central_dir, zi->ci.central_header, (uLong)zi->ci.size_centralheader);

    free(zi->ci.central_header);

    if (err==ZIP_OK)
    {
        if ((zi->flags & ZIP_SEQUENTIAL) == 0) {
            /* Update the LocalFileHeader with the new values. */

            ZPOS64_T cur_pos_inzip = ZTELL64(zi->z_filefunc,zi->filestream);

            if (ZSEEK64(zi->z_filefunc,zi->filestream, zi->ci.pos_local_header + 14,ZLIB_FILEFUNC_SEEK_SET)!=0)
                err = ZIP_ERRNO;

            if (err==ZIP_OK)
                err = zip64local_putValue(&zi->z_filefunc,zi->filestream,crc32,4); /* crc 32, unknown */

            if(uncompressed_size >= 0xffffffff || compressed_size >= 0xffffffff)
            {
                if(zi->ci.pos_zip64extrainfo > 0)
                {
                    /* Update the size in the ZIP64 extended field. */
                    if (ZSEEK64(zi->z_filefunc,zi->filestream, zi->ci.pos_zip64extrainfo + 4,ZLIB_FILEFUNC_SEEK_SET)!=0)
                        err = ZIP_ERRNO;

                    if (err==ZIP_OK) /* compressed size, unknown */
                        err = zip64local_putValue(&zi->z_filefunc, zi->filestream, uncompressed_size, 8);

                    if (err==ZIP_OK) /* uncompressed size, unknown */
                        err = zip64local_putValue(&zi->z_filefunc, zi->filestream, compressed_size, 8);
                }
            }
```

```
            else
            {
                if (err==ZIP_OK) /* compressed size, unknown */
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,compressed_size,4);

                if (err==ZIP_OK) /* uncompressed size, unknown */
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,uncompressed_size,4);
            }

            if (ZSEEK64(zi->z_filefunc,zi->filestream, cur_pos_inzip, ZLIB_FILEFUNC_SEEK_SET)!=0)
                err = ZIP_ERRNO;
        }

        if ((zi->ci.flag & 8) != 0) {
            /* Write local Descriptor after file data */
            if (err==ZIP_OK)
                err = zip64local_putValue(&zi->z_filefunc,zi->filestream,(uLong)DESCRIPTORHEADERMAGIC,4);
            if (err==ZIP_OK)
                err = zip64local_putValue(&zi->z_filefunc,zi->filestream,crc32,4); /* crc 32, unknown */
            if (zi->ci.zip64) {
                if (err==ZIP_OK) /* compressed size, unknown */
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,compressed_size,8);

                if (err==ZIP_OK) /* uncompressed size, unknown */
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,uncompressed_size,8);
            } else {
                if (err==ZIP_OK) /* compressed size, unknown */
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,compressed_size,4);
                if (err==ZIP_OK) /* uncompressed size, unknown */
                    err = zip64local_putValue(&zi->z_filefunc,zi->filestream,uncompressed_size,4);
            }
        }
    }

    zi->number_entry ++;
    zi->in_opened_file_inzip = 0;

    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.33 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:

9.105.4.34 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().

```
{
    if (pzlib_filefunc32_def != NULL)
    {
        zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
        fill_zlib_filefunc64_32_def_from_filefunc32(&zlib_filefunc64_32_def_fill,pzlib_filefunc32_def);
        return zipOpen3(file, append, globalcomment, &zlib_filefunc64_32_def_fill, ZIP_DEFAULT_FLAGS);
    }
    else
        return zipOpen3(file, append, globalcomment, NULL, ZIP_DEFAULT_FLAGS);
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.35 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.

```
{
    if (pzlib_filefunc_def != NULL)
    {
```

```
        zlib_filefunc64_32_def zlib_filefunc64_32_def_fill;
        zlib_filefunc64_32_def_fill.zfile_func64 = *pzlib_filefunc_def;
        zlib_filefunc64_32_def_fill.ztell32_file = NULL;
        zlib_filefunc64_32_def_fill.zseek32_file = NULL;
        return zipOpen3(file, append, globalcomment, &zlib_filefunc64_32_def_fill,
    ZIP_DEFAULT_FLAGS);
    }
    else
        return zipOpen3(file, append, globalcomment, NULL,
    ZIP_DEFAULT_FLAGS);
}
```

Here is the call graph for this function:

9.105.4.36 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().

```
{
    zip64_internal ziinit;
    zip64_internal* zi;
    int err=ZIP_OK;

    ziinit.flags = flags;
    ziinit.z_filefunc.zseek32_file = NULL;
    ziinit.z_filefunc.ztell32_file = NULL;
    if (pzlib_filefunc64_32_def==NULL)
        fill_qiodevice64_filefunc(&ziinit.z_filefunc.
    zfile_func64);
```

```
    else
        ziinit.z_filefunc = *pzlib_filefunc64_32_def;

    ziinit.filestream = ZOPEN64(ziinit.z_filefunc,
                    file,
                    (append == APPEND_STATUS_CREATE) ?
                    (ZLIB_FILEFUNC_MODE_READ | ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_CREATE) :
                      (ZLIB_FILEFUNC_MODE_READ | ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_EXISTING));

    if (ziinit.filestream == NULL)
        return NULL;

    if (append == APPEND_STATUS_CREATEAFTER)
        ZSEEK64(ziinit.z_filefunc,ziinit.filestream,0,SEEK_END);

    ziinit.begin_pos = ZTELL64(ziinit.z_filefunc,ziinit.filestream);
    ziinit.in_opened_file_inzip = 0;
    ziinit.ci.stream_initialised = 0;
    ziinit.number_entry = 0;
    ziinit.add_position_when_writting_offset = 0;
    init_linkedlist(&(ziinit.central_dir));



    zi = (zip64_internal*)ALLOC(sizeof(zip64_internal));
    if (zi==NULL)
    {
        if ((ziinit.flags & ZIP_AUTO_CLOSE) != 0) {
            ZCLOSE64(ziinit.z_filefunc,ziinit.filestream);
        } else {
            ZFAKECLOSE64(ziinit.z_filefunc,ziinit.filestream);
        }
        return NULL;
    }

    /* now we add file in a zipfile */
#    ifndef NO_ADDFILEINEXISTINGZIP
    ziinit.globalcomment = NULL;
    if (append == APPEND_STATUS_ADDINZIP)
    {
      /* Read and Cache Central Directory Records */
      err = LoadCentralDirectoryRecord(&ziinit);
    }

    if (globalcomment)
    {
      *globalcomment = ziinit.globalcomment;
    }
#    endif /* !NO_ADDFILEINEXISTINGZIP*/

    if (err != ZIP_OK)
    {
#    ifndef NO_ADDFILEINEXISTINGZIP
        TRYFREE(ziinit.globalcomment);
#    endif /* !NO_ADDFILEINEXISTINGZIP*/
        TRYFREE(zi);
        return NULL;
    }
    else
    {
        *zi = ziinit;
        return (zipFile)zi;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.37 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:

9.105.4.38 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:

9.105.4.39 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().

```
{
    return zipOpenNewFileInZip4_64 (file, filename, zipfi,
                                    extrafield_local, size_extrafield_local,
                                    extrafield_global, size_extrafield_global,
                                    comment, method, level, raw,
                                    -MAX_WBITS, DEF_MEM_LEVEL, Z_DEFAULT_STRATEGY,
                                    NULL, 0, VERSIONMADEBY, 0, 0);
}
```

Here is the call graph for this function:

9.105.4.40 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().

```
{
    return zipOpenNewFileInZip4_64 (file, filename, zipfi,
                                    extrafield_local, size_extrafield_local,
                                    extrafield_global, size_extrafield_global,
                                    comment, method, level, raw,
                                    -MAX_WBITS, DEF_MEM_LEVEL, Z_DEFAULT_STRATEGY,
                                    NULL, 0, VERSIONMADEBY, 0, zip64);
}
```

Here is the call graph for this function:

9.105.4.41 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().

```
{
    return zipOpenNewFileInZip4_64 (file, filename, zipfi,
                                    extrafield_local, size_extrafield_local,
                                    extrafield_global, size_extrafield_global,
                                    comment, method, level, raw,
                                    windowBits, memLevel, strategy,
                                    password, crcForCrypting, VERSIONMADEBY, 0, 0);
}
```

Here is the call graph for this function:

9.105.4.42 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:

9.105.4.43 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().

```
{
    return zipOpenNewFileInZip4_64 (file, filename, zipfi,
                                    extrafield_local, size_extrafield_local,
                                    extrafield_global, size_extrafield_global,
                                    comment, method, level, raw,
                                    windowBits, memLevel, strategy,
                                    password, crcForCrypting, versionMadeBy, flagBase, 0);
}
```

Here is the call graph for this function:

9.105.4.44 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().

```
{
    zip64_internal* zi;
    uInt size_filename;
    uInt size_comment;
    uInt i;
    int err = ZIP_OK;
    uLong version_to_extract;

#    ifdef NOCRYPT
    if (password != NULL)
        return ZIP_PARAMERROR;
#    endif

    if (file == NULL)
        return ZIP_PARAMERROR;

#ifdef HAVE_BZIP2
    if ((method!=0) && (method!=Z_DEFLATED) && (method!=Z_BZIP2ED))
      return ZIP_PARAMERROR;
#else
    if ((method!=0) && (method!=Z_DEFLATED))
      return ZIP_PARAMERROR;
#endif
```

```
    zi = (zip64_internal*)file;

    if (zi->in_opened_file_inzip == 1)
    {
        err = zipCloseFileInZip (file);
        if (err != ZIP_OK)
            return err;
    }

    if (method == 0
            && (level == 0 || (zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) == 0)
            && (zi->flags & ZIP_SEQUENTIAL) == 0)
    {
        version_to_extract = 10;
    }
    else
    {
        version_to_extract = 20;
    }

    if (filename==NULL)
        filename="-";

    if (comment==NULL)
        size_comment = 0;
    else
        size_comment = (uInt)strlen(comment);

    size_filename = (uInt)strlen(filename);

    if (zipfi == NULL)
        zi->ci.dosDate = 0;
    else
    {
        if (zipfi->dosDate != 0)
            zi->ci.dosDate = zipfi->dosDate;
        else
          zi->ci.dosDate = zip64local_TmzDateToDosDate(&zipfi->tmz_date);
    }

    zi->ci.flag = flagBase;
    if ((level==8) || (level==9))
      zi->ci.flag |= 2;
    if (level==2)
      zi->ci.flag |= 4;
    if (level==1)
      zi->ci.flag |= 6;
    if (password != NULL)
      zi->ci.flag |= 1;
    if (version_to_extract >= 20
            && ((zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) != 0
                || (zi->flags & ZIP_SEQUENTIAL) != 0))
      zi->ci.flag |= 8;

    zi->ci.crc32 = 0;
    zi->ci.method = method;
    zi->ci.encrypt = 0;
    zi->ci.stream_initialised = 0;
    zi->ci.pos_in_buffered_data = 0;
    zi->ci.raw = raw;
    zi->ci.pos_local_header = ZTELL64(zi->z_filefunc,zi->filestream);

    zi->ci.size_centralheader = SIZECENTRALHEADER + size_filename + size_extrafield_global + size_comment;
    zi->ci.size_centralExtraFree = 32; /* Extra space we have reserved in case we need to add ZIP64 extra info data */

    zi->ci.central_header = (char*)ALLOC((uInt)zi->ci.size_centralheader + zi->ci.size_centralExtraFree);

    zi->ci.size_centralExtra = size_extrafield_global;
    zip64local_putValue_inmemory(zi->ci.central_header,(uLong)CENTRALHEADERMAGIC,4);
    /* version info */
    zip64local_putValue_inmemory(zi->ci.central_header+4,(uLong)versionMadeBy,2);
    zip64local_putValue_inmemory(zi->ci.central_header+6,(uLong)version_to_extract,2);
    zip64local_putValue_inmemory(zi->ci.central_header+8,(uLong)zi->ci.flag,2);
    zip64local_putValue_inmemory(zi->ci.central_header+10,(uLong)zi->ci.method,2);
    zip64local_putValue_inmemory(zi->ci.central_header+12,(uLong)zi->ci.dosDate,4);
```

```
    zip64local_putValue_inmemory(zi->ci.central_header+16,(uLong)0,4); /*crc*/
    zip64local_putValue_inmemory(zi->ci.central_header+20,(uLong)0,4); /*compr size*/
    zip64local_putValue_inmemory(zi->ci.central_header+24,(uLong)0,4); /*uncompr size*/
    zip64local_putValue_inmemory(zi->ci.central_header+28,(uLong)size_filename,2);
    zip64local_putValue_inmemory(zi->ci.central_header+30,(uLong)size_extrafield_global,2);
    zip64local_putValue_inmemory(zi->ci.central_header+32,(uLong)size_comment,2);
    zip64local_putValue_inmemory(zi->ci.central_header+34,(uLong)0,2); /*disk nm start*/

    if (zipfi==NULL)
        zip64local_putValue_inmemory(zi->ci.central_header+36,(uLong)0,2);
    else
        zip64local_putValue_inmemory(zi->ci.central_header+36,(uLong)zipfi->internal_fa,2);

    if (zipfi==NULL)
        zip64local_putValue_inmemory(zi->ci.central_header+38,(uLong)0,4);
    else
        zip64local_putValue_inmemory(zi->ci.central_header+38,(uLong)zipfi->external_fa,4);

    if(zi->ci.pos_local_header >= 0xffffffff)
      zip64local_putValue_inmemory(zi->ci.central_header+42,(uLong)0xffffffff,4);
    else
      zip64local_putValue_inmemory(zi->ci.central_header+42,(uLong)zi->ci.pos_local_header - zi->add_position_when_writting_offset,4);

    for (i=0;i<size_filename;i++)
        *(zi->ci.central_header+SIZECENTRALHEADER+i) = *(filename+i);

    for (i=0;i<size_extrafield_global;i++)
        *(zi->ci.central_header+SIZECENTRALHEADER+size_filename+i) =
              *(((const char*)extrafield_global)+i);

    for (i=0;i<size_comment;i++)
        *(zi->ci.central_header+SIZECENTRALHEADER+size_filename+
              size_extrafield_global+i) = *(comment+i);
    if (zi->ci.central_header == NULL)
        return ZIP_INTERNALERROR;

    zi->ci.zip64 = zip64;
    zi->ci.totalCompressedData = 0;
    zi->ci.totalUncompressedData = 0;
    zi->ci.pos_zip64extrainfo = 0;

    err = Write_LocalFileHeader(zi, filename, size_extrafield_local,
                                extrafield_local, version_to_extract);

#ifdef HAVE_BZIP2
    zi->ci.bstream.avail_in = (uInt)0;
    zi->ci.bstream.avail_out = (uInt)Z_BUFSIZE;
    zi->ci.bstream.next_out = (char*)zi->ci.buffered_data;
    zi->ci.bstream.total_in_hi32 = 0;
    zi->ci.bstream.total_in_lo32 = 0;
    zi->ci.bstream.total_out_hi32 = 0;
    zi->ci.bstream.total_out_lo32 = 0;
#endif

    zi->ci.stream.avail_in = (uInt)0;
    zi->ci.stream.avail_out = (uInt)Z_BUFSIZE;
    zi->ci.stream.next_out = zi->ci.buffered_data;
    zi->ci.stream.total_in = 0;
    zi->ci.stream.total_out = 0;
    zi->ci.stream.data_type = Z_BINARY;

#ifdef HAVE_BZIP2
    if ((err==ZIP_OK) && (zi->ci.method == Z_DEFLATED || zi->ci.method == Z_BZIP2ED) && (!zi->ci.raw))
#else
    if ((err==ZIP_OK) && (zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
#endif
    {
        if(zi->ci.method == Z_DEFLATED)
        {
          zi->ci.stream.zalloc = (alloc_func)0;
          zi->ci.stream.zfree = (free_func)0;
```

```
            zi->ci.stream.opaque = (voidpf)0;

            if (windowBits>0)
                windowBits = -windowBits;

            err = deflateInit2(&zi->ci.stream, level, Z_DEFLATED, windowBits, memLevel, strategy);

            if (err==Z_OK)
                zi->ci.stream_initialised = Z_DEFLATED;
        }
        else if(zi->ci.method == Z_BZIP2ED)
        {
#ifdef HAVE_BZIP2
            /* Init BZip stuff here */
            zi->ci.bstream.bzalloc = 0;
            zi->ci.bstream.bzfree = 0;
            zi->ci.bstream.opaque = (voidpf)0;

            err = BZ2_bzCompressInit(&zi->ci.bstream, level, 0,35);
            if(err == BZ_OK)
              zi->ci.stream_initialised = Z_BZIP2ED;
#endif
        }

    }

#    ifndef NOCRYPT
    zi->ci.crypt_header_size = 0;
    if ((err==Z_OK) && (password != NULL))
    {
        unsigned char bufHead[RAND_HEAD_LEN];
        unsigned int sizeHead;
        zi->ci.encrypt = 1;
        zi->ci.pcrc_32_tab = get_crc_table();
        /*init_keys(password,zi->ci.keys,zi->ci.pcrc_32_tab);*/
        if (crcForCrypting == 0) {
            crcForCrypting = (uLong)zi->ci.dosDate << 16; /* ATTANTION! Without this row, you don't unpack your password protected archive in other app. */
        }
        sizeHead=crypthead(password,bufHead,RAND_HEAD_LEN,zi->ci.keys,zi->ci.pcrc_32_tab,crcForCrypting);
        zi->ci.crypt_header_size = sizeHead;

        if (ZWRITE64(zi->z_filefunc,zi->filestream,bufHead,sizeHead) != sizeHead)
                err = ZIP_ERRNO;
    }
#    endif

    if (err==Z_OK)
        zi->in_opened_file_inzip = 1;
    return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.4.45 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:

9.105.4.46 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;
```

```
  char* pNewHeader;
  char* pTmp;
  short header;
  short dataSize;

  int retVal = ZIP_OK;

  if(pData == NULL || *dataLen < 4)
    return ZIP_PARAMERROR;

  pNewHeader = (char*)ALLOC(*dataLen);
  pTmp = pNewHeader;

  while(p < (pData + *dataLen))
  {
    header = *(short*)p;
    dataSize = *(((short*)p)+1);

    if( header == sHeader ) /* Header found. */
    {
      p += dataSize + 4; /* skip it. do not copy to temp buffer */
    }
    else
    {
      /* Extra Info block should not be removed, So copy it to the temp buffer. */
      memcpy(pTmp, p, dataSize + 4);
      p += dataSize + 4;
      size += dataSize + 4;
    }

  }

  if(size < *dataLen)
  {
    /* clean old extra info block. */
    memset(pData,0, *dataLen);

    /* copy the new extra info block over the old */
    if(size > 0)
      memcpy(pData, pNewHeader, size);

    /* set the new extra info size */
    *dataLen = size;

    retVal = ZIP_OK;
  }
  else
    retVal = ZIP_ERRNO;

  TRYFREE(pNewHeader);

  return retVal;
}
```

9.105.4.47 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().

```
{
    zip64_internal* zi;
    if (file == NULL)
        return ZIP_PARAMERROR;
    zi = (zip64_internal*)file;
    zi->flags |= flags;
    // If the output is non-seekable, the data descriptor is needed.
    if ((zi->flags & ZIP_SEQUENTIAL) != 0) {
        zi->flags |= ZIP_WRITE_DATA_DESCRIPTOR;
    }
    return ZIP_OK;
}
```

Here is the caller graph for this function:

9.105.4.48 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;
```

```
    zi->ci.stream.avail_in = len;

    while ((err==ZIP_OK) && (zi->ci.stream.avail_in>0))
    {
        if (zi->ci.stream.avail_out == 0)
        {
            if (zip64FlushWriteBuffer(zi) == ZIP_ERRNO)
                err = ZIP_ERRNO;
            zi->ci.stream.avail_out = (uInt)Z_BUFSIZE;
            zi->ci.stream.next_out = zi->ci.buffered_data;
        }


        if(err != ZIP_OK)
            break;

        if ((zi->ci.method == Z_DEFLATED) && (!zi->ci.raw))
        {
            uLong uTotalOutBefore = zi->ci.stream.total_out;
            err=deflate(&zi->ci.stream,  Z_NO_FLUSH);
            if(uTotalOutBefore > zi->ci.stream.total_out)
            {
              int bBreak = 0;
              bBreak++;
            }

            zi->ci.pos_in_buffered_data += (uInt)(zi->ci.stream.total_out - uTotalOutBefore) ;
        }
        else
        {
            uInt copy_this,i;
            if (zi->ci.stream.avail_in < zi->ci.stream.avail_out)
                copy_this = zi->ci.stream.avail_in;
            else
                copy_this = zi->ci.stream.avail_out;

            for (i = 0; i < copy_this; i++)
                *(((char*)zi->ci.stream.next_out)+i) =
                    *(((const char*)zi->ci.stream.next_in)+i);
            {
                zi->ci.stream.avail_in -= copy_this;
                zi->ci.stream.avail_out-= copy_this;
                zi->ci.stream.next_in+= copy_this;
                zi->ci.stream.next_out+= copy_this;
                zi->ci.stream.total_in+= copy_this;
                zi->ci.stream.total_out+= copy_this;
                zi->ci.pos_in_buffered_data += copy_this;
            }
        }
    }/* while(...) */
  }

  return err;
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.105.5 Variable Documentation

9.105.5.1 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.

9.106 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)

9.106.1 Data Structure Documentation

9.106.1.1 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	

9.106.1.2 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	

9.106.2 Macro Definition Documentation

9.106.2.1 #define APPEND_STATUS_ADDINZIP (2)

Definition at line 125 of file zip.h.

Referenced by QuaZip::open(), and zipOpen3().

9.106.2.2 #define APPEND_STATUS_CREATE (0)

Definition at line 123 of file zip.h.

Referenced by QuaZip::open(), and zipOpen3().

9.106.2.3 #define APPEND_STATUS_CREATEAFTER (1)

Definition at line 124 of file zip.h.

Referenced by QuaZip::open(), and zipOpen3().

9.106.2.4 #define DEF_MEM_LEVEL MAX_MEM_LEVEL

Definition at line 94 of file zip.h.

Referenced by zipOpenNewFileInZip(), zipOpenNewFileInZip2(), zipOpenNewFileInZip2_64(), and zipOpenNewFileInZip64().

9.106.2.5 #define Z_BZIP2ED 12

Definition at line 67 of file zip.h.

9.106.2.6 #define ZIP_AUTO_CLOSE 0x1u

Definition at line 86 of file zip.h.

Referenced by LoadCentralDirectoryRecord(), QuaZip::open(), zipClose(), and zipOpen3().

9.106.2.7 #define ZIP_BADZIPFILE (-103)

Definition at line 82 of file zip.h.

Referenced by LoadCentralDirectoryRecord(), and zipCloseFileInZipRaw64().

9.106.2.8 #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().

9.106.2.9 #define ZIP_EOF (0)

Definition at line 79 of file zip.h.

Referenced by zip64local_getByte().

9.106.2.10 #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().

9.106.2.11 #define ZIP_INTERNALERROR (-104)

Definition at line 83 of file zip.h.

Referenced by add_data_in_datablock(), and zipOpenNewFileInZip4_64().

9.106.2.12 #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().

9.106.2.13 #define ZIP_PARAMERROR (-102)

Definition at line 81 of file zip.h.

Referenced by zipClearFlags(), zipClose(), zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), zipRemoveExtraInfoBlock(), zipSetFlags(), and zipWriteInFileInZip().

9.106.2.14 #define ZIP_SEQUENTIAL 0x2u

Definition at line 87 of file zip.h.

Referenced by QuaZip::open(), zipClearFlags(), zipCloseFileInZipRaw64(), zipOpenNewFileInZip4_64(), and zipSetFlags().

9.106.2.15 #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().

9.106.3 Typedef Documentation

9.106.3.1 typedef struct tm_zip_s tm_zip

9.106.3.2 typedef const char* zipcharpc

Definition at line 120 of file zip.h.

9.106.3.3 typedef voidp zipFile

Definition at line 75 of file zip.h.

9.106.4 Function Documentation

9.106.4.1 zipFile ZEXPORT zipOpen _Z_OF ((voidpf file, int append))

9.106.4.2 zipFile ZEXPORT zipOpen2 _Z_OF ((voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc_def *pzlib_filefunc_def))

9.106.4.3 zipFile ZEXPORT zipOpen2_64 _Z_OF ((voidpf file, int append, zipcharpc *globalcomment, zlib_filefunc64_def *pzlib_filefunc_def))

9.106.4.4 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))

9.106.4.5 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))

9.106.4.6 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))

9.106.4.7 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))

9.106.4.8 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))

9.106.4.9 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))

9.106.4.10 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))

9.106.4.11 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))

9.106.4.12 int ZEXPORT **zipWriteInFileInZip _Z_OF** ((**zipFile** file, const void ∗buf, unsigned len))

9.106.4.13 int ZEXPORT **zipCloseFileInZip _Z_OF** ((**zipFile** file))

9.106.4.14 int ZEXPORT **zipCloseFileInZipRaw _Z_OF** ((**zipFile** file, uLong uncompressed_size, uLong crc32))

9.106.4.15 int ZEXPORT **zipCloseFileInZipRaw64 _Z_OF** ((**zipFile** file, **ZPOS64_T** uncompressed_size, uLong crc32))

9.106.4.16 int ZEXPORT **zipClose _Z_OF** ((**zipFile** file, const char ∗global_comment))

9.106.4.17 int ZEXPORT **zipRemoveExtraInfoBlock _Z_OF** ((char ∗pData, int ∗dataLen, short sHeader))

9.106.4.18 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;
```

```
    if (file == NULL)
        return ZIP_PARAMERROR;
    zi = (zip64_internal*)file;
    zi->flags &= ~flags;
    // If the data descriptor is not written, we can't use a non-seekable output.
    if ((zi->flags & ZIP_WRITE_DATA_DESCRIPTOR) == 0) {
        zi->flags &= ~ZIP_SEQUENTIAL;
    }
    return ZIP_OK;
}
```

Here is the caller graph for this function:

9.106.4.19 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().

```
{
    zip64_internal ziinit;
    zip64_internal* zi;
    int err=ZIP_OK;

    ziinit.flags = flags;
    ziinit.z_filefunc.zseek32_file = NULL;
    ziinit.z_filefunc.ztell32_file = NULL;
    if (pzlib_filefunc64_32_def==NULL)
        fill_qiodevice64_filefunc(&ziinit.z_filefunc.zfile_func64);
    else
        ziinit.z_filefunc = *pzlib_filefunc64_32_def;

    ziinit.filestream = ZOPEN64(ziinit.z_filefunc,
                    file,
                    (append == APPEND_STATUS_CREATE) ?
                    (ZLIB_FILEFUNC_MODE_READ | ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_CREATE) :
                      (ZLIB_FILEFUNC_MODE_READ | ZLIB_FILEFUNC_MODE_WRITE | ZLIB_FILEFUNC_MODE_EXISTING));

    if (ziinit.filestream == NULL)
        return NULL;

    if (append == APPEND_STATUS_CREATEAFTER)
        ZSEEK64(ziinit.z_filefunc,ziinit.filestream,0,SEEK_END);

```

```
    ziinit.begin_pos = ZTELL64(ziinit.z_filefunc,ziinit.filestream);
    ziinit.in_opened_file_inzip = 0;
    ziinit.ci.stream_initialised = 0;
    ziinit.number_entry = 0;
    ziinit.add_position_when_writting_offset = 0;
    init_linkedlist(&(ziinit.central_dir));



    zi = (zip64_internal*)ALLOC(sizeof(zip64_internal));
    if (zi==NULL)
    {
        if ((ziinit.flags & ZIP_AUTO_CLOSE) != 0) {
            ZCLOSE64(ziinit.z_filefunc,ziinit.filestream);
        } else {
            ZFAKECLOSE64(ziinit.z_filefunc,ziinit.filestream);
        }
        return NULL;
    }

    /* now we add file in a zipfile */
#    ifndef NO_ADDFILEINEXISTINGZIP
    ziinit.globalcomment = NULL;
    if (append == APPEND_STATUS_ADDINZIP)
    {
      /* Read and Cache Central Directory Records */
      err = LoadCentralDirectoryRecord(&ziinit);
    }

    if (globalcomment)
    {
      *globalcomment = ziinit.globalcomment;
    }
#    endif /* !NO_ADDFILEINEXISTINGZIP*/

    if (err != ZIP_OK)
    {
#    ifndef NO_ADDFILEINEXISTINGZIP
        TRYFREE(ziinit.globalcomment);
#    endif /* !NO_ADDFILEINEXISTINGZIP*/
        TRYFREE(zi);
        return NULL;
    }
    else
    {
        *zi = ziinit;
        return (zipFile)zi;
    }
}
```

Here is the call graph for this function:

Here is the caller graph for this function:

9.106.4.20 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().

```
{
    zip64_internal* zi;
    if (file == NULL)
        return ZIP_PARAMERROR;
    zi = (zip64_internal*)file;
    zi->flags |= flags;
    // If the output is non-seekable, the data descriptor is needed.
    if ((zi->flags & ZIP_SEQUENTIAL) != 0) {
        zi->flags |= ZIP_WRITE_DATA_DESCRIPTOR;
    }
    return ZIP_OK;
}
```

Here is the caller graph for this function:

9.107 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_UP + 1
- static const int DPI_CUST_IDX = KeyAction::DPI_CUSTOM + DPI_OFFSET

9.107.1 Variable Documentation

9.107.1.1 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().

9.107.1.2 const int DPI_OFFSET = -KeyAction::DPI_UP + 1 [static]

Definition at line 6 of file rebindwidget.cpp.

Referenced by RebindWidget::applyChanges(), and RebindWidget::setSelection().

9.108 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

9.109 src/ckb/rlistwidget.cpp File Reference

```
#include <QUuid>
#include "rlistwidget.h"
```

Include dependency graph for rlistwidget.cpp:

9.110 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

9.111 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

- static QStringList modKeys
- static QStringList modNames

9.111.1 Function Documentation

9.111.1.1 static QString right (const QString & *left*) [inline], [static]

Definition at line 118 of file settingswidget.cpp.

Referenced by GradientDialogWidget::mouseMoveEvent(), 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:

9.111.2 Variable Documentation

9.111.2.1 QString devpath

Definition at line 4 of file kbmanager.cpp.

Referenced by KbManager::scanKeyboards().

9.111.2.2 QStringList modKeys [static]

Definition at line 13 of file settingswidget.cpp.

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

9.111.2.3 QStringList modNames [static]

Definition at line 13 of file settingswidget.cpp.

Referenced by SettingsWidget::SettingsWidget().

9.112 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

Index

~AnimAddDialog
AnimAddDialog, 21
~AnimScript
AnimScript, 30
~AnimSettingDialog
AnimSettingDialog, 64
~CkbSettings
CkbSettings, 84
~CkbSettingsWriter
CkbSettingsWriter, 97
~ColorMap
ColorMap, 109
~ExtraSettingsWidget
ExtraSettingsWidget, 120
~FwUpgradeDialog
FwUpgradeDialog, 129
~GradientDialog
GradientDialog, 145
~KPerfWidget
KPerfWidget, 567
~Kb
Kb, 191
~KbAnimWidget
KbAnimWidget, 268
~KbBindWidget
KbBindWidget, 315
~KbFirmware
KbFirmware, 323
~KbLight
KbLight, 337
~KbLightWidget
KbLightWidget, 365
~KbMode
KbMode, 390
~KbProfileDialog
KbProfileDialog, 455
~KbWidget
KbWidget, 466
~KeyAction
KeyAction, 492
~LayoutDialog
LayoutDialog, 575
~MPerfWidget
MPerfWidget, 608
~MacroReader
MacroReader, 578
~MainWindow
MainWindow, 587
~ModeSelectDialog
ModeSelectDialog, 600
~QuaGzipFile
QuaGzipFile, 643
~QuaZIODevice
QuaZIODevice, 649
~QuaZip
QuaZip, 658
~QuaZipDir
QuaZipDir, 689
~QuaZipFile
QuaZipFile, 709
~QuaZipFilePrivate
QuaZipFilePrivate, 740
~RebindWidget
RebindWidget, 766
~SGroup
SGroup, 813
~SettingsWidget
SettingsWidget, 804
_LAYOUT_MAX
KeyMap, 520
_MODEL_MAX
KeyMap, 521
_FILE_OFFSET_BIT
ioapi.h, 942
_LARGEFILE64_SOURCE
ioapi.h, 943
_Z_OF
ioapi.h, 945
unzip.c, 993
unzip.h, 1048, 1049
zip.c, 1067
zip.h, 1120, 1121
__USE_FILE_OFFSET64
ioapi.h, 942
__USE_LARGEFILE64
ioapi.h, 942
_alpha
ColorButton, 107
GradientButton, 141
_angleSnap
KbPerf, 432
_anim
AnimSettingDialog, 75
_animList
KbLight, 360
_animMap
KbLight, 361
_backing

CkbSettingsWriter, 98
_bigIcons
ColorButton, 107
_bind
KbBind, 311
KbMode, 398
_bindMap
KeyWidget, 560
_checkUpdates
KbFirmware, 324
_color
ColorButton, 107
_colorBuffer
AnimScript, 54
_colorMap
KbLight, 361
KeyWidget, 560
_colors
AnimScript, 54
ColorMap, 116
GradientDialogWidget, 168
_count
ColorMap, 116
_curDpi
KbPerf, 406
_current
GradientDialogWidget, 168
_currentMode
Kb, 232
KbProfile, 452
_currentProfile
Kb, 232
_daemonVersion
KbManager, 385
_delay
Kb, 232
_devParent
KbBind, 311
_devices
KbManager, 385
_dimming
KbLight, 361
_displayColorMap
KeyWidget, 560
_dither
Kb, 232
_dpiIndicator
KbPerf, 432
_eventTimer
KbManager, 385
_fileForBoard
KbFirmware, 324
_frameRate
Kb, 232
_friendlyName
Key, 486
_globalRemap
KbBind, 311
_globalSettings
ckbsettings.cpp, 848
_guiVersion
KbManager, 385
_guid
KbAnim, 262
_hwProfile
Kb, 232
_iOpacity
KbPerf, 432
_id
KbMode, 399
KbProfile, 452
_indicatorList
KbLight, 361
_indicatorMap
KbLight, 361
_indicators
KeyWidget, 560
_info
AnimScript, 54
_isActive
KbAnim, 262
_isActiveKp
KbAnim, 262
_kbManager
KbManager, 385
_keyMap
KbProfile, 452
_keyNames
ColorMap, 116
_keys
AnimAddDialog, 24
AnimScript, 54
KbAnim, 262
KeyMap, 540
_latestForBoard
KbFirmware, 325
_layout
Kb, 233
_liftHeight
KbPerf, 432
_light
AnimAddDialog, 24
KbMode, 399
_map
AnimScript, 54
KbAnim, 262
KbBind, 311
KbLight, 361
_mapCount
ColorMap, 116
_mode
KbAnim, 262
_modeList
ModeSelectDialog, 602
_model
Kb, 233

_modes
KbProfile, 452
_monochrome
KeyWidget, 560
_mouseAccel
Kb, 233
_name
KbAnim, 262
KbMode, 399
KbProfile, 452
_needsMapRefresh
KbLight, 361
_needsSave
Kb, 233
KbAnim, 262
KbBind, 311
KbLight, 361
KbMode, 399
KbPerf, 432
KbProfile, 452
_needsUpdate
KbBind, 312
KbPerf, 432
_opacity
KbAnim, 263
_paramValues
AnimScript, 54
_parameters
KbAnim, 263
_patch
keymap.cpp, 898
_path
AnimScript, 54
_perf
KbMode, 399
_presetValues
AnimScript, 55
_presets
AnimScript, 55
_previewAnim
KbLight, 361
_profiles
Kb, 233
_qColorMap
KbLight, 361
_removals
CkbSettingsWriter, 98
_rgbMode
KeyWidget, 560
_scanTimer
KbManager, 385
_script
KbAnim, 263
_scriptGuid
KbAnim, 263
_scriptName
KbAnim, 263
_scrollSpeed
Kb, 233
_setLabel
ColorButton, 107
_settings
SGroup, 813
_shareDimming
kblight.cpp, 879
_start
KbLight, 362
_stops
GradientButton, 141
GradientDialogWidget, 168
_storageName
Key, 486
_tempParameters
KbAnim, 263
_updates
CkbSettingsWriter, 98
_value
KeyAction, 514
_winLock
KbBind, 312
_xyLink
MPerfWidget, 618

ADD
KeyWidget, 545
ALWAYS_OFF
KPerfWidget, 566
ALWAYS_ON
KPerfWidget, 566
ALLOC
unzip.c, 992
zip.c, 1064
ANIM_KRSTOP
keyaction.cpp, 891
ANIM_ONCE
keyaction.cpp, 891
ANSI_ENTER_H
keymap.cpp, 895
ANSI_ENTER_W
keymap.cpp, 895
ANSI_ENTER_X
keymap.cpp, 895
ANSI_ENTER_Y
keymap.cpp, 895
ANSI_LSHIFT_W
keymap.cpp, 895
ANSI_LSHIFT_X
keymap.cpp, 896
APPEND_STATUS_CREATE
zip.h, 1118
AUTO_CHECK_TIME
kbfirmware.cpp, 875
action
KbBind, 287
activeDevices
kb.cpp, 866
activeLights

kblight.cpp, 879
Add
KbAnim, 239
add_data_in_datablock
zip.c, 1067
addAnim
KbAnimWidget, 268
KbLight, 337
addCurrentFileToDirectoryMap
QuaZipPrivate, 754
addDevice
MainWindow, 588
addNewModeItem
KbWidget, 467
addNewProfileItem
KbProfileDialog, 455
addPreset
GradientDialog, 145
adjustDisplay
KeyAction, 492
advance
AnimScript, 31
alloc
ColorMap, 110
allocate_new_datablock
zip.c, 1068
allowAlpha
ColorButton, 101, 102
angleDialChanged
AnimSettingDialog, 64
angleDialMapper
AnimSettingDialog, 75
angleFlip
animsettingdialog.cpp, 824
angleSnap
KbPerf, 406, 407
angleSpinnerChanged
AnimSettingDialog, 65
angleSpinnerMapper
AnimSettingDialog, 75
angleSpinners
AnimSettingDialog, 75
AnimScript::Param, 29, 822
animAction
KeyAction, 493
AnimAddDialog, 19
~AnimAddDialog, 21
_keys, 24
_light, 24
AnimAddDialog, 21
AnimAddDialog, 21
chosenPreset, 21
chosenScript, 22
on_animBox_activated, 22
on_presetBox_activated, 23
on_previewBox_clicked, 24
scripts, 24
showPreview, 24
ui, 25
animChanged
KbAnimWidget, 269
animInfo
KeyAction, 494
animKeypress
KbLight, 338
AnimList
KbLight, 336
animList
KbLight, 338, 339
AnimScript, 25
~AnimScript, 30
_colorBuffer, 54
_colors, 54
_info, 54
_keys, 54
_map, 54
_paramValues, 54
_path, 54
_presetValues, 55
_presets, 55
advance, 31
AnimScript, 30
AnimScript, 30
author, 31
begin, 32
colors, 33
copy, 33
copyright, 34
count, 34
description, 35
durationMsec, 55
end, 35
firstFrame, 55
frame, 36
guid, 37
hasFrame, 37
hasKeypress, 38
hasParam, 38
inFrame, 55
init, 39
initialized, 55
KP_NAME, 55
KP_NONE, 55
KP_POSITION, 55
keypress, 40
lastFrame, 56
license, 41
list, 41
load, 42
minX, 56
minY, 56
name, 45
param, 45
paramIterator, 47
parameters, 46
path, 47

preset, 48
PresetValue, 30
presets, 48
printParams, 49
process, 56
readAnyFrame, 56
readFrame, 56
readProcess, 49
repeatMsec, 56
retrigger, 50
scan, 51
scripts, 56
setDuration, 52
stop, 52
stopped, 56
version, 53
year, 53
AnimSettingDialog, 57
~AnimSettingDialog, 64
_anim, 75
angleDialChanged, 64
angleDialMapper, 75
angleSpinnerChanged, 65
angleSpinnerMapper, 75
angleSpinners, 75
AnimSettingDialog, 59
AnimSettingDialog, 59
hLine, 66
hasRepeat, 76
kpStopCheck, 76
lastDuration, 76
name, 67
newDuration, 67
on_delayBox_valueChanged, 68
on_kpDelayBox_valueChanged, 69
on_kpReleaseBox_stateChanged, 69
on_kpRepeatBox_valueChanged, 70
on_repeatBox_valueChanged, 70
settingWidgets, 76
stopCheck, 76
ui, 76
updateMapper, 76
updateParam, 71
updateStops, 74
animation
KeyWidget, 560
animations
KbAnimWidget, 281
animscript.cpp
ONE_DAY, 821
urlParam, 821
animsettingdialog.cpp
angleFlip, 824
appShare
main.cpp, 918
mainwindow.cpp, 923
append
KbProfile, 440
appendProfile
Kb, 194
applyChanges
RebindWidget, 766
applyIndicators
KbPerf, 407
atEnd
QuaZipFile, 709
author
AnimScript, 31
autoClose
QuaZipPrivate, 758
AutoRun, 77
available, 77
disable, 78
enable, 78
isEnabled, 79
once, 80
autoSave
Kb, 194
autorun.cpp
settingPath, 826
available
AutoRun, 77

BOTH
KPerfWidget, 566
BUFREADCOMMENT
unzip.c, 992
zip.c, 1065
backing
CkbSettings, 95
base
KbLight, 339
begin
AnimScript, 32
beginGroup
CkbSettings, 84
bigIcons
ColorButton, 102, 103
bind
KbBindWidget, 320
KbMode, 390
KbPerf, 408
RebindWidget, 791
bindAction
KbBind, 288
BindMap
KeyWidget, 544
bindMap
KeyWidget, 545
blend
KbAnim, 242
blendAdd
kbanim.cpp, 868
blendDivide
kbanim.cpp, 868
blendFunc
kbanim.cpp, 868

blendMultiply
kbanim.cpp, 868
blendNormal
kbanim.cpp, 868
blendSubtract
kbanim.cpp, 869
blob
FwUpgradeDialog, 134
boxXChanged
MPerfWidget, 608
boxXMapper
MPerfWidget, 618
boxYChanged
MPerfWidget, 609
boxYMapper
MPerfWidget, 618
builtIn
GradientDialog::Preset, 621
buttonMapper1
MPerfWidget, 618
buttonMapper2
MPerfWidget, 618
byPosition
KeyMap, 522
bytesAvailable
QuaZipFile, 710

CAPS
KbPerf, 404
CENTRALHEADERMAGIC
zip.c, 1065
CKB_CONTAINER
ckb-anim.h, 832
CKB_COPYRIGHT
ckb-anim.h, 832
CKB_DESCRIPTION
ckb-anim.h, 832
CKB_DOWN
ckb-anim.h, 832
CKB_GRAD_MAX
ckb-anim.h, 832
CKB_GUID
ckb-anim.h, 833
CKB_KEYCLEAR
ckb-anim.h, 833
CKB_KEYNAME_MAX
ckb-anim.h, 833
CKB_KP_NAME
ckb-anim.h, 833
CKB_KP_NONE
ckb-anim.h, 833
CKB_KP_POSITION
ckb-anim.h, 833
CKB_KPMODE
ckb-anim.h, 833
CKB_LICENSE
ckb-anim.h, 833
CKB_LIVEPARAMS
ckb-anim.h, 833
CKB_MAX_WORD
ckb-anim.h, 833
CKB_NAME
ckb-anim.h, 833
CKB_PARAM
ckb-anim.h, 834
CKB_PARAM_AGRADIENT
ckb-anim.h, 834
CKB_PARAM_ANGLE
ckb-anim.h, 834
CKB_PARAM_ARGB
ckb-anim.h, 834
CKB_PARAM_BOOL
ckb-anim.h, 834
CKB_PARAM_DOUBLE
ckb-anim.h, 834
CKB_PARAM_GRADIENT
ckb-anim.h, 834
CKB_PARAM_LABEL
ckb-anim.h, 834
CKB_PARAM_LONG
ckb-anim.h, 834
CKB_PARAM_RGB
ckb-anim.h, 834
CKB_PARAM_STRING
ckb-anim.h, 834
CKB_PARSE_AGRADIENT
ckb-anim.h, 835
CKB_PARSE_ANGLE
ckb-anim.h, 835
CKB_PARSE_ARGB
ckb-anim.h, 835
CKB_PARSE_BOOL
ckb-anim.h, 835
CKB_PARSE_DOUBLE
ckb-anim.h, 835
CKB_PARSE_GRADIENT
ckb-anim.h, 835
CKB_PARSE_LONG
ckb-anim.h, 835
CKB_PARSE_RGB
ckb-anim.h, 835
CKB_PARSE_STRING
ckb-anim.h, 835
CKB_PREEMPT
ckb-anim.h, 835
CKB_PRESET_END
ckb-anim.h, 836
CKB_PRESET_PARAM
ckb-anim.h, 836
CKB_PRESET_START
ckb-anim.h, 836
CKB_REAL_ANGLE
ckb-anim.h, 836
CKB_REPEAT
ckb-anim.h, 836
CKB_TIME_ABSOLUTE
ckb-anim.h, 836

CKB_TIME_DURATION
ckb-anim.h, 836
CKB_TIMEMODE
ckb-anim.h, 836
CKB_UP
ckb-anim.h, 836
CKB_VERSION
ckb-anim.h, 836
CRC32
crypt.h, 936
cacheWritesInProgress
ckbsettings.cpp, 847
ckbsettingswriter.cpp, 851
calculate
QuaAdler32, 632
QuaChecksum32, 635
QuaCrc32, 638
call_zopen64
qioapi.cpp, 950
call_zseek64
qioapi.cpp, 950
call_ztell64
qioapi.cpp, 950
CaseSensitivity
QuaZip, 657
caseSensitivity
QuaZipDir, 690
QuaZipFilePrivate, 741
catchUp
KbAnim, 243
cd
QuaZipDir, 690
cdUp
QuaZipDir, 692
change
RListWidget, 796
changeAnim
KbLightWidget, 366
changeAnimKeys
KbLightWidget, 366
changeColor
KbLightWidget, 367
checkFwUpdates
MainWindow, 589
checkUpdates
KbFirmware, 326
checksum
QuaAdler32, 633
QuaCrc32, 639
childGroups
CkbSettings, 84
childKeys
CkbSettings, 85
chosenPreset
AnimAddDialog, 21
chosenScript
AnimAddDialog, 22
ckb-anim.h
CKB_CONTAINER, 832
CKB_COPYRIGHT, 832
CKB_DESCRIPTION, 832
CKB_DOWN, 832
CKB_GRAD_MAX, 832
CKB_GUID, 833
CKB_KEYCLEAR, 833
CKB_KEYNAME_MAX, 833
CKB_KP_NAME, 833
CKB_KP_NONE, 833
CKB_KP_POSITION, 833
CKB_KPMODE, 833
CKB_LICENSE, 833
CKB_LIVEPARAMS, 833
CKB_MAX_WORD, 833
CKB_NAME, 833
CKB_PARAM, 834
CKB_PARAM_AGRADIENT, 834
CKB_PARAM_ANGLE, 834
CKB_PARAM_ARGB, 834
CKB_PARAM_BOOL, 834
CKB_PARAM_DOUBLE, 834
CKB_PARAM_GRADIENT, 834
CKB_PARAM_LABEL, 834
CKB_PARAM_LONG, 834
CKB_PARAM_RGB, 834
CKB_PARAM_STRING, 834
CKB_PARSE_AGRADIENT, 835
CKB_PARSE_ANGLE, 835
CKB_PARSE_ARGB, 835
CKB_PARSE_BOOL, 835
CKB_PARSE_DOUBLE, 835
CKB_PARSE_GRADIENT, 835
CKB_PARSE_LONG, 835
CKB_PARSE_RGB, 835
CKB_PARSE_STRING, 835
CKB_PREEMPT, 835
CKB_PRESET_END, 836
CKB_PRESET_PARAM, 836
CKB_PRESET_START, 836
CKB_REAL_ANGLE, 836
CKB_REPEAT, 836
CKB_TIME_ABSOLUTE, 836
CKB_TIME_DURATION, 836
CKB_TIMEMODE, 836
CKB_UP, 836
CKB_VERSION, 836
ckb_alpha_blend, 837
ckb_frame, 837
ckb_getline, 837
ckb_grad_color, 838
ckb_info, 838
ckb_init, 839
ckb_keypress, 839
ckb_parameter, 839
ckb_read_params, 840
ckb_scan_grad, 840
ckb_start, 841

ckb_time, 841
FALSE, 836
M_PI, 836
main, 842
printurl, 844
TRUE, 837
urldecode, 844
ckb_gradient, 831
ckb_key, 830
ckb_runctx, 830
ckb_alpha_blend
ckb-anim.h, 837
ckb_frame
ckb-anim.h, 837
ckb_getline
ckb-anim.h, 837
ckb_grad_color
ckb-anim.h, 838
ckb_info
ckb-anim.h, 838
ckb_init
ckb-anim.h, 839
ckb_keypress
ckb-anim.h, 839
ckb_parameter
ckb-anim.h, 839
ckb_read_params
ckb-anim.h, 840
ckb_scan_grad
ckb-anim.h, 840
ckb_start
ckb-anim.h, 841
ckb_time
ckb-anim.h, 841
ckbDaemonVersion
KbManager, 377
ckbDaemonVersionF
KbManager, 377
ckbGuiVersion
KbManager, 378
ckbGuiVersionF
KbManager, 378
CkbSettings, 81
~CkbSettings, 84
backing, 95
beginGroup, 84
childGroups, 84
childKeys, 85
CkbSettings, 83, 84
CkbSettings, 83, 84
cleanUp, 86
contains, 87
containsGroup, 87
endGroup, 88
get, 88
groups, 95
isBusy, 89
pwd, 90, 91
remove, 92
removeCache, 96
set, 92
setValue, 93
value, 94
writeCache, 96
CkbSettingsWriter, 96
~CkbSettingsWriter, 97
_backing, 98
_removals, 98
_updates, 98
CkbSettingsWriter, 97
CkbSettingsWriter, 97
run, 98
ckbVersion
KbFirmware::FW, 126
ckbsettings.cpp
_globalSettings, 848
cacheWritesInProgress, 847
globalCache, 848
globalSettings, 847
globalThread, 848
lockMutex, 846
lockMutexCache, 846
lockMutexStatic, 846
lockMutexStatic2, 846
settingsCacheMutex, 848
settingsMutex, 848
ckbsettingswriter.cpp
cacheWritesInProgress, 851
lockMutex, 850
lockMutexCache, 850
lockMutexStatic, 850
lockMutexStatic2, 850
settingsCacheMutex, 851
settingsMutex, 851
cleanBlob
FwUpgradeDialog, 129
cleanUp
CkbSettings, 86
cleanup
MainWindow, 589
clear
ColorMap, 110
clearAnimation
KeyWidget, 545
clearDirectoryMap
QuaZipPrivate, 755
clearSelection
KbAnimWidget, 269
KeyWidget, 546
close
KbLight, 340
QuaGzipFile, 643
QuaZIODevice, 649
QuaZip, 659
QuaZipFile, 710
closeAction

MainWindow, 597
closeEvent
FwUpgradeDialog, 130
MainWindow, 590
cmd
Kb, 233
cmdpath
Kb, 233
color
ColorButton, 103, 104
KbLight, 341, 342
colorAt
GradientDialogWidget, 157
ColorButton, 98
_alpha, 107
_bigIcons, 107
_color, 107
_setLabel, 107
allowAlpha, 101, 102
bigIcons, 102, 103
color, 103, 104
ColorButton, 101
colorChanged, 104
ColorButton, 101
pickColor, 105
setLabel, 105, 106
updateImage, 106
colorChanged
ColorButton, 104
GradientDialog, 146
MPerfWidget, 610
colorClicked
MPerfWidget, 611
colorForName
ColorMap, 111, 112
colorLink
MPerfWidget, 619
ColorMap, 107
~ColorMap, 109
_colors, 116
_count, 116
_keyNames, 116
_mapCount, 116
alloc, 110
clear, 110
colorForName, 111, 112
ColorMap, 109
ColorMap, 109
colors, 112, 113
count, 113
deAlloc, 113
init, 114
keyNames, 115
operator=, 115
colorMap
KbLight, 342
KeyWidget, 546
colormap.cpp
qs_strcmp, 853
colormap.h
QColorMap, 855
QColorMapIterator, 855
QMutableColorMapIterator, 855
colors
AnimScript, 33
ColorMap, 112, 113
CommandLineBackground
main.cpp, 918
CommandLineClose
main.cpp, 918
CommandLineError
main.cpp, 918
CommandLineHelpRequested
main.cpp, 918
CommandLineOK
main.cpp, 918
CommandLineVersionRequested
main.cpp, 918
CommandLineParseResults
main.cpp, 918
comment
QuaZipFileInfo, 729
QuaZipFileInfo64, 736
QuaZipNewInfo, 750
QuaZipPrivate, 758
commentCodec
QuaZipPrivate, 758
commitParams
KbAnim, 245
compressDir
JlCompress, 170
compressFile
JlCompress, 171, 172
compressFiles
JlCompress, 173
compressSubDir
JlCompress, 174
compressedSize
QuaZipFileInfo, 729
QuaZipFileInfo64, 736
configLabel
mainwindow.cpp, 923
Constants
QuaZip, 657
contains
CkbSettings, 87
KeyMap, 522
containsGroup
CkbSettings, 87
convertCaseSensitivity
QuaZip, 660
convertMacroBox
RebindWidget, 768
copy
AnimScript, 33
copyData

JlCompress.cpp, 947
copyright
AnimScript, 34
count
AnimScript, 34
ColorMap, 113
KeyMap, 523
QuaZipDir, 692
crc
QuaZipFileInfo, 729
QuaZipFileInfo64, 736
QuaZipFilePrivate, 741
crypt.h
CRC32, 936
decrypt_byte, 937
init_keys, 937
update_keys, 938
zdecode, 937
zencode, 937
csDefault
QuaZip, 657
csInsensitive
QuaZip, 657
csSensitive
QuaZip, 657
csize
QuaZipFile, 711
curDpi
KbPerf, 409, 410
curDpiIdx
KbPerf, 411
curfile64_info, 1063
current
GradientDialogWidget, 158
KbAnimWidget, 281
currentBind
Kb, 194
currentChanged
GradientDialog, 147
GradientDialogWidget, 158
currentData
RListWidget, 799
currentLight
Kb, 195
currentMode
Kb, 196
KbProfile, 440, 441
KbWidget, 483
currentPerf
Kb, 196
currentPreset
GradientDialog, 153
currentProfile
Kb, 197
currentSelection
KbBindWidget, 320
KbLightWidget, 372

d
QuaGzipFile, 646
QuaZIODevice, 653
QuaZipDir, 703
DE
KeyMap, 520
DK
KeyMap, 520
DATA_ROLE
RListWidget, 799
DEF_MEM_LEVEL
zip.h, 1119
DESCRIPTORHEADERMAGIC
zip.c, 1065
DIALOG_HEIGHT_MAX
fwupgradedialog.cpp, 858
DIALOG_HEIGHT_MIN
fwupgradedialog.cpp, 859
DIALOG_WIDTH
fwupgradedialog.cpp, 859
DPI_BASE
mperfwidget.cpp, 933
DPI_COUNT
KbPerf, 432
MPerfWidget, 619
DPI_CUST_IDX
rebindwidget.cpp, 1125
DPI_CUSTOM
KeyAction, 514
DPI_DOWN
KeyAction, 514
DPI_MAX
KbPerf, 433
MPerfWidget, 619
DPI_MIN
KbPerf, 433
MPerfWidget, 619
DPI_OFFSET
rebindwidget.cpp, 1125
DPI_SNIPER
KeyAction, 514
DPI_UP
KeyAction, 514
dataDescriptorWritingEnabled
QuaZipPrivate, 758
dataForBoard
KbFirmware, 327
dateTime
QuaZipFileInfo, 729
QuaZipFileInfo64, 736
QuaZipNewInfo, 750
deAlloc
ColorMap, 113
decrypt_byte
crypt.h, 937
defaultAction
KbBind, 288
KeyAction, 494
defaultFileNameCodec

QuaZipPrivate, 759
deleteHw
Kb, 197
deletePrevious
Kb, 198
description
AnimScript, 35
devParent
KbBind, 289
devUpdate
KbWidget, 467
device
KbProfileDialog, 461
KbWidget, 483
devices
KbManager, 379
devpath
Kb, 233
kbmanager.cpp, 882
settingswidget.cpp, 1129
didLoad
KbBind, 290
KbLight, 343
KbPerf, 412
didUpdateSelection
KbAnimWidget, 270
dimming
KbLight, 343
dirName
QuaZipDir, 693
directoryCaseInsensitive
QuaZipPrivate, 759
directoryCaseSensitive
QuaZipPrivate, 759
disable
AutoRun, 78
disableAppNap
media_mac.m, 928
diskNumberStart
QuaZipFileInfo, 729
QuaZipFileInfo64, 736
displayColorMap
KeyWidget, 546
dither
Kb, 198, 199
Divide
KbAnim, 240
doUpdate
KbMode, 390
downloadFinished
KbFirmware, 328
downloaded
KbFirmware, 327
dpi
KbPerf, 412, 413
dpiAction
KeyAction, 495
dpiClr
KbPerf, 433
dpiColor
KbPerf, 414
dpiCurIdx
KbPerf, 433
dpiCurX
KbPerf, 433
dpiCurY
KbPerf, 433
dpiDown
KbPerf, 414
dpiEnabled
KbPerf, 415
dpiExp
mperfwidget.cpp, 932
dpiIndicator
KbPerf, 416
dpiInfo
KeyAction, 496
dpiLastIdx
KbPerf, 433
dpiLog
mperfwidget.cpp, 932
dpiOn
KbPerf, 433
dpiRound
mperfwidget.cpp, 933
dpiUp
KbPerf, 416
dpiX
KbPerf, 433
dpiY
KbPerf, 434
dragged
RListWidget, 799
drawInfo
KeyWidget, 546
driverName
KeyAction, 496
duplicateAnim
KbAnimWidget, 270
KbLight, 344
durationMsec
AnimScript, 55

ES
KeyMap, 520
EU
KeyMap, 520
EU_DVORAK
KeyMap, 520
ENDHEADERMAGIC
zip.c, 1065
ENUM_C
media.h, 925
ENUM_END_C
media.h, 925
EXTERN_C
media.h, 925

effectiveParams
KbAnim, 245
enable
AutoRun, 78
enableChanged
MPerfWidget, 611
enableMapper
MPerfWidget, 619
end
AnimScript, 35
endGroup
CkbSettings, 88
enter
RListWidget, 796
entryInfoList
QuaZipDir, 693, 694
entryInfoList64
QuaZipDir, 694, 695
entryList
QuaZipDir, 695, 696
evLoop
FwUpgradeDialog, 134
eventTimer
KbManager, 379
exec
FwUpgradeDialog, 130
exists
QuaZipDir, 696, 697
exitSuccess
FwUpgradeDialog, 135
externalAttr
QuaZipFileInfo, 729
QuaZipFileInfo64, 736
QuaZipNewInfo, 750
extra
QuaZipFileInfo, 729
QuaZipFileInfo64, 737
SettingsWidget, 811
extraGlobal
QuaZipNewInfo, 751
extraLocal
QuaZipNewInfo, 751
ExtraSettingsWidget, 116
~ExtraSettingsWidget, 120
ExtraSettingsWidget, 119
ExtraSettingsWidget, 119
on_animScanButton_clicked, 120
on_brightnessBox_clicked, 121
on_delayBox_clicked, 121
on_ditherBox_clicked, 121
on_fpsBox_valueChanged, 122
on_mAccelBox_clicked, 122
on_sAccelBox_clicked, 123
on_sSpeedBox_valueChanged, 123
on_trayBox_clicked, 123
pollUpdates, 124
ui, 125
extractDir
JlCompress, 177
extractFile
JlCompress, 179, 181
extractFiles
JlCompress, 182
extrasettingswidget.cpp
lastSharedDimming, 856

FR
KeyMap, 520
FALSE
ckb-anim.h, 836
FW
KbFirmware::FW, 126
features
Kb, 234
file_in_zip64_read_info_s, 989
fileName
KbFirmware::FW, 126
QuaZipFilePrivate, 741
fileNameCodec
QuaZipPrivate, 759
filePath
QuaZipDir, 698
fill_qiodevice64_filefunc
qioapi.cpp, 951
fill_qiodevice_filefunc
qioapi.cpp, 952
fill_zlib_filefunc64_32_def_from_filefunc32
ioapi.h, 945
qioapi.cpp, 953
fillRect
GradientDialogWidget, 158
filter
QuaZipDir, 698
find
Kb, 199
KbProfile, 441
findAnim
KbLight, 345
findAnimIdx
KbLight, 345
firmware
Kb, 234
firstFrame
AnimScript, 55
flags
QuaZipFileInfo, 729
QuaZipFileInfo64, 737
flush
QuaGzipFile, 644
QuaZIODevice, 650
fnKeys
RebindWidget, 791
fps
KbManager, 379
frame
AnimScript, 36
frameDisplayed

KbLight, 346
frameRate
Kb, 200
frameUpdate
Kb, 201
KbLight, 346
free_datablock
zip.c, 1069
free_linkedlist
zip.c, 1069
friendlyActionName
KbBind, 290
friendlyName
KbBind, 290
Key, 485
KeyAction, 497
KeyMap, 523
fromName
KeyMap, 524
fromStorage
KeyMap, 525
fromString
GradientButton, 138
functions
kbanim.cpp, 869
fwTable
KbFirmware, 331
fwTableHash
KbFirmware, 331
fwUpdPath
Kb, 234
fwUpdate
Kb, 202
fwUpdateFinished
FwUpgradeDialog, 131
Kb, 203
fwUpdateProgress
FwUpgradeDialog, 132
Kb, 203
FwUpgradeDialog, 126
~FwUpgradeDialog, 129
blob, 134
cleanBlob, 129
closeEvent, 130
evLoop, 134
exec, 130
exitSuccess, 135
fwUpdateFinished, 131
fwUpdateProgress, 132
FwUpgradeDialog, 129
FwUpgradeDialog, 129
kb, 135
on_actionButton_clicked, 132
on_cancelButton_clicked, 133
removeDev, 133
saveBlob, 134
savePath, 135
ui, 135
fwVersion
KbFirmware::FW, 126
fwupgradedialog.cpp
DIALOG_HEIGHT_MAX, 858
DIALOG_HEIGHT_MIN, 859
DIALOG_WIDTH, 859
ids, 859
verifyFw, 859

GB
KeyMap, 520
GB_DVORAK
KeyMap, 520
GUID
KbProfileDialog, 462
KbWidget, 483
get
CkbSettings, 88
getActualFileName
QuaZipFile, 712
getCaseSensitivity
QuaZipFile, 713
getComment
QuaZip, 661
getCommentCodec
QuaZip, 661
getCurrentFileInfo
QuaZip, 662, 663
getCurrentFileName
QuaZip, 664
getDefaultFileNameCodec
QuaZipPrivate, 756
getEntriesCount
QuaZip, 665
getFileInfo
QuaZipFile, 713, 714
getFileInfoList
QuaZip, 666
QuaZipPrivate, 756
getFileInfoList64
QuaZip, 666
getFileList
JlCompress, 183
getFileName
QuaGzipFile, 644
QuaZipFile, 715
getFileNameCodec
QuaZip, 667
getFileNameList
QuaZip, 667
getGradient
GradientDialog, 148
getIndicator
KbPerf, 417
getIoDevice
QuaZIODevice, 651
QuaZip, 668
getKeyMap
Kb, 203

getLayout
KeyMap, 526
getMacroNumber
Kb, 204
KbBind, 291
getMacroPath
Kb, 204
KbBind, 292
getMap
keymap.cpp, 899
getMode
QuaZip, 668
getModel
KeyMap, 527, 528
getMuteState
media.h, 926
media_linux.cpp, 927
media_mac.m, 928
getNTFSTime
quazipfileinfo.cpp, 979
getNTFSaTime
QuaZipFileInfo64, 732
getNTFScTime
QuaZipFileInfo64, 733
getNTFSmTime
QuaZipFileInfo64, 733
getPermissions
QuaZipFileInfo, 728
QuaZipFileInfo64, 734
getUnzFile
QuaZip, 669
getZip
QuaZipFile, 715
getZipError
QuaZip, 670
QuaZipFile, 715
getZipFile
QuaZip, 671
getZipName
QuaZip, 671
QuaZipFile, 716
globalCache
ckbsettings.cpp, 848
globalRemap
KbBind, 292
globalRemapTime
KbBind, 312
globalSettings
ckbsettings.cpp, 847
globalThread
ckbsettings.cpp, 848
goToFirstFile
QuaZip, 672
goToFirstUnmappedFile
QuaZipPrivate, 757
goToNextFile
QuaZip, 673
gradient
GradientDialog::Preset, 621
GradientButton, 135
_alpha, 141
_stops, 141
fromString, 138
GradientButton, 137
gradientChanged, 139
GradientButton, 137
pickGradient, 139
toString, 140
updateImage, 140
gradientChanged
GradientButton, 139
GradientDialog, 141
~GradientDialog, 145
addPreset, 145
colorChanged, 146
currentChanged, 147
currentPreset, 153
getGradient, 148
GradientDialog, 144
GradientDialog, 144
makeIcon, 148
on_presetDelete_clicked, 149
on_presetList_currentItemChanged, 149
on_presetName_textEdited, 150
on_presetSave_clicked, 150
on_stopOpacity_valueChanged, 151
on_stopPos_valueChanged, 151
presets, 153
setPreset, 152
ui, 154
updatePresets, 152
GradientDialog::Preset, 620
builtIn, 621
gradient, 621
name, 621
Preset, 621
GradientDialogWidget, 154
_colors, 168
_current, 168
_stops, 168
colorAt, 157
current, 158
currentChanged, 158
fillRect, 158
GradientDialogWidget, 157
GradientDialogWidget, 157
makeStops, 159
mouseMoveEvent, 160
mousePressEvent, 161
mouseReleaseEvent, 161
moveCurrent, 162
paintEvent, 163
selected, 168
selectedOffset, 168
selectedPos, 168
selectedStop, 164

selectionColors, 164
setCurrentColor, 165
setStops, 166
stopCount, 166
stopRect, 166
stops, 167
gradientdialog.cpp
prefsPath, 863
groups
CkbSettings, 95
guid
AnimScript, 37
KbAnim, 246
UsbId, 818
guidString
UsbId, 815

HIGH
KbPerf, 403
HW_IMAX
KbPerf, 404
hLine
AnimSettingDialog, 66
HW_I_COUNT
KbPerf, 434
KPerfWidget, 572
HWMODE_MAX
Kb, 234
handleNotificationChannel
KbBind, 293
hasCurrentFile
QuaZip, 674
hasCurrentFile_f
QuaZipPrivate, 759
hasDownloaded
KbFirmware, 328
hasFrame
AnimScript, 37
hasGPG
KbFirmware, 331
hasKeypress
AnimScript, 38
hasLed
Key, 486
hasParam
AnimScript, 38
hasParameter
KbAnim, 246
hasRepeat
AnimSettingDialog, 76
hasScan
Key, 486
hasShownNewFW
KbWidget, 483
hash
KbFirmware::FW, 126
height
KbPerf, 403
Key, 486
KeyMap, 528
helpStatus
RebindWidget, 768
hwIType
KbPerf, 434
hwLoading
Kb, 234
HwMode
KPerfWidget, 566
hwModeCount
Kb, 234
hwModified
UsbId, 818
hwModifiedString
UsbId, 816
hwProfile
Kb, 205
hwSave
Kb, 206

IT
KeyMap, 520
I_COUNT
KbPerf, 434
KPerfWidget, 572
i_hw
KbPerf, 403
KPerfWidget, 565
iColor
KbPerf, 434
iEnable
KbPerf, 434
iOpacity
KbPerf, 417, 418
iState
Kb, 234
iconClicked
MainWindow, 591
id
KbMode, 391
KbProfile, 442
ids
fwupgradedialog.cpp, 859
inFrame
AnimScript, 55
indexOf
Kb, 207
KbProfile, 443
indicator
KbPerf, 404
KPerfWidget, 566
indicators
KPerfWidget, 573
infoUpdated
Kb, 208
init
AnimScript, 39
ColorMap, 114
KbManager, 380

init_keys
crypt.h, 937
init_linkedlist
zip.c, 1070
initManager
KbFirmware, 329
initialized
AnimScript, 55
insert
KbProfile, 443
instance
KbFirmware, 331
internal
QuaZipFilePrivate, 741
internalAttr
QuaZipFileInfo, 730
QuaZipFileInfo64, 737
QuaZipNewInfo, 751
ioDevice
QuaZipPrivate, 759
ioapi.h
_FILE_OFFSET_BIT, 942
_LARGEFILE64_SOURCE, 943
_Z_OF, 945
__USE_LARGEFILE64, 942
fill_zlib_filefunc64_32_def_from_filefunc32, 945
OF, 943
ZCALLBACK, 943
ZCLOSE64, 943
ZERROR64, 943
ZFAKECLOSE64, 943
ZOPEN64, 944
ZPOS64_T, 945
ZREAD64, 944
ZSEEK64, 944
ZTELL64, 945
ZWRITE64, 945
zlib_filefunc64_32_def, 945
zlib_filefunc64_def, 945
zlib_filefunc_def, 945
isActive
KbAnim, 247
isAnim
KeyAction, 498
isAutoClose
QuaZip, 674
isBusy
CkbSettings, 89
isDPI
KeyAction, 499
isDataDescriptorWritingEnabled
QuaZip, 674
isEnabled
AutoRun, 79
isEncrypted
QuaZipFileInfo64, 735
isISO
KbBind, 294
KeyMap, 528, 529
isKeyboard
Kb, 208
KbBind, 294
KeyMap, 529, 530
isMacro
KeyAction, 499
isMedia
KeyAction, 499
isMouse
Kb, 209
KbBind, 295
KeyAction, 500
KeyMap, 531
isNormal
KeyAction, 500
isOpen
Kb, 209
QuaZip, 675
isProgram
KeyAction, 501
isRaw
QuaZipFile, 716
isRoot
QuaZipDir, 699
isRunning
KbAnim, 247
main.cpp, 918
isSequential
QuaGzipFile, 644
QuaZIODevice, 651
QuaZipFile, 717
isSetting
MPerfWidget, 619
isSpecial
KeyAction, 501
isStarted
KbLight, 348
isUnbound
KeyAction, 502
isValidMacro
KeyAction, 502
isZip64Enabled
QuaZip, 675

JlCompress, 169
compressDir, 170
compressFile, 171, 172
compressFiles, 173
compressSubDir, 174
extractDir, 177
extractFile, 179, 181
extractFiles, 182
getFileList, 183
removeFile, 184
JlCompress.cpp
copyData, 947

K65

KeyMap, 520
K65_HEIGHT
keymap.cpp, 896
K65_TOP_COUNT
keymap.cpp, 896
K65_WIDTH
keymap.cpp, 896
K65TopRow
keymap.cpp, 902
K70
KeyMap, 520
K70_HEIGHT
keymap.cpp, 896
K70_WIDTH
keymap.cpp, 896
K70_X_START
keymap.cpp, 896
K95
KeyMap, 520
K95_HEIGHT
keymap.cpp, 896
K95_WIDTH
keymap.cpp, 896
K95BaseMap
keymap.cpp, 902
K95Keys
keymap.cpp, 902
k95Widgets
KPerfWidget, 573
KPerfWidget
ALWAYS_OFF, 566
ALWAYS_ON, 566
BOTH, 566
NORMAL, 566
RGB, 566
KPerfWidget::IndicatorUi, 564, 912
KEY_SIZE
keywidget.cpp, 909
KEYCOUNT_K95
keymap.cpp, 896
KEYCOUNT_M65
keymap.cpp, 897
KEYCOUNT_SABRE
keymap.cpp, 897
KEYCOUNT_SCIM
keymap.cpp, 897
KP_NAME
AnimScript, 55
KP_NONE
AnimScript, 55
KP_POSITION
AnimScript, 55
KPerfWidget, 562
~KPerfWidget, 567
HW_I_COUNT, 572
HwMode, 566
I_COUNT, 572
i_hw, 565
indicator, 566
indicators, 573
k95Widgets, 573
KPerfWidget, 566
KPerfWidget, 566
mode2Raw, 568
on_copyButton_clicked, 568
on_intensityBox_valueChanged, 569
perf, 573
profile, 573
raw2Mode, 569
setPerf, 570
ui, 573
uiUpdated, 571
updateMapper, 573
KSTRAFE_HEIGHT
keymap.cpp, 897
KSTRAFE_WIDTH
keymap.cpp, 897
KSTRAFE_X_START
keymap.cpp, 897
KStrafeKeys
keymap.cpp, 903
Kb, 186
~Kb, 191
_currentMode, 232
_currentProfile, 232
_delay, 232
_dither, 232
_frameRate, 232
_hwProfile, 232
_layout, 233
_model, 233
_mouseAccel, 233
_needsSave, 233
_profiles, 233
_scrollSpeed, 233
appendProfile, 194
autoSave, 194
cmd, 233
cmdpath, 233
currentBind, 194
currentLight, 195
currentMode, 196
currentPerf, 196
currentProfile, 197
deleteHw, 197
deletePrevious, 198
devpath, 233
dither, 198, 199
features, 234
find, 199
firmware, 234
frameRate, 200
frameUpdate, 201
fwUpdPath, 234
fwUpdate, 202
fwUpdateFinished, 203

fwUpdateProgress, 203
getKeyMap, 203
getMacroNumber, 204
getMacroPath, 204
HWMODE_MAX, 234
hwLoading, 234
hwModeCount, 234
hwProfile, 205
hwSave, 206
iState, 234
indexOf, 207
infoUpdated, 208
isKeyboard, 208
isMouse, 209
isOpen, 209
Kb, 191
KbManager, 232
lastAutoSave, 234
layout, 210
load, 211
macroDelay, 212
macroNumber, 234
macroPath, 235
matches, 212
modeChanged, 213
modeRenamed, 213
model, 213
monochrome, 235
mouseAccel, 213, 214
needsSave, 214
newMode, 215, 216
newProfile, 216, 217
notifyNumber, 235
notifyPath, 235
pollrate, 235
prefsPath, 235
prevMode, 235
prevProfile, 235
profileAdded, 217
profileChanged, 218
profileRenamed, 218
profiles, 218, 219
readNotify, 219
run, 224
save, 225
scrollSpeed, 226
setCurrentMode, 227, 228
setCurrentProfile, 229
updateLayout, 230
usbModel, 235
usbSerial, 236
writeProfileHeader, 231
kb
FwUpgradeDialog, 135
kb.cpp
activeDevices, 866
notifyPathMutex, 866
notifyPaths, 866
KbAnim
Add, 239
Divide, 240
Multiply, 240
Normal, 239
Subtract, 239
KbFirmware
NO, 323
UNKNOWN, 323
YES, 323
KbId, 858
KbPerf
CAPS, 404
HIGH, 403
HW_IMAX, 404
LIGHT, 404
LOCK, 404
LOW, 403
LOWMED, 403
MACRO, 404
MEDHIGH, 403
MEDIUM, 403
MODE, 404
MUTE, 404
NONE, 404
NORMAL, 404
NUM, 404
OFF, 404
ON, 404
SCROLL, 404
KbAnim, 236
_guid, 262
_isActive, 262
_isActiveKp, 262
_keys, 262
_map, 262
_mode, 262
_name, 262
_needsSave, 262
_opacity, 263
_parameters, 263
_script, 263
_scriptGuid, 263
_scriptName, 263
_tempParameters, 263
blend, 242
catchUp, 243
commitParams, 245
effectiveParams, 245
guid, 246
hasParameter, 246
isActive, 247
isRunning, 247
KbAnim, 240–242
KbAnim, 240–242
keypress, 248
keys, 249, 250
kpRepeatMsec, 263

kpRepeatTime, 263
kpStopTime, 263
map, 250, 251
Mode, 239
mode, 251, 252
name, 252
needsSave, 253
newId, 253
opacity, 253, 254
parameter, 254, 255
reInit, 255
repeatKey, 264
repeatMsec, 264
repeatTime, 264
resetParams, 256
save, 257
script, 257
scriptName, 258
stop, 259
stopTime, 264
trigger, 260
updateParams, 261
KbAnimWidget, 264
~KbAnimWidget, 268
addAnim, 268
animChanged, 269
animations, 281
clearSelection, 269
current, 281
didUpdateSelection, 270
duplicateAnim, 270
KbAnimWidget, 267
KbAnimWidget, 267
light, 282
noReorder, 282
on_animList_currentItemChanged, 271
on_animList_customContextMenuRequested, 272
on_animList_itemChanged, 273
on_blendBox_activated, 273
on_deleteButton_clicked, 274
on_keyButton_clicked, 274
on_nameBox_textEdited, 275
on_opacityBox_valueChanged, 276
on_propertyButton_clicked, 276
refreshList, 277
reorderAnims, 278
selectedKeys, 282
setCurrent, 279
setLight, 280
setSelectedKeys, 281
ui, 282
KbBind, 282
_bind, 311
_devParent, 311
_globalRemap, 311
_map, 311
_needsSave, 311
_needsUpdate, 312
_winLock, 312
action, 287
bindAction, 288
defaultAction, 288
devParent, 289
didLoad, 290
friendlyActionName, 290
friendlyName, 290
getMacroNumber, 291
getMacroPath, 292
globalRemap, 292
globalRemapTime, 312
handleNotificationChannel, 293
isISO, 294
isKeyboard, 294
isMouse, 295
KbBind, 286
KbBind, 286
KeyAction, 311
keyEvent, 295
lastCmd, 312
lastGlobalRemapTime, 312
layoutChanged, 296
light, 297
load, 297
loadGlobalRemap, 298
map, 299, 300
modeParent, 300
needsSave, 301
noAction, 301, 302
perf, 302
resetAction, 303, 304
save, 304
saveGlobalRemap, 305
setAction, 306, 307
setGlobalRemap, 307
setNeedsUpdate, 308
update, 308
updated, 310
winLock, 310, 311
KbBindWidget, 312
~KbBindWidget, 315
bind, 320
currentSelection, 320
KbBindWidget, 315
KbBindWidget, 315
newLayout, 315
newSelection, 316
on_copyButton_clicked, 316
on_resetButton_clicked, 317
profile, 320
setBind, 318
ui, 320
updateBind, 318
updateSelDisplay, 319
kbConnected
KbManager, 381
kbDisconnected

KbManager, 381
KbFirmware, 321
~KbFirmware, 323
_checkUpdates, 324
_fileForBoard, 324
_latestForBoard, 325
checkUpdates, 326
dataForBoard, 327
downloadFinished, 328
downloaded, 327
fwTable, 331
fwTableHash, 331
hasDownloaded, 328
hasGPG, 331
initManager, 329
instance, 331
KbFirmware, 323
KbFirmware, 323
lastCheck, 332
lastFinished, 332
networkManager, 332
processDownload, 329
tableDownload, 332
versionForBoard, 331
KbFirmware::FW, 125
ckbVersion, 126
FW, 126
fileName, 126
fwVersion, 126
hash, 126
url, 126
KbLight, 332
~KbLight, 337
_animList, 360
_animMap, 361
_colorMap, 361
_dimming, 361
_indicatorList, 361
_indicatorMap, 361
_map, 361
_needsMapRefresh, 361
_needsSave, 361
_previewAnim, 361
_qColorMap, 361
_start, 362
addAnim, 337
animKeypress, 338
AnimList, 336
animList, 338, 339
base, 339
close, 340
color, 341, 342
colorMap, 342
didLoad, 343
dimming, 343
duplicateAnim, 344
findAnim, 345
findAnimIdx, 345
frameDisplayed, 346
frameUpdate, 346
isStarted, 348
KbLight, 336
KbLight, 336
lastFrameSignal, 362
load, 349
MAX_DIM, 362
map, 350, 351
needsSave, 352
open, 352
previewAnim, 353
printRGB, 354
rebuildBaseMap, 355
resetIndicators, 356
restartAnimation, 356
save, 357
setIndicator, 358
shareDimming, 359
stopPreview, 360
updated, 360
KbLightWidget, 362
~KbLightWidget, 365
changeAnim, 366
changeAnimKeys, 366
changeColor, 367
currentSelection, 372
KbLightWidget, 365
KbLightWidget, 365
light, 373
newSelection, 367
on_animButton_clicked, 368
on_bgButton_clicked, 369
on_brightnessBox_activated, 369
on_showAnimBox_clicked, 369
setLight, 370
setMonochrome, 371
toggleSidelight, 371
ui, 373
updateLight, 372
KbManager, 373
_daemonVersion, 385
_devices, 385
_eventTimer, 385
_guiVersion, 385
_kbManager, 385
_scanTimer, 385
ckbDaemonVersion, 377
ckbDaemonVersionF, 377
ckbGuiVersion, 378
ckbGuiVersionF, 378
devices, 379
eventTimer, 379
fps, 379
init, 380
Kb, 232
kbConnected, 381
kbDisconnected, 381

KbManager, 376
kbManager, 381
KbManager, 376
parseVersionString, 381
scanKeyboards, 382
scanTimer, 384
stop, 384
versionUpdated, 384
kbManager
KbManager, 381
KbMode, 386
~KbMode, 390
_bind, 398
_id, 399
_light, 399
_name, 399
_needsSave, 399
_perf, 399
bind, 390
doUpdate, 390
id, 391
KbMode, 388, 389
KbMode, 388, 389
keyMap, 392
light, 392
name, 393, 394
needsSave, 394
newId, 395
perf, 395
save, 396
setNeedsSave, 397
setNeedsUpdate, 397
updated, 398
KbPerf, 399
_angleSnap, 432
_curDpi, 406
_dpiIndicator, 432
_iOpacity, 432
_liftHeight, 432
_needsSave, 432
_needsUpdate, 432
angleSnap, 406, 407
applyIndicators, 407
bind, 408
curDpi, 409, 410
curDpiIdx, 411
DPI_COUNT, 432
DPI_MAX, 433
DPI_MIN, 433
didLoad, 412
dpi, 412, 413
dpiClr, 433
dpiColor, 414
dpiCurIdx, 433
dpiCurX, 433
dpiCurY, 433
dpiDown, 414
dpiEnabled, 415
dpiIndicator, 416
dpiLastIdx, 433
dpiOn, 433
dpiUp, 416
dpiX, 433
dpiY, 434
getIndicator, 417
HW_I_COUNT, 434
height, 403
hwIType, 434
I_COUNT, 434
i_hw, 403
iColor, 434
iEnable, 434
iOpacity, 417, 418
indicator, 404
KbPerf, 404, 405
KbPerf, 404, 405
liftHeight, 418, 419
light, 419
light100Color, 434
lightIndicator, 420
load, 420
modeParent, 422
muteNAColor, 434
needsSave, 422
OTHER, 434
operator=, 423
popDpi, 423
pushDpi, 424, 425
pushSniper, 426
pushedDpis, 435
runningPushIdx, 435
SNIPER, 435
save, 426
setIndicator, 427
setNeedsUpdate, 428
settingsUpdated, 428
sniperDpi, 429, 430
update, 430
KbProfile, 435
_currentMode, 452
_id, 452
_keyMap, 452
_modes, 452
_name, 452
_needsSave, 452
append, 440
currentMode, 440, 441
find, 441
id, 442
indexOf, 443
insert, 443
KbProfile, 438, 439
KbProfile, 438, 439
keyMap, 444
modeCount, 445
ModeList, 438

modes, 445, 446
move, 446
name, 447
needsSave, 447
newId, 448
removeAll, 449
save, 450
setNeedsSave, 451
setNeedsUpdate, 451
KbProfileDialog, 453
~KbProfileDialog, 455
addNewProfileItem, 455
device, 461
GUID, 462
KbProfileDialog, 455
KbProfileDialog, 455
NEW_FLAG, 462
on_profileList_currentItemChanged, 456
on_profileList_customContextMenuRequested, 456
on_profileList_itemChanged, 458
on_profileList_itemClicked, 459
profileList_reordered, 459
repopulate, 460
ui, 462
KbWidget, 462
~KbWidget, 466
addNewModeItem, 467
currentMode, 483
devUpdate, 467
device, 483
GUID, 483
hasShownNewFW, 483
KbWidget, 465
KbWidget, 465
lastAutoSave, 483
modeChanged, 467
modeIcon, 469
modeUpdate, 471
modesList_reordered, 470
NEW_FLAG, 483
name, 471
on_fwUpdButton_clicked, 471
on_hwSaveButton_clicked, 473
on_modesList_currentItemChanged, 474
on_modesList_customContextMenuRequested, 475
on_modesList_itemChanged, 476
on_modesList_itemClicked, 477
on_profileBox_activated, 477
on_tabWidget_currentChanged, 478
profileChanged, 478
showFirstTab, 479
showFwUpdate, 480
showLastTab, 480
ui, 483
updateFwButton, 481
updateProfileList, 481
kbWidgets
MainWindow, 597
kbanim.cpp
blendAdd, 868
blendDivide, 868
blendFunc, 868
blendMultiply, 868
blendNormal, 868
blendSubtract, 869
functions, 869
kbfirmware.cpp
AUTO_CHECK_TIME, 875
tableName, 875
kblight.cpp
_shareDimming, 879
activeLights, 879
lToS, 877
monoRgb, 877
sToL, 878
kbmanager.cpp
devpath, 882
kbperf.cpp
map_last, 885
Key, 484
_friendlyName, 486
_storageName, 486
friendlyName, 485
hasLed, 486
hasScan, 486
height, 486
name, 486
operator bool, 485
storageName, 485
width, 487
x, 487
y, 487
key
KeyMap, 532
KeyAction
NORMAL, 491
SPECIAL, 491
UNBOUND, 491
KeyMap
_LAYOUT_MAX, 520
_MODEL_MAX, 521
DE, 520
DK, 520
ES, 520
EU, 520
EU_DVORAK, 520
FR, 520
GB, 520
GB_DVORAK, 520
IT, 520
K65, 520
K70, 520
K95, 520
M65, 521

MX, 520
NO, 520
NO_LAYOUT, 520
NO_MODEL, 520
PL, 520
SABRE, 521
SCIMITAR, 521
SE, 520
STRAFE, 521
US, 520
US_DVORAK, 520
KeyPatch, 894
KeyWidget
ADD, 545
NONE, 545
SET, 545
SUBTRACT, 545
TOGGLE, 545
KeyAction, 487
~KeyAction, 492
_value, 514
adjustDisplay, 492
animAction, 493
animInfo, 494
DPI_CUSTOM, 514
DPI_DOWN, 514
DPI_SNIPER, 514
DPI_UP, 514
defaultAction, 494
dpiAction, 495
dpiInfo, 496
driverName, 496
friendlyName, 497
isAnim, 498
isDPI, 499
isMacro, 499
isMedia, 499
isMouse, 500
isNormal, 500
isProgram, 501
isSpecial, 501
isUnbound, 502
isValidMacro, 502
KbBind, 311
KeyAction, 492
keyEvent, 503
keyRelease, 506
KeyAction, 492
LIGHT_DOWN, 514
LIGHT_DOWN_WRAP, 514
LIGHT_UP, 515
LIGHT_UP_WRAP, 515
LOCK_OFF, 515
LOCK_ON, 515
LOCK_TOGGLE, 515
lightAction, 506
lockAction, 507
MODE_NEXT, 515
MODE_NEXT_WRAP, 515
MODE_PREV, 515
MODE_PREV_WRAP, 515
macroAction, 507
macroContent, 508
macroDisplay, 508
macroFullLine, 509
macroLine, 510
modeAction, 510
noAction, 511
operator QString, 511
operator=, 511
PROGRAM_PR_INDEF, 516
PROGRAM_PR_KPSTOP, 516
PROGRAM_PR_KRSTOP, 516
PROGRAM_PR_MULTI, 516
PROGRAM_RE_INDEF, 516
PROGRAM_RE_KPSTOP, 516
PROGRAM_RE_MULTI, 516
preProgram, 516
programAction, 511
programInfo, 511
relProgram, 516
sniperValue, 517
specialInfo, 512
Type, 491
type, 513
value, 513
keyEvent
KbBind, 295
KeyAction, 503
keyHeight
KeyMap, 540
keyLayout
KeyMap, 540
KeyMap, 517
_keys, 540
byPosition, 522
contains, 522
count, 523
friendlyName, 523
fromName, 524
fromStorage, 525
getLayout, 526
getModel, 527, 528
height, 528
isISO, 528, 529
isKeyboard, 529, 530
isMouse, 531
key, 532
keyHeight, 540
keyLayout, 540
KeyMap, 521
keyModel, 540
keyWidth, 541
KeyMap, 521
keys, 532
Layout, 520

layout, 533
layoutNames, 533
locale, 534
map, 535
Model, 520
model, 535
modelHeight, 536
modelWidth, 536
name, 536
operator const QHash< QString, Key > &, 537
positions, 538
strLayout, 538
strModel, 539
toStorage, 539
width, 540
keyMap
KbMode, 392
KbProfile, 444
KeyWidget, 561
keyModel
KeyMap, 540
keyNames
ColorMap, 115
keyRelease
KeyAction, 506
KeyWidget, 541
_bindMap, 560
_colorMap, 560
_displayColorMap, 560
_indicators, 560
_monochrome, 560
_rgbMode, 560
animation, 560
BindMap, 544
bindMap, 545
clearAnimation, 545
clearSelection, 546
colorMap, 546
displayColorMap, 546
drawInfo, 546
keyMap, 561
KeyWidget, 545
KeyWidget, 545
map, 547
monochrome, 548
mouseCurrentX, 561
mouseCurrentY, 561
mouseDownMode, 561
mouseDownX, 561
mouseDownY, 561
mouseMoveEvent, 548
mousePressEvent, 550
mouseReleaseEvent, 551
newSelection, 561
paintEvent, 551
rgbMode, 557
selectAll, 557
selection, 561
selectionChanged, 558
setAnimation, 558
setAnimationToSelection, 559
setSelection, 559
sidelightToggled, 559
keyWidth
KeyMap, 541
keyaction.cpp
ANIM_KRSTOP, 891
ANIM_ONCE, 891
keymap.cpp
_patch, 898
ANSI_ENTER_H, 895
ANSI_ENTER_W, 895
ANSI_ENTER_X, 895
ANSI_ENTER_Y, 895
ANSI_LSHIFT_W, 895
ANSI_LSHIFT_X, 896
getMap, 899
K65_HEIGHT, 896
K65_TOP_COUNT, 896
K65_WIDTH, 896
K65TopRow, 902
K70_HEIGHT, 896
K70_WIDTH, 896
K70_X_START, 896
K95_HEIGHT, 896
K95_WIDTH, 896
K95BaseMap, 902
K95Keys, 902
KEYCOUNT_K95, 896
KEYCOUNT_M65, 897
KEYCOUNT_SABRE, 897
KEYCOUNT_SCIM, 897
KSTRAFE_HEIGHT, 897
KSTRAFE_WIDTH, 897
KSTRAFE_X_START, 897
KStrafeKeys, 903
M65_HEIGHT, 897
M65_WIDTH, 897
M65Keys, 904
N_LAYOUTS, 897
N_MODELS, 898
NS, 898
PATCH_COUNT, 898
patch, 898
patchANSI, 901
patchDE, 904
patchDK, 904
patchDvorak, 904
patchES, 904
patchEU, 905
patchFR, 905
patchISO, 902
patchIT, 905
patchMX, 905
patchNO, 905
patchSE, 906

SABRE_HEIGHT, 898
SABRE_WIDTH, 898
SCIM_HEIGHT, 898
SCIM_WIDTH, 898
SabreKeys, 906
ScimKeys, 906
standardMaps, 906
keypress
AnimScript, 40
KbAnim, 248
keys
KbAnim, 249, 250
KeyMap, 532
keywidget.cpp
KEY_SIZE, 909
m65Overlay, 909
monoRgb, 908
sabOverlay, 909
scimOverlay, 909
kpRepeatMsec
KbAnim, 263
kpRepeatTime
KbAnim, 263
kpStopCheck
AnimSettingDialog, 76
kpStopTime
KbAnim, 263

LIGHT
KbPerf, 404
LOCK
KbPerf, 404
LOW
KbPerf, 403
LOWMED
KbPerf, 403
LIGHT_DOWN
KeyAction, 514
LIGHT_DOWN_WRAP
KeyAction, 514
LIGHT_UP
KeyAction, 515
LIGHT_UP_WRAP
KeyAction, 515
LOCALHEADERMAGIC
zip.c, 1065
LOCK_OFF
KeyAction, 515
LOCK_ON
KeyAction, 515
LOCK_TOGGLE
KeyAction, 515
lToS
kblight.cpp, 877
lastAutoSave
Kb, 234
KbWidget, 483
lastCheck
KbFirmware, 332
lastCmd
KbBind, 312
lastDuration
AnimSettingDialog, 76
lastFinished
KbFirmware, 332
lastFrame
AnimScript, 56
lastFrameSignal
KbLight, 362
lastGlobalRemapTime
KbBind, 312
lastMappedDirectoryEntry
QuaZipPrivate, 759
lastSharedDimming
extrasettingswidget.cpp, 856
Layout
KeyMap, 520
layout
Kb, 210
KeyMap, 533
layoutChanged
KbBind, 296
LayoutDialog, 573
~LayoutDialog, 575
LayoutDialog, 575
LayoutDialog, 575
selected, 575
ui, 576
layoutNames
KeyMap, 533
license
AnimScript, 41
liftHeight
KbPerf, 418, 419
light
KbAnimWidget, 282
KbBind, 297
KbLightWidget, 373
KbMode, 392
KbPerf, 419
light100Color
KbPerf, 434
lightAction
KeyAction, 506
lightIndicator
KbPerf, 420
linkedlist_data_s, 1062
linkedlist_datablock_internal_s, 1061
linkedlist_data
zip.c, 1067
linkedlist_datablock_internal
zip.c, 1067
list
AnimScript, 41
load
AnimScript, 42
Kb, 211

KbBind, 297
KbLight, 349
KbPerf, 420
LoadCentralDirectoryRecord
zip.c, 1071
loadGlobalRemap
KbBind, 298
local
unzip.c, 992
zip.c, 1065
locale
KeyMap, 534
lockAction
KeyAction, 507
lockMutex
ckbsettings.cpp, 846
ckbsettingswriter.cpp, 850
lockMutexCache
ckbsettings.cpp, 846
ckbsettingswriter.cpp, 850
lockMutexStatic
ckbsettings.cpp, 846
ckbsettingswriter.cpp, 850
lockMutexStatic2
ckbsettings.cpp, 846
ckbsettingswriter.cpp, 850

M65
KeyMap, 521
M65_HEIGHT
keymap.cpp, 897
M65_WIDTH
keymap.cpp, 897
M65Keys
keymap.cpp, 904
m65Overlay
keywidget.cpp, 909
MACRO
KbPerf, 404
MAX_FILE_NAME_LENGTH
QuaZip, 657
MEDHIGH
KbPerf, 403
MEDIUM
KbPerf, 403
MODE
KbPerf, 404
MPerfWidget::DpiUi, 605, 934
MUTE
KbPerf, 404
MUTED
media.h, 925
MX
KeyMap, 520
M_PI
ckb-anim.h, 836
MAKEULONG64
zip.c, 1065
MAX_DIM
KbLight, 362
MODE_NEXT
KeyAction, 515
MODE_NEXT_WRAP
KeyAction, 515
MODE_PREV
KeyAction, 515
MODE_PREV_WRAP
KeyAction, 515
MPerfWidget, 602
~MPerfWidget, 608
_xyLink, 618
boxXChanged, 608
boxXMapper, 618
boxYChanged, 609
boxYMapper, 618
buttonMapper1, 618
buttonMapper2, 618
colorChanged, 610
colorClicked, 611
colorLink, 619
DPI_COUNT, 619
DPI_MAX, 619
DPI_MIN, 619
enableChanged, 611
enableMapper, 619
isSetting, 619
MPerfWidget, 606
MPerfWidget, 606
on_aSnapBox_clicked, 612
on_copyButton_clicked, 612
on_indicBox_clicked, 613
on_lHeightBox_activated, 613
on_spinBox_valueChanged, 614
on_xyBox_clicked, 614
perf, 619
profile, 619
setPerf, 615
sliderXMapper, 619
sliderXMoved, 616
sliderYMapper, 619
sliderYMoved, 617
stages, 620
ui, 620
macReader
RebindWidget, 792
macroAction
KeyAction, 507
macroBox
MacroReaderThread, 582
macroContent
KeyAction, 508
macroDelay
Kb, 212
macroDisplay
KeyAction, 508
macroFullLine
KeyAction, 509

macroLine
KeyAction, 510
macroNumber
Kb, 234
MacroReaderThread, 583
macroPath
Kb, 235
MacroReaderThread, 583
MacroReader, 576
~MacroReader, 578
MacroReader, 578
macroReaderThread, 579
MacroReader, 578
startWorkInAThread, 578
MacroReaderThread, 579
macroBox, 582
macroNumber, 583
macroPath, 583
MacroReaderThread, 581
macroText, 583
MacroReaderThread, 581
readMacro, 582
run, 582
macroReaderThread
MacroReader, 579
macroText
MacroReaderThread, 583
main
ckb-anim.h, 842
main.cpp, 919
main.cpp
CommandLineBackground, 918
CommandLineClose, 918
CommandLineError, 918
CommandLineHelpRequested, 918
CommandLineOK, 918
CommandLineVersionRequested, 918
main.cpp
appShare, 918
CommandLineParseResults, 918
isRunning, 918
main, 919
parseCommandLine, 921
pidActive, 922
SHMEM_SIZE, 917
SHMEM_SIZE_V015, 917
MainWindow, 583
~MainWindow, 587
addDevice, 588
checkFwUpdates, 589
cleanup, 589
closeAction, 597
closeEvent, 590
iconClicked, 591
kbWidgets, 597
MainWindow, 586
mainWindow, 597
MainWindow, 586
quitApp, 591
removeDevice, 592
restoreAction, 597
settingsWidget, 598
SettingsWidget, 811
showFwUpdateNotification, 592
showWindow, 593
stateChange, 594
timerTick, 595
toggleTrayIcon, 596
trayIcon, 598
trayIconMenu, 598
ui, 598
updateVersion, 596
mainWindow
MainWindow, 597
mainwindow.cpp
appShare, 923
configLabel, 923
makeIcon
GradientDialog, 148
makeStops
GradientDialogWidget, 159
map
KbAnim, 250, 251
KbBind, 299, 300
KbLight, 350, 351
KeyMap, 535
KeyWidget, 547
map_last
kbperf.cpp, 885
matches
Kb, 212
mdAdd
QuaZip, 657
mdAppend
QuaZip, 657
mdCreate
QuaZip, 657
mdNotOpen
QuaZip, 657
mdUnzip
QuaZip, 657
media.h
MUTED, 925
UNKNOWN, 925
UNMUTED, 925
media.h
ENUM_C, 925
ENUM_END_C, 925
EXTERN_C, 925
getMuteState, 926
muteState, 925
media_linux.cpp
getMuteState, 927
muteProcess, 927
media_mac.m
disableAppNap, 928

getMuteState, 928
mediaKeys
RebindWidget, 792
method
QuaZipFileInfo, 730
QuaZipFileInfo64, 737
minX
AnimScript, 56
minY
AnimScript, 56
modKeys
RebindWidget, 792
settingswidget.cpp, 1129
modNames
settingswidget.cpp, 1129
Mode
KbAnim, 239
QuaZip, 657
mode
KbAnim, 251, 252
QuaZipPrivate, 759
mode2Raw
KPerfWidget, 568
modeAction
KeyAction, 510
modeChanged
Kb, 213
KbWidget, 467
modeCount
KbProfile, 445
modeIcon
KbWidget, 469
ModeList
KbProfile, 438
modeParent
KbBind, 300
KbPerf, 422
modeRenamed
Kb, 213
ModeSelectDialog, 598
~ModeSelectDialog, 600
_modeList, 602
ModeSelectDialog, 600
ModeSelectDialog, 600
on_selAllButton_clicked, 601
on_selNoneButton_clicked, 601
selection, 601
ui, 602
modeUpdate
KbWidget, 471
Model
KeyMap, 520
model
Kb, 213
KeyMap, 535
modelHeight
KeyMap, 536
modelWidth
KeyMap, 536
modes
KbProfile, 445, 446
modesList_reordered
KbWidget, 470
modified
UsbId, 818
modifiedString
UsbId, 816, 817
monoRgb
kblight.cpp, 877
keywidget.cpp, 908
monochrome
Kb, 235
KeyWidget, 548
mouseAccel
Kb, 213, 214
mouseCurrentX
KeyWidget, 561
mouseCurrentY
KeyWidget, 561
mouseDownMode
KeyWidget, 561
mouseDownX
KeyWidget, 561
mouseDownY
KeyWidget, 561
mouseExtKeys
RebindWidget, 792
mouseKeys
RebindWidget, 792
mouseMoveEvent
GradientDialogWidget, 160
KeyWidget, 548
mousePressEvent
GradientDialogWidget, 161
KeyWidget, 550
mouseReleaseEvent
GradientDialogWidget, 161
KeyWidget, 551
move
KbProfile, 446
moveCurrent
GradientDialogWidget, 162
mperfwidget.cpp
DPI_BASE, 933
dpiExp, 932
dpiLog, 932
dpiRound, 933
SET_END, 932
SET_START, 932
xyLinkPath, 933
Multiply
KbAnim, 240
muteNAColor
KbPerf, 434
muteProcess
media_linux.cpp, 927

muteState
media.h, 925

NO
KbFirmware, 323
KeyMap, 520
NO_LAYOUT
KeyMap, 520
NO_MODEL
KeyMap, 520
NONE
KbPerf, 404
KeyWidget, 545
NORMAL
KPerfWidget, 566
KbPerf, 404
KeyAction, 491
NUM
KbPerf, 404
N_LAYOUTS
keymap.cpp, 897
N_MODELS
keymap.cpp, 898
NEW_FLAG
KbProfileDialog, 462
KbWidget, 483
NS
keymap.cpp, 898
name
AnimScript, 45
AnimSettingDialog, 67
GradientDialog::Preset, 621
KbAnim, 252
KbMode, 393, 394
KbProfile, 447
KbWidget, 471
Key, 486
KeyMap, 536
QuaZipFileInfo, 730
QuaZipFileInfo64, 737
QuaZipNewInfo, 751
nameFilters
QuaZipDir, 699
needsSave
Kb, 214
KbAnim, 253
KbBind, 301
KbLight, 352
KbMode, 394
KbPerf, 422
KbProfile, 447
networkManager
KbFirmware, 332
newDuration
AnimSettingDialog, 67
newGuid
UsbId, 817
newId
KbAnim, 253
KbMode, 395
KbProfile, 448
newLayout
KbBindWidget, 315
newMode
Kb, 215, 216
newModified
UsbId, 817
newProfile
Kb, 216, 217
newSelection
KbBindWidget, 316
KbLightWidget, 367
KeyWidget, 561
noAction
KbBind, 301, 302
KeyAction, 511
noReorder
KbAnimWidget, 282
Normal
KbAnim, 239
notifyNumber
Kb, 235
notifyPath
Kb, 235
notifyPathMutex
kb.cpp, 866
notifyPaths
kb.cpp, 866
numKeys
RebindWidget, 792

OFF
KbPerf, 404
ON
KbPerf, 404
OF
ioapi.h, 943
ONE_DAY
animscript.cpp, 821
OTHER
KbPerf, 434
on_aSnapBox_clicked
MPerfWidget, 612
on_actionButton_clicked
FwUpgradeDialog, 132
on_altBox_activated
SettingsWidget, 804
on_animBox_activated
AnimAddDialog, 22
on_animBox_currentIndexChanged
RebindWidget, 769
on_animButton_clicked
KbLightWidget, 368
RebindWidget, 770
on_animList_currentItemChanged
KbAnimWidget, 271
on_animList_customContextMenuRequested
KbAnimWidget, 272

on_animList_itemChanged
KbAnimWidget, 273
on_animScanButton_clicked
ExtraSettingsWidget, 120
on_applyButton_clicked
RebindWidget, 770
on_autoFWBox_clicked
SettingsWidget, 804
on_bgButton_clicked
KbLightWidget, 369
on_blendBox_activated
KbAnimWidget, 273
on_brightnessBox_activated
KbLightWidget, 369
on_brightnessBox_clicked
ExtraSettingsWidget, 121
on_btnClearMacro_clicked
RebindWidget, 770
on_btnStartMacro_clicked
RebindWidget, 771
on_btnStopMacro_clicked
RebindWidget, 771
on_cancelButton_clicked
FwUpgradeDialog, 133
RebindWidget, 772
on_capsBox_activated
SettingsWidget, 805
on_copyButton_clicked
KbBindWidget, 316
KPerfWidget, 568
MPerfWidget, 612
on_ctrlBox_activated
SettingsWidget, 805
on_delayBox_clicked
ExtraSettingsWidget, 121
on_delayBox_valueChanged
AnimSettingDialog, 68
on_deleteButton_clicked
KbAnimWidget, 274
on_ditherBox_clicked
ExtraSettingsWidget, 121
on_dpiBox_currentIndexChanged
RebindWidget, 773
on_dpiButton_clicked
RebindWidget, 774
on_extraButton_clicked
SettingsWidget, 805
on_fnBox_currentIndexChanged
RebindWidget, 774
on_fnButton_clicked
RebindWidget, 774
on_fpsBox_valueChanged
ExtraSettingsWidget, 122
on_fwUpdButton_clicked
KbWidget, 471
on_hwSaveButton_clicked
KbWidget, 473
on_indicBox_clicked
MPerfWidget, 613
on_intensityBox_valueChanged
KPerfWidget, 569
on_keyButton_clicked
KbAnimWidget, 274
on_kpDelayBox_valueChanged
AnimSettingDialog, 69
on_kpReleaseBox_stateChanged
AnimSettingDialog, 69
on_kpRepeatBox_valueChanged
AnimSettingDialog, 70
on_lHeightBox_activated
MPerfWidget, 613
on_layoutBox_activated
SettingsWidget, 806
on_lightBox_currentIndexChanged
RebindWidget, 774
on_lightButton_clicked
RebindWidget, 775
on_lockBox_currentIndexChanged
RebindWidget, 775
on_lockButton_clicked
RebindWidget, 775
on_loginItemBox_clicked
SettingsWidget, 806
on_mAccelBox_clicked
ExtraSettingsWidget, 122
on_mb2Box_currentIndexChanged
RebindWidget, 776
on_mb2Button_clicked
RebindWidget, 776
on_mbBox_currentIndexChanged
RebindWidget, 776
on_mbButton_clicked
RebindWidget, 777
on_mediaBox_currentIndexChanged
RebindWidget, 777
on_mediaButton_clicked
RebindWidget, 777
on_modBox_currentIndexChanged
RebindWidget, 777
on_modButton_clicked
RebindWidget, 778
on_modeBox_currentIndexChanged
RebindWidget, 778
on_modeButton_clicked
RebindWidget, 778
on_modesList_currentItemChanged
KbWidget, 474
on_modesList_customContextMenuRequested
KbWidget, 475
on_modesList_itemChanged
KbWidget, 476
on_modesList_itemClicked
KbWidget, 477
on_nameBox_textEdited
KbAnimWidget, 275
on_numBox_currentIndexChanged

RebindWidget, 779
on_numButton_clicked
RebindWidget, 779
on_opacityBox_valueChanged
KbAnimWidget, 276
on_presetBox_activated
AnimAddDialog, 23
on_presetDelete_clicked
GradientDialog, 149
on_presetList_currentItemChanged
GradientDialog, 149
on_presetName_textEdited
GradientDialog, 150
on_presetSave_clicked
GradientDialog, 150
on_previewBox_clicked
AnimAddDialog, 24
on_profileBox_activated
KbWidget, 477
on_profileList_currentItemChanged
KbProfileDialog, 456
on_profileList_customContextMenuRequested
KbProfileDialog, 456
on_profileList_itemChanged
KbProfileDialog, 458
on_profileList_itemClicked
KbProfileDialog, 459
on_programKpBox_textChanged
RebindWidget, 779
on_programKpButton_clicked
RebindWidget, 780
on_programKpSIBox_clicked
RebindWidget, 780
on_programKrBox_textChanged
RebindWidget, 780
on_programKrButton_clicked
RebindWidget, 781
on_programKrSIBox_clicked
RebindWidget, 781
on_propertyButton_clicked
KbAnimWidget, 276
on_pteMacroBox_textChanged
RebindWidget, 781
on_pushButton_clicked
SettingsWidget, 807
on_repeatBox_valueChanged
AnimSettingDialog, 70
on_resetButton_clicked
KbBindWidget, 317
RebindWidget, 782
on_sAccelBox_clicked
ExtraSettingsWidget, 123
on_sSpeedBox_valueChanged
ExtraSettingsWidget, 123
on_selAllButton_clicked
ModeSelectDialog, 601
on_selNoneButton_clicked
ModeSelectDialog, 601
on_shiftBox_activated
SettingsWidget, 807
on_showAnimBox_clicked
KbLightWidget, 369
on_spinBox_valueChanged
MPerfWidget, 614
on_stopOpacity_valueChanged
GradientDialog, 151
on_stopPos_valueChanged
GradientDialog, 151
on_tabWidget_currentChanged
KbWidget, 478
on_trayBox_clicked
ExtraSettingsWidget, 123
on_typingBox_currentIndexChanged
RebindWidget, 782
on_typingButton_clicked
RebindWidget, 782
on_unbindButton_clicked
RebindWidget, 783
on_wheelBox_currentIndexChanged
RebindWidget, 783
on_wheelButton_clicked
RebindWidget, 783
on_winBox_activated
SettingsWidget, 807
on_xyBox_clicked
MPerfWidget, 614
once
AutoRun, 80
opacity
KbAnim, 253, 254
open
KbLight, 352
QuaGzipFile, 644, 645
QuaZIODevice, 651
QuaZip, 676
QuaZipFile, 717–719
operator bool
Key, 485
operator const QHash< QString, Key > &
KeyMap, 537
operator QString
KeyAction, 511
operator=
ColorMap, 115
KbPerf, 423
KeyAction, 511
QuaGzipFile, 645
QuaZip, 679
QuaZipDir, 700
QuaZipFile, 721
operator==
QuaZipDir, 700
orderChanged
RListWidget, 797

p
QuaZip, 686

QuaZipFile, 727
PL
KeyMap, 520
PATCH_COUNT
keymap.cpp, 898
PROGRAM_PR_INDEF
KeyAction, 516
PROGRAM_PR_KPSTOP
KeyAction, 516
PROGRAM_PR_KRSTOP
KeyAction, 516
PROGRAM_PR_MULTI
KeyAction, 516
PROGRAM_RE_INDEF
KeyAction, 516
PROGRAM_RE_KPSTOP
KeyAction, 516
PROGRAM_RE_MULTI
KeyAction, 516
paintEvent
GradientDialogWidget, 163
KeyWidget, 551
param
AnimScript, 45
paramIterator
AnimScript, 47
parameter
KbAnim, 254, 255
parameters
AnimScript, 46
parseCommandLine
main.cpp, 921
parseVersionString
KbManager, 381
patch
keymap.cpp, 898
patchANSI
keymap.cpp, 901
patchDE
keymap.cpp, 904
patchDK
keymap.cpp, 904
patchDvorak
keymap.cpp, 904
patchES
keymap.cpp, 904
patchEU
keymap.cpp, 905
patchFR
keymap.cpp, 905
patchISO
keymap.cpp, 902
patchIT
keymap.cpp, 905
patchMX
keymap.cpp, 905
patchNO
keymap.cpp, 905
patchSE
keymap.cpp, 906
path
AnimScript, 47
QuaZipDir, 700
perf
KbBind, 302
KbMode, 395
KPerfWidget, 573
MPerfWidget, 619
permissionsFromExternalAttr
quazipfileinfo.cpp, 980
pickColor
ColorButton, 105
pickGradient
GradientButton, 139
pidActive
main.cpp, 922
pollUpdates
ExtraSettingsWidget, 124
SettingsWidget, 807
pollrate
Kb, 235
popDpi
KbPerf, 423
pos
QIODevice_descriptor, 624
QuaZipFile, 721
positions
KeyMap, 538
preProgram
KeyAction, 516
prefsPath
gradientdialog.cpp, 863
Kb, 235
Preset
GradientDialog::Preset, 621
preset
AnimScript, 48
PresetValue
AnimScript, 30
presets
AnimScript, 48
GradientDialog, 153
prevMode
Kb, 235
prevProfile
Kb, 235
previewAnim
KbLight, 353
previousItems
RListWidget, 799
printParams
AnimScript, 49
printRGB
KbLight, 354
printurl
ckb-anim.h, 844

process
AnimScript, 56
processDownload
KbFirmware, 329
profile
KbBindWidget, 320
KPerfWidget, 573
MPerfWidget, 619
RebindWidget, 792
profileAdded
Kb, 217
profileChanged
Kb, 218
KbWidget, 478
profileList_reordered
KbProfileDialog, 459
profileRenamed
Kb, 218
profiles
Kb, 218, 219
programAction
KeyAction, 511
programInfo
KeyAction, 511
pushDpi
KbPerf, 424, 425
pushSniper
KbPerf, 426
pushedDpis
KbPerf, 435
pwd
CkbSettings, 90, 91

q
QuaZipFilePrivate, 742
QuaZipPrivate, 760
QColorMap
colormap.h, 855
QColorMapIterator
colormap.h, 855
QDialog, 622
QIODevice, 623
QIODevice_descriptor, 624
pos, 624
QIODevice_descriptor, 624
QIODevice_descriptor, 624
QListWidget, 625
QMainWindow, 626
QMutableColorMapIterator
colormap.h, 855
QObject, 627
QPushButton, 628
QThread, 629
QUAZIO_INBUFSIZE
quaziodevice.cpp, 968
QUAZIO_OUTBUFSIZE
quaziodevice.cpp, 968
QUAZIP_EXPORT
quazip_global.h, 973
QWidget, 760
qioapi.cpp
call_zopen64, 950
call_zseek64, 950
call_ztell64, 950
fill_qiodevice64_filefunc, 951
fill_qiodevice_filefunc, 952
fill_zlib_filefunc64_32_def_from_filefunc32, 953
qiodevice64_seek_file_func, 954
qiodevice64_tell_file_func, 955
qiodevice_close_file_func, 955
qiodevice_error_file_func, 956
qiodevice_fakeclose_file_func, 956
qiodevice_open_file_func, 957
qiodevice_read_file_func, 958
qiodevice_seek_file_func, 958
qiodevice_tell_file_func, 959
qiodevice_write_file_func, 959
SEEK_CUR, 950
SEEK_END, 950
SEEK_SET, 950
qiodevice64_seek_file_func
qioapi.cpp, 954
qiodevice64_tell_file_func
qioapi.cpp, 955
qiodevice_close_file_func
qioapi.cpp, 955
qiodevice_error_file_func
qioapi.cpp, 956
qiodevice_fakeclose_file_func
qioapi.cpp, 956
qiodevice_open_file_func
qioapi.cpp, 957
qiodevice_read_file_func
qioapi.cpp, 958
qiodevice_seek_file_func
qioapi.cpp, 958
qiodevice_tell_file_func
qioapi.cpp, 959
qiodevice_write_file_func
qioapi.cpp, 959
qs_strcmp
colormap.cpp, 853
QuaZip
csDefault, 657
csInsensitive, 657
csSensitive, 657
MAX_FILE_NAME_LENGTH, 657
mdAdd, 657
mdAppend, 657
mdCreate, 657
mdNotOpen, 657
mdUnzip, 657
QuaAdler32, 630
calculate, 632
checksum, 633
QuaAdler32, 632
QuaAdler32, 632

reset, 632
update, 633
value, 633
QuaChecksum32, 633
calculate, 635
reset, 635
update, 635
value, 635
QuaCrc32, 636
calculate, 638
checksum, 639
QuaCrc32, 638
QuaCrc32, 638
reset, 638
update, 639
value, 639
QuaGzipFile, 639
~QuaGzipFile, 643
close, 643
d, 646
flush, 644
getFileName, 644
isSequential, 644
open, 644, 645
operator=, 645
QuaGzipFile, 642, 643
QuaGzipFile, 642, 643
readData, 645
setFileName, 645
writeData, 645
QuaZIODevice, 646
~QuaZIODevice, 649
close, 649
d, 653
flush, 650
getIoDevice, 651
isSequential, 651
open, 651
QuaZIODevice, 649
QuaZIODevice, 649
readData, 652
writeData, 653
QuaZip, 654
~QuaZip, 658
CaseSensitivity, 657
close, 659
Constants, 657
convertCaseSensitivity, 660
getComment, 661
getCommentCodec, 661
getCurrentFileInfo, 662, 663
getCurrentFileName, 664
getEntriesCount, 665
getFileInfoList, 666
getFileInfoList64, 666
getFileNameCodec, 667
getFileNameList, 667
getIoDevice, 668
getMode, 668
getUnzFile, 669
getZipError, 670
getZipFile, 671
getZipName, 671
goToFirstFile, 672
goToNextFile, 673
hasCurrentFile, 674
isAutoClose, 674
isDataDescriptorWritingEnabled, 674
isOpen, 675
isZip64Enabled, 675
Mode, 657
open, 676
operator=, 679
p, 686
QuaZip, 658
QuaZipPrivate, 686
QuaZip, 658
QuaZipPrivate, 758
setAutoClose, 679
setComment, 680
setCommentCodec, 680
setCurrentFile, 680
setDataDescriptorWritingEnabled, 682
setDefaultFileNameCodec, 683
setFileNameCodec, 684
setIoDevice, 684
setZip64Enabled, 685
setZipName, 685
QuaZip_getFileInfo
quazip.cpp, 969, 970
QuaZipDir, 686
~QuaZipDir, 689
caseSensitivity, 690
cd, 690
cdUp, 692
count, 692
d, 703
dirName, 693
entryInfoList, 693, 694
entryInfoList64, 694, 695
entryList, 695, 696
exists, 696, 697
filePath, 698
filter, 698
isRoot, 699
nameFilters, 699
operator=, 700
operator==, 700
path, 700
QuaZipDir, 689
QuaZipDir, 689
relativeFilePath, 701
setCaseSensitivity, 701
setFilter, 702
setNameFilters, 702
setPath, 702

setSorting, 703
sorting, 703
QuaZipDir_convertInfoList
quazipdir.cpp, 974
QuaZipDir_getFileInfo
quazipdir.cpp, 975
QuaZipFile, 704
~QuaZipFile, 709
atEnd, 709
bytesAvailable, 710
close, 710
csize, 711
getActualFileName, 712
getCaseSensitivity, 713
getFileInfo, 713, 714
getFileName, 715
getZip, 715
getZipError, 715
getZipName, 716
isRaw, 716
isSequential, 717
open, 717–719
operator=, 721
p, 727
pos, 721
QuaZipFile, 707, 708
QuaZipFilePrivate, 726
QuaZipFile, 707, 708
QuaZipFilePrivate, 741
readData, 722
setFileName, 723
setZip, 723
setZipName, 724
size, 724
usize, 725
writeData, 726
QuaZipFileInfo, 727
comment, 729
compressedSize, 729
crc, 729
dateTime, 729
diskNumberStart, 729
externalAttr, 729
extra, 729
flags, 729
getPermissions, 728
internalAttr, 730
method, 730
name, 730
uncompressedSize, 730
versionCreated, 730
versionNeeded, 730
QuaZipFileInfo64, 730
comment, 736
compressedSize, 736
crc, 736
dateTime, 736
diskNumberStart, 736
externalAttr, 736
extra, 737
flags, 737
getNTFSaTime, 732
getNTFScTime, 733
getNTFSmTime, 733
getPermissions, 734
internalAttr, 737
isEncrypted, 735
method, 737
name, 737
toQuaZipFileInfo, 735
uncompressedSize, 737
versionCreated, 737
versionNeeded, 737
QuaZipFilePrivate, 738
~QuaZipFilePrivate, 740
caseSensitivity, 741
crc, 741
fileName, 741
internal, 741
q, 742
QuaZipFile, 741
QuaZipFilePrivate, 739, 740
QuaZipFile, 726
QuaZipFilePrivate, 739, 740
raw, 742
resetZipError, 740
setZipError, 740
uncompressedSize, 742
writePos, 742
zip, 742
zipError, 742
QuaZipNewInfo, 743
comment, 750
dateTime, 750
externalAttr, 750
extraGlobal, 751
extraLocal, 751
internalAttr, 751
name, 751
QuaZipNewInfo, 744, 745
QuaZipNewInfo, 744, 745
setFileDateTime, 746
setFileNTFSTimes, 749
setFileNTFSaTime, 746
setFileNTFScTime, 747
setFileNTFSmTime, 748
setFilePermissions, 749
setPermissions, 750
uncompressedSize, 751
QuaZipNewInfo_init
quazipnewinfo.cpp, 982
QuaZipNewInfo_setPermissions
quazipnewinfo.cpp, 983
QuaZipPrivate, 751
addCurrentFileToDirectoryMap, 754
autoClose, 758

clearDirectoryMap, 755
comment, 758
commentCodec, 758
dataDescriptorWritingEnabled, 758
defaultFileNameCodec, 759
directoryCaseInsensitive, 759
directoryCaseSensitive, 759
fileNameCodec, 759
getDefaultFileNameCodec, 756
getFileInfoList, 756
goToFirstUnmappedFile, 757
hasCurrentFile_f, 759
ioDevice, 759
lastMappedDirectoryEntry, 759
mode, 759
q, 760
QuaZip, 758
QuaZipPrivate, 753, 754
QuaZip, 686
QuaZipPrivate, 753, 754
zip64, 760
zipError, 760
zipName, 760
quaziodevice.cpp
QUAZIO_INBUFSIZE, 968
QUAZIO_OUTBUFSIZE, 968
quazip.cpp
QuaZip_getFileInfo, 969, 970
quazip.h
UNZ_OPENERROR, 972
quazip_global.h
QUAZIP_EXPORT, 973
UNUSED, 973
quazipdir.cpp
QuaZipDir_convertInfoList, 974
QuaZipDir_getFileInfo, 975
quazipfileinfo.cpp
getNTFSTime, 979
permissionsFromExternalAttr, 980
quazipnewinfo.cpp
QuaZipNewInfo_init, 982
QuaZipNewInfo_setPermissions, 983
setNTFSTime, 983
quitApp
MainWindow, 591

RGB
KPerfWidget, 566
RListWidget, 794
change, 796
currentData, 799
DATA_ROLE, 799
dragged, 799
enter, 796
orderChanged, 797
previousItems, 799
RListWidget, 796
reorderTimer, 799
rescanItems, 797
RListWidget, 796
timerTick, 798
raw
QuaZipFilePrivate, 742
raw2Mode
KPerfWidget, 569
reInit
KbAnim, 255
readAnyFrame
AnimScript, 56
readData
QuaGzipFile, 645
QuaZIODevice, 652
QuaZipFile, 722
readFrame
AnimScript, 56
readMacro
MacroReaderThread, 582
readNotify
Kb, 219
readProcess
AnimScript, 49
RebindWidget, 761
~RebindWidget, 766
applyChanges, 766
bind, 791
convertMacroBox, 768
fnKeys, 791
helpStatus, 768
macReader, 792
mediaKeys, 792
modKeys, 792
mouseExtKeys, 792
mouseKeys, 792
numKeys, 792
on_animBox_currentIndexChanged, 769
on_animButton_clicked, 770
on_applyButton_clicked, 770
on_btnClearMacro_clicked, 770
on_btnStartMacro_clicked, 771
on_btnStopMacro_clicked, 771
on_cancelButton_clicked, 772
on_dpiBox_currentIndexChanged, 773
on_dpiButton_clicked, 774
on_fnBox_currentIndexChanged, 774
on_fnButton_clicked, 774
on_lightBox_currentIndexChanged, 774
on_lightButton_clicked, 775
on_lockBox_currentIndexChanged, 775
on_lockButton_clicked, 775
on_mb2Box_currentIndexChanged, 776
on_mb2Button_clicked, 776
on_mbBox_currentIndexChanged, 776
on_mbButton_clicked, 777
on_mediaBox_currentIndexChanged, 777
on_mediaButton_clicked, 777
on_modBox_currentIndexChanged, 777
on_modButton_clicked, 778

on_modeBox_currentIndexChanged, 778
on_modeButton_clicked, 778
on_numBox_currentIndexChanged, 779
on_numButton_clicked, 779
on_programKpBox_textChanged, 779
on_programKpButton_clicked, 780
on_programKpSIBox_clicked, 780
on_programKrBox_textChanged, 780
on_programKrButton_clicked, 781
on_programKrSIBox_clicked, 781
on_pteMacroBox_textChanged, 781
on_resetButton_clicked, 782
on_typingBox_currentIndexChanged, 782
on_typingButton_clicked, 782
on_unbindButton_clicked, 783
on_wheelBox_currentIndexChanged, 783
on_wheelButton_clicked, 783
profile, 792
RebindWidget, 765
RebindWidget, 765
selection, 792
setBind, 784
setBox, 785
setSelection, 787
TAB_ANIM, 792
TAB_KB, 793
TAB_MACRO, 793
TAB_MOUSE, 793
TAB_PROGRAM, 793
TAB_SPECIAL, 793
typingKeys, 793
ui, 793
wheelKeys, 794
rebindwidget.cpp
DPI_CUST_IDX, 1125
DPI_OFFSET, 1125
rebuildBaseMap
KbLight, 355
refreshList
KbAnimWidget, 277
relProgram
KeyAction, 516
relativeFilePath
QuaZipDir, 701
remove
CkbSettings, 92
removeAll
KbProfile, 449
removeCache
CkbSettings, 96
removeDev
FwUpgradeDialog, 133
removeDevice
MainWindow, 592
removeFile
JlCompress, 184
reorderAnims
KbAnimWidget, 278
reorderTimer
RListWidget, 799
repeatKey
KbAnim, 264
repeatMsec
AnimScript, 56
KbAnim, 264
repeatTime
KbAnim, 264
repopulate
KbProfileDialog, 460
rescanItems
RListWidget, 797
reset
QuaAdler32, 632
QuaChecksum32, 635
QuaCrc32, 638
resetAction
KbBind, 303, 304
resetIndicators
KbLight, 356
resetParams
KbAnim, 256
resetZipError
QuaZipFilePrivate, 740
restartAnimation
KbLight, 356
restoreAction
MainWindow, 597
retrigger
AnimScript, 50
rgbMode
KeyWidget, 557
right
settingswidget.cpp, 1129
run
CkbSettingsWriter, 98
Kb, 224
MacroReaderThread, 582
runningPushIdx
KbPerf, 435

SABRE
KeyMap, 521
SCIMITAR
KeyMap, 521
SCROLL
KbPerf, 404
SE
KeyMap, 520
SET
KeyWidget, 545
SPECIAL
KeyAction, 491
STRAFE
KeyMap, 521
SUBTRACT
KeyWidget, 545
SABRE_HEIGHT

keymap.cpp, 898
SABRE_WIDTH
keymap.cpp, 898
SCIM_HEIGHT
keymap.cpp, 898
SCIM_WIDTH
keymap.cpp, 898
SEEK_CUR
qioapi.cpp, 950
zip.c, 1066
SEEK_END
qioapi.cpp, 950
zip.c, 1066
SEEK_SET
qioapi.cpp, 950
zip.c, 1066
SET_END
mperfwidget.cpp, 932
SET_START
mperfwidget.cpp, 932
SGroup, 811
~SGroup, 813
_settings, 813
SGroup, 813
SGroup, 813
SHMEM_SIZE
main.cpp, 917
SHMEM_SIZE_V015
main.cpp, 917
SIZECENTRALDIRITEM
unzip.c, 992
SIZECENTRALHEADER
zip.c, 1066
SIZEDATA_INDATABLOCK
zip.c, 1066
SIZEZIPLOCALHEADER
unzip.c, 992
SNIPER
KbPerf, 435
sToL
kblight.cpp, 878
sabOverlay
keywidget.cpp, 909
SabreKeys
keymap.cpp, 906
save
Kb, 225
KbAnim, 257
KbBind, 304
KbLight, 357
KbMode, 396
KbPerf, 426
KbProfile, 450
saveBlob
FwUpgradeDialog, 134
saveGlobalRemap
KbBind, 305
savePath
FwUpgradeDialog, 135
scan
AnimScript, 51
scanKeyboards
KbManager, 382
scanTimer
KbManager, 384
ScimKeys
keymap.cpp, 906
scimOverlay
keywidget.cpp, 909
script
KbAnim, 257
scriptName
KbAnim, 258
scripts
AnimAddDialog, 24
AnimScript, 56
scrollSpeed
Kb, 226
selectAll
KeyWidget, 557
selected
GradientDialogWidget, 168
LayoutDialog, 575
selectedKeys
KbAnimWidget, 282
selectedOffset
GradientDialogWidget, 168
selectedPos
GradientDialogWidget, 168
selectedStop
GradientDialogWidget, 164
selection
KeyWidget, 561
ModeSelectDialog, 601
RebindWidget, 792
selectionChanged
KeyWidget, 558
selectionColors
GradientDialogWidget, 164
set
CkbSettings, 92
setAction
KbBind, 306, 307
setAnimation
KeyWidget, 558
setAnimationToSelection
KeyWidget, 559
setAutoClose
QuaZip, 679
setBind
KbBindWidget, 318
RebindWidget, 784
setBox
RebindWidget, 785
setCaseSensitivity
QuaZipDir, 701

setComment
QuaZip, 680
setCommentCodec
QuaZip, 680
setCurrent
KbAnimWidget, 279
setCurrentColor
GradientDialogWidget, 165
setCurrentFile
QuaZip, 680
setCurrentMode
Kb, 227, 228
setCurrentProfile
Kb, 229
setDataDescriptorWritingEnabled
QuaZip, 682
setDefaultFileNameCodec
QuaZip, 683
setDuration
AnimScript, 52
setFileDateTime
QuaZipNewInfo, 746
setFileNTFSTimes
QuaZipNewInfo, 749
setFileNTFSaTime
QuaZipNewInfo, 746
setFileNTFScTime
QuaZipNewInfo, 747
setFileNTFSmTime
QuaZipNewInfo, 748
setFileName
QuaGzipFile, 645
QuaZipFile, 723
setFileNameCodec
QuaZip, 684
setFilePermissions
QuaZipNewInfo, 749
setFilter
QuaZipDir, 702
setGlobalRemap
KbBind, 307
setIndicator
KbLight, 358
KbPerf, 427
setIoDevice
QuaZip, 684
setLabel
ColorButton, 105, 106
setLight
KbAnimWidget, 280
KbLightWidget, 370
setMonochrome
KbLightWidget, 371
setNTFSTime
quazipnewinfo.cpp, 983
setNameFilters
QuaZipDir, 702
setNeedsSave
KbMode, 397
KbProfile, 451
setNeedsUpdate
KbBind, 308
KbMode, 397
KbPerf, 428
KbProfile, 451
setPath
QuaZipDir, 702
setPerf
KPerfWidget, 570
MPerfWidget, 615
setPermissions
QuaZipNewInfo, 750
setPreset
GradientDialog, 152
setSelectedKeys
KbAnimWidget, 281
setSelection
KeyWidget, 559
RebindWidget, 787
setSorting
QuaZipDir, 703
setStatus
SettingsWidget, 808
setStops
GradientDialogWidget, 166
setValue
CkbSettings, 93
setVersion
SettingsWidget, 808
setZip
QuaZipFile, 723
setZip64Enabled
QuaZip, 685
setZipError
QuaZipFilePrivate, 740
setZipName
QuaZip, 685
QuaZipFile, 724
settingPath
autorun.cpp, 826
settingWidgets
AnimSettingDialog, 76
settingsCacheMutex
ckbsettings.cpp, 848
ckbsettingswriter.cpp, 851
settingsMutex
ckbsettings.cpp, 848
ckbsettingswriter.cpp, 851
settingsUpdated
KbPerf, 428
SettingsWidget, 799
~SettingsWidget, 804
extra, 811
MainWindow, 811
on_altBox_activated, 804
on_autoFWBox_clicked, 804

on_capsBox_activated, 805
on_ctrlBox_activated, 805
on_extraButton_clicked, 805
on_layoutBox_activated, 806
on_loginItemBox_clicked, 806
on_pushButton_clicked, 807
on_shiftBox_activated, 807
on_winBox_activated, 807
pollUpdates, 807
setStatus, 808
setVersion, 808
SettingsWidget, 802
SettingsWidget, 802
showLayoutDialog, 809
ui, 811
updateModifiers, 809
settingsWidget
MainWindow, 598
settingswidget.cpp
devpath, 1129
modKeys, 1129
modNames, 1129
right, 1129
shareDimming
KbLight, 359
showFirstTab
KbWidget, 479
showFwUpdate
KbWidget, 480
showFwUpdateNotification
MainWindow, 592
showLastTab
KbWidget, 480
showLayoutDialog
SettingsWidget, 809
showPreview
AnimAddDialog, 24
showWindow
MainWindow, 593
sidelightToggled
KeyWidget, 559
size
QuaZipFile, 724
sliderXMapper
MPerfWidget, 619
sliderXMoved
MPerfWidget, 616
sliderYMapper
MPerfWidget, 619
sliderYMoved
MPerfWidget, 617
sniperDpi
KbPerf, 429, 430
sniperValue
KeyAction, 517
sorting
QuaZipDir, 703
specialInfo
KeyAction, 512
src/ckb/animadddialog.cpp, 819
src/ckb/animadddialog.h, 819
src/ckb/animscript.cpp, 820
src/ckb/animscript.h, 822
src/ckb/animsettingdialog.cpp, 823
src/ckb/animsettingdialog.h, 825
src/ckb/autorun.cpp, 826
src/ckb/autorun.h, 827
src/ckb/ckb-anim.h, 827
src/ckb/ckbsettings.cpp, 845
src/ckb/ckbsettings.h, 848
src/ckb/ckbsettingswriter.cpp, 849
src/ckb/ckbsettingswriter.h, 851
src/ckb/colorbutton.cpp, 852
src/ckb/colorbutton.h, 852
src/ckb/colormap.cpp, 853
src/ckb/colormap.h, 854
src/ckb/extrasettingswidget.cpp, 855
src/ckb/extrasettingswidget.h, 856
src/ckb/fwupgradedialog.cpp, 857
src/ckb/fwupgradedialog.h, 860
src/ckb/gradientbutton.cpp, 861
src/ckb/gradientbutton.h, 861
src/ckb/gradientdialog.cpp, 862
src/ckb/gradientdialog.h, 863
src/ckb/gradientdialogwidget.cpp, 864
src/ckb/gradientdialogwidget.h, 864
src/ckb/kb.cpp, 865
src/ckb/kb.h, 866
src/ckb/kbanim.cpp, 867
src/ckb/kbanim.h, 869
src/ckb/kbanimwidget.cpp, 870
src/ckb/kbanimwidget.h, 870
src/ckb/kbbind.cpp, 871
src/ckb/kbbind.h, 872
src/ckb/kbbindwidget.cpp, 873
src/ckb/kbbindwidget.h, 873
src/ckb/kbfirmware.cpp, 874
src/ckb/kbfirmware.h, 875
src/ckb/kblight.cpp, 876
src/ckb/kblight.h, 879
src/ckb/kblightwidget.cpp, 880
src/ckb/kblightwidget.h, 880
src/ckb/kbmanager.cpp, 881
src/ckb/kbmanager.h, 882
src/ckb/kbmode.cpp, 883
src/ckb/kbmode.h, 884
src/ckb/kbperf.cpp, 885
src/ckb/kbperf.h, 885
src/ckb/kbprofile.cpp, 886
src/ckb/kbprofile.h, 887
src/ckb/kbprofiledialog.cpp, 887
src/ckb/kbprofiledialog.h, 888
src/ckb/kbwidget.cpp, 889
src/ckb/kbwidget.h, 890
src/ckb/keyaction.cpp, 891
src/ckb/keyaction.h, 892

src/ckb/keymap.cpp, 892
src/ckb/keymap.h, 907
src/ckb/keywidget.cpp, 907
src/ckb/keywidget.h, 909
src/ckb/kperfwidget.cpp, 910
src/ckb/kperfwidget.h, 911
src/ckb/layoutdialog.cpp, 913
src/ckb/layoutdialog.h, 914
src/ckb/macroreader.cpp, 915
src/ckb/macroreader.h, 915
src/ckb/main.cpp, 916
src/ckb/mainwindow.cpp, 923
src/ckb/mainwindow.h, 923
src/ckb/media.h, 925
src/ckb/media_linux.cpp, 926
src/ckb/media_mac.m, 928
src/ckb/modeselectdialog.cpp, 929
src/ckb/modeselectdialog.h, 930
src/ckb/mperfwidget.cpp, 931
src/ckb/mperfwidget.h, 933
src/ckb/quazip/JlCompress.cpp, 946
src/ckb/quazip/JlCompress.h, 948
src/ckb/quazip/crypt.h, 935
src/ckb/quazip/ioapi.h, 939
src/ckb/quazip/qioapi.cpp, 948
src/ckb/quazip/quaadler32.cpp, 960
src/ckb/quazip/quaadler32.h, 961
src/ckb/quazip/quachecksum32.h, 962
src/ckb/quazip/quacrc32.cpp, 963
src/ckb/quazip/quacrc32.h, 964
src/ckb/quazip/quagzipfile.cpp, 965
src/ckb/quazip/quagzipfile.h, 966
src/ckb/quazip/quaziodevice.cpp, 967
src/ckb/quazip/quaziodevice.h, 968
src/ckb/quazip/quazip.cpp, 969
src/ckb/quazip/quazip.h, 971
src/ckb/quazip/quazip_global.h, 972
src/ckb/quazip/quazipdir.cpp, 973
src/ckb/quazip/quazipdir.h, 975
src/ckb/quazip/quazipfile.cpp, 976
src/ckb/quazip/quazipfile.h, 977
src/ckb/quazip/quazipfileinfo.cpp, 978
src/ckb/quazip/quazipfileinfo.h, 980
src/ckb/quazip/quazipnewinfo.cpp, 981
src/ckb/quazip/quazipnewinfo.h, 985
src/ckb/quazip/unzip.c, 986
src/ckb/quazip/unzip.h, 1038
src/ckb/quazip/zip.c, 1058
src/ckb/quazip/zip.h, 1114
src/ckb/rebindwidget.cpp, 1125
src/ckb/rebindwidget.h, 1126
src/ckb/rlistwidget.cpp, 1126
src/ckb/rlistwidget.h, 1127
src/ckb/settingswidget.cpp, 1128
src/ckb/settingswidget.h, 1130
stages
    MPerfWidget, 620
standardMaps
    keymap.cpp, 906
startWorkInAThread
    MacroReader, 578
stateChange
    MainWindow, 594
stop
    AnimScript, 52
    KbAnim, 259
    KbManager, 384
stopCheck
    AnimSettingDialog, 76
stopCount
    GradientDialogWidget, 166
stopPreview
    KbLight, 360
stopRect
    GradientDialogWidget, 166
stopTime
    KbAnim, 264
stopped
    AnimScript, 56
stops
    GradientDialogWidget, 167
storageName
    Key, 485
strLayout
    KeyMap, 538
strModel
    KeyMap, 539
strcmpcasenosensitive_internal
    unzip.c, 993
Subtract
    KbAnim, 239

TOGGLE
    KeyWidget, 545
TAB_ANIM
    RebindWidget, 792
TAB_KB
    RebindWidget, 793
TAB_MACRO
    RebindWidget, 793
TAB_MOUSE
    RebindWidget, 793
TAB_PROGRAM
    RebindWidget, 793
TAB_SPECIAL
    RebindWidget, 793
TRUE
    ckb-anim.h, 837
TRYFREE
    unzip.c, 992
    zip.c, 1066
tableDownload
    KbFirmware, 332
tableName
    kbfirmware.cpp, 875
timerTick
    MainWindow, 595

RListWidget, 798
tm_unz_s, 1040
tm_zip_s, 1116
tm_unz
unzip.h, 1048
tm_zip
zip.h, 1120
toQuaZipFileInfo
QuaZipFileInfo64, 735
toStorage
KeyMap, 539
toString
GradientButton, 140
toggleSidelight
KbLightWidget, 371
toggleTrayIcon
MainWindow, 596
trayIcon
MainWindow, 598
trayIconMenu
MainWindow, 598
trigger
KbAnim, 260
Type
KeyAction, 491
type
KeyAction, 513
typingKeys
RebindWidget, 793

UNBOUND
KeyAction, 491
UNKNOWN
KbFirmware, 323
media.h, 925
UNMUTED
media.h, 925
US
KeyMap, 520
US_DVORAK
KeyMap, 520
UNUSED
quazip_global.h, 973
UNZ_AUTO_CLOSE
unzip.h, 1046
UNZ_BADZIPFILE
unzip.h, 1046
UNZ_BUFSIZE
unzip.c, 993
UNZ_CRCERROR
unzip.h, 1046
UNZ_DEFAULT_FLAGS
unzip.h, 1046
UNZ_EOF
unzip.h, 1047
UNZ_ERRNO
unzip.h, 1047
UNZ_INTERNALERROR
unzip.h, 1047
UNZ_MAXFILENAMEINZIP
unzip.c, 993
UNZ_OK
unzip.h, 1047
UNZ_OPENERROR
quazip.h, 972
UNZ_PARAMERROR
unzip.h, 1047
Ui, 17
ui
AnimAddDialog, 25
AnimSettingDialog, 76
ExtraSettingsWidget, 125
FwUpgradeDialog, 135
GradientDialog, 154
KbAnimWidget, 282
KbBindWidget, 320
KbLightWidget, 373
KbProfileDialog, 462
KbWidget, 483
KPerfWidget, 573
LayoutDialog, 576
MainWindow, 598
ModeSelectDialog, 602
MPerfWidget, 620
RebindWidget, 793
SettingsWidget, 811
uiUpdated
KPerfWidget, 571
uncompressedSize
QuaZipFileInfo, 730
QuaZipFileInfo64, 737
QuaZipFilePrivate, 742
QuaZipNewInfo, 751
unz64_file_pos_s, 1045
unz64_s, 991
unz64_file_pos
unzip.h, 1048
unz64local_CheckCurrentFileCoherencyHeader
unzip.c, 994
unz64local_DosDateToTmuDate
unzip.c, 995
unz64local_GetCurrentFileInfoInternal
unzip.c, 997
unz64local_SearchCentralDir
unzip.c, 1004
unz64local_SearchCentralDir64
unzip.c, 1005
unz64local_getByte
unzip.c, 996
unz64local_getLong
unzip.c, 1001
unz64local_getLong64
unzip.c, 1002
unz64local_getShort
unzip.c, 1003
unz_file_info64_internal_s, 989
unz_file_info64_s, 1042

unz_file_info_s, 1043
unz_file_pos_s, 1045
unz_global_info64_s, 1040
unz_global_info_s, 1041
unz_copyright
    unzip.c, 1037
unz_file_info
    unzip.h, 1048
unz_file_info64
    unzip.h, 1048
unz_file_info64_internal
    unzip.c, 993
unz_file_pos
    unzip.h, 1048
unz_global_info
    unzip.h, 1048
unz_global_info64
    unzip.h, 1048
unzClearFlags
    unzip.c, 1007
    unzip.h, 1049
unzClose
    unzip.c, 1007
unzCloseCurrentFile
    unzip.c, 1008
unzFile
    unzip.h, 1048
unzGetCurrentFileInfo
    unzip.c, 1010
unzGetCurrentFileInfo64
    unzip.c, 1011
unzGetCurrentFileZStreamPos64
    unzip.c, 1011
unzGetFilePos
    unzip.c, 1012
    unzip.h, 1049
unzGetFilePos64
    unzip.c, 1012
    unzip.h, 1050
unzGetGlobalComment
    unzip.c, 1013
unzGetGlobalInfo
    unzip.c, 1013
unzGetGlobalInfo64
    unzip.c, 1014
unzGetLocalExtrafield
    unzip.c, 1014
unzGetOffset
    unzip.c, 1015
    unzip.h, 1050
unzGetOffset64
    unzip.c, 1015
    unzip.h, 1051
unzGoToFilePos
    unzip.c, 1016
    unzip.h, 1051
unzGoToFilePos64
    unzip.c, 1016
    unzip.h, 1052
unzGoToFirstFile
    unzip.c, 1017
unzGoToNextFile
    unzip.c, 1018
unzLocateFile
    unzip.c, 1019
unzOpen
    unzip.c, 1020
unzOpen2
    unzip.c, 1021
unzOpen2_64
    unzip.c, 1022
unzOpen64
    unzip.c, 1023
unzOpenCurrentFile
    unzip.c, 1023
unzOpenCurrentFile2
    unzip.c, 1024
unzOpenCurrentFile3
    unzip.c, 1024
unzOpenCurrentFilePassword
    unzip.c, 1027
unzOpenInternal
    unzip.c, 1028
    unzip.h, 1053
unzReadCurrentFile
    unzip.c, 1031
unzSetFlags
    unzip.c, 1034
    unzip.h, 1056
unzSetOffset
    unzip.c, 1035
    unzip.h, 1057
unzSetOffset64
    unzip.c, 1035
    unzip.h, 1057
unzStringFileNameCompare
    unzip.c, 1036
unzeof
    unzip.c, 1009
unzip.c
    _Z_OF, 993
    ALLOC, 992
    BUFREADCOMMENT, 992
    local, 992
    SIZECENTRALDIRITEM, 992
    SIZEZIPLOCALHEADER, 992
    strcmpcasenosensitive_internal, 993
    TRYFREE, 992
    UNZ_BUFSIZE, 993
    UNZ_MAXFILENAMEINZIP, 993
    unz64local_CheckCurrentFileCoherencyHeader, 994
    unz64local_DosDateToTmuDate, 995
    unz64local_GetCurrentFileInfoInternal, 997
    unz64local_SearchCentralDir, 1004
    unz64local_SearchCentralDir64, 1005

unz64local_getByte, 996
unz64local_getLong, 1001
unz64local_getLong64, 1002
unz64local_getShort, 1003
unz_copyright, 1037
unz_file_info64_internal, 993
unzClearFlags, 1007
unzClose, 1007
unzCloseCurrentFile, 1008
unzGetCurrentFileInfo, 1010
unzGetCurrentFileInfo64, 1011
unzGetCurrentFileZStreamPos64, 1011
unzGetFilePos, 1012
unzGetFilePos64, 1012
unzGetGlobalComment, 1013
unzGetGlobalInfo, 1013
unzGetGlobalInfo64, 1014
unzGetLocalExtrafield, 1014
unzGetOffset, 1015
unzGetOffset64, 1015
unzGoToFilePos, 1016
unzGoToFilePos64, 1016
unzGoToFirstFile, 1017
unzGoToNextFile, 1018
unzLocateFile, 1019
unzOpen, 1020
unzOpen2, 1021
unzOpen2_64, 1022
unzOpen64, 1023
unzOpenCurrentFile, 1023
unzOpenCurrentFile2, 1024
unzOpenCurrentFile3, 1024
unzOpenCurrentFilePassword, 1027
unzOpenInternal, 1028
unzReadCurrentFile, 1031
unzSetFlags, 1034
unzSetOffset, 1035
unzSetOffset64, 1035
unzStringFileNameCompare, 1036
unzeof, 1009
unztell, 1036
unztell64, 1037
z_crc_t, 993
unzip.h
_Z_OF, 1048, 1049
tm_unz, 1048
UNZ_AUTO_CLOSE, 1046
UNZ_BADZIPFILE, 1046
UNZ_CRCERROR, 1046
UNZ_DEFAULT_FLAGS, 1046
UNZ_EOF, 1047
UNZ_ERRNO, 1047
UNZ_INTERNALERROR, 1047
UNZ_OK, 1047
UNZ_PARAMERROR, 1047
unz64_file_pos, 1048
unz_file_info, 1048
unz_file_info64, 1048
unz_file_pos, 1048
unz_global_info, 1048
unz_global_info64, 1048
unzClearFlags, 1049
unzFile, 1048
unzGetFilePos, 1049
unzGetFilePos64, 1050
unzGetOffset, 1050
unzGetOffset64, 1051
unzGoToFilePos, 1051
unzGoToFilePos64, 1052
unzOpenInternal, 1053
unzSetFlags, 1056
unzSetOffset, 1057
unzSetOffset64, 1057
Z_BZIP2ED, 1047
unztell
unzip.c, 1036
unztell64
unzip.c, 1037
update
KbBind, 308
KbPerf, 430
QuaAdler32, 633
QuaChecksum32, 635
QuaCrc32, 639
update_keys
crypt.h, 938
updateBind
KbBindWidget, 318
updateFwButton
KbWidget, 481
updateImage
ColorButton, 106
GradientButton, 140
updateLayout
Kb, 230
updateLight
KbLightWidget, 372
updateMapper
AnimSettingDialog, 76
KPerfWidget, 573
updateModifiers
SettingsWidget, 809
updateParam
AnimSettingDialog, 71
updateParams
KbAnim, 261
updatePresets
GradientDialog, 152
updateProfileList
KbWidget, 481
updateSelDisplay
KbBindWidget, 319
updateStops
AnimSettingDialog, 74
updateVersion
MainWindow, 596

updated
KbBind, 310
KbLight, 360
KbMode, 398
url
KbFirmware::FW, 126
urlParam
animscript.cpp, 821
urldecode
ckb-anim.h, 844
UsbId, 814
guid, 818
guidString, 815
hwModified, 818
hwModifiedString, 816
modified, 818
modifiedString, 816, 817
newGuid, 817
newModified, 817
UsbId, 815
UsbId, 815
usbModel
Kb, 235
usbSerial
Kb, 236
usize
QuaZipFile, 725

VERSIONMADEBY
zip.c, 1066
value
CkbSettings, 94
KeyAction, 513
QuaAdler32, 633
QuaChecksum32, 635
QuaCrc32, 639
verifyFw
fwupgradedialog.cpp, 859
version
AnimScript, 53
versionCreated
QuaZipFileInfo, 730
QuaZipFileInfo64, 737
versionForBoard
KbFirmware, 331
versionNeeded
QuaZipFileInfo, 730
QuaZipFileInfo64, 737
versionUpdated
KbManager, 384

wheelKeys
RebindWidget, 794
width
Key, 487
KeyMap, 540
winLock
KbBind, 310, 311
Write_EndOfCentralDirectoryRecord
zip.c, 1074
Write_GlobalComment
zip.c, 1075
Write_LocalFileHeader
zip.c, 1076
Write_Zip64EndOfCentralDirectoryLocator
zip.c, 1078
Write_Zip64EndOfCentralDirectoryRecord
zip.c, 1079
writeCache
CkbSettings, 96
writeData
QuaGzipFile, 645
QuaZIODevice, 653
QuaZipFile, 726
writePos
QuaZipFilePrivate, 742
writeProfileHeader
Kb, 231

x
Key, 487
xyLinkPath
mperfwidget.cpp, 933

y
Key, 487
YES
KbFirmware, 323
year
AnimScript, 53

Z_BUFSIZE
zip.c, 1066
Z_BZIP2ED
unzip.h, 1047
zip.h, 1119
Z_MAXFILENAMEINZIP
zip.c, 1066
z_crc_t
unzip.c, 993
zip.c, 1067
ZCALLBACK
ioapi.h, 943
ZCLOSE64
ioapi.h, 943
ZERROR64
ioapi.h, 943
ZFAKECLOSE64
ioapi.h, 943
ZIP64ENDHEADERMAGIC
zip.c, 1066
ZIP64ENDLOCHEADERMAGIC
zip.c, 1067
ZIP_AUTO_CLOSE
zip.h, 1119
ZIP_BADZIPFILE
zip.h, 1119
ZIP_DEFAULT_FLAGS

zip.h, 1119
ZIP_EOF
zip.h, 1119
ZIP_ERRNO
zip.h, 1119
ZIP_INTERNALERROR
zip.h, 1119
ZIP_OK
zip.h, 1119
ZIP_PARAMERROR
zip.h, 1120
ZIP_SEQUENTIAL
zip.h, 1120
ZOPEN64
ioapi.h, 944
ZPOS64_T
ioapi.h, 945
ZREAD64
ioapi.h, 944
ZSEEK64
ioapi.h, 944
ZTELL64
ioapi.h, 945
ZWRITE64
ioapi.h, 945
zdecode
crypt.h, 937
zencode
crypt.h, 937
zip
QuaZipFilePrivate, 742
zip.c
_Z_OF, 1067
ALLOC, 1064
add_data_in_datablock, 1067
allocate_new_datablock, 1068
BUFREADCOMMENT, 1065
CENTRALHEADERMAGIC, 1065
DESCRIPTORHEADERMAGIC, 1065
ENDHEADERMAGIC, 1065
free_datablock, 1069
free_linkedlist, 1069
init_linkedlist, 1070
LOCALHEADERMAGIC, 1065
linkedlist_data, 1067
linkedlist_datablock_internal, 1067
LoadCentralDirectoryRecord, 1071
local, 1065
MAKEULONG64, 1065
SEEK_CUR, 1066
SEEK_END, 1066
SEEK_SET, 1066
SIZECENTRALHEADER, 1066
SIZEDATA_INDATABLOCK, 1066
TRYFREE, 1066
VERSIONMADEBY, 1066
Write_EndOfCentralDirectoryRecord, 1074
Write_GlobalComment, 1075
Write_LocalFileHeader, 1076
Write_Zip64EndOfCentralDirectoryLocator, 1078
Write_Zip64EndOfCentralDirectoryRecord, 1079
Z_BUFSIZE, 1066
Z_MAXFILENAMEINZIP, 1066
z_crc_t, 1067
ZIP64ENDHEADERMAGIC, 1066
ZIP64ENDLOCHEADERMAGIC, 1067
zip64FlushWriteBuffer, 1080
zip64local_SearchCentralDir, 1086
zip64local_SearchCentralDir64, 1087
zip64local_TmzDateToDosDate, 1089
zip64local_getByte, 1081
zip64local_getLong, 1082
zip64local_getLong64, 1083
zip64local_getShort, 1084
zip64local_putValue, 1085
zip64local_putValue_inmemory, 1086
zip_copyright, 1114
zipClearFlags, 1090
zipClose, 1090
zipCloseFileInZip, 1092
zipCloseFileInZipRaw, 1093
zipCloseFileInZipRaw64, 1093
zipOpen, 1098
zipOpen2, 1098
zipOpen2_64, 1099
zipOpen3, 1100
zipOpen64, 1102
zipOpenNewFileInZip, 1103
zipOpenNewFileInZip2, 1103
zipOpenNewFileInZip2_64, 1104
zipOpenNewFileInZip3, 1104
zipOpenNewFileInZip3_64, 1105
zipOpenNewFileInZip4, 1105
zipOpenNewFileInZip4_64, 1106
zipOpenNewFileInZip64, 1110
zipRemoveExtraInfoBlock, 1110
zipSetFlags, 1111
zipWriteInFileInZip, 1112
zip.h
_Z_OF, 1120, 1121
DEF_MEM_LEVEL, 1119
tm_zip, 1120
Z_BZIP2ED, 1119
ZIP_AUTO_CLOSE, 1119
ZIP_BADZIPFILE, 1119
ZIP_DEFAULT_FLAGS, 1119
ZIP_EOF, 1119
ZIP_ERRNO, 1119
ZIP_INTERNALERROR, 1119
ZIP_OK, 1119
ZIP_PARAMERROR, 1120
ZIP_SEQUENTIAL, 1120
zipClearFlags, 1121
zipFile, 1120
zipOpen3, 1122
zipSetFlags, 1124

zipcharpc, 1120
zip64
QuaZipPrivate, 760
zip64_internal, 1064
zip64FlushWriteBuffer
zip.c, 1080
zip64local_SearchCentralDir
zip.c, 1086
zip64local_SearchCentralDir64
zip.c, 1087
zip64local_TmzDateToDosDate
zip.c, 1089
zip64local_getByte
zip.c, 1081
zip64local_getLong
zip.c, 1082
zip64local_getLong64
zip.c, 1083
zip64local_getShort
zip.c, 1084
zip64local_putValue
zip.c, 1085
zip64local_putValue_inmemory
zip.c, 1086
zip_fileinfo, 1117
zip_copyright
zip.c, 1114
zipClearFlags
zip.c, 1090
zip.h, 1121
zipClose
zip.c, 1090
zipCloseFileInZip
zip.c, 1092
zipCloseFileInZipRaw
zip.c, 1093
zipCloseFileInZipRaw64
zip.c, 1093
zipError
QuaZipFilePrivate, 742
QuaZipPrivate, 760
zipFile
zip.h, 1120
zipName
QuaZipPrivate, 760
zipOpen
zip.c, 1098
zipOpen2
zip.c, 1098
zipOpen2_64
zip.c, 1099
zipOpen3
zip.c, 1100
zip.h, 1122
zipOpen64
zip.c, 1102
zipOpenNewFileInZip
zip.c, 1103
zipOpenNewFileInZip2
zip.c, 1103
zipOpenNewFileInZip2_64
zip.c, 1104
zipOpenNewFileInZip3
zip.c, 1104
zipOpenNewFileInZip3_64
zip.c, 1105
zipOpenNewFileInZip4
zip.c, 1105
zipOpenNewFileInZip4_64
zip.c, 1106
zipOpenNewFileInZip64
zip.c, 1110
zipRemoveExtraInfoBlock
zip.c, 1110
zipSetFlags
zip.c, 1111
zip.h, 1124
zipWriteInFileInZip
zip.c, 1112
zipcharpc
zip.h, 1120
zlib_filefunc64_32_def_s, 942
zlib_filefunc64_def_s, 941
zlib_filefunc_def_s, 941
zlib_filefunc64_32_def
ioapi.h, 945
zlib_filefunc64_def
ioapi.h, 945
zlib_filefunc_def
ioapi.h, 945