12.20.2.30 isAlwaysOnTop() ........................................ 179
12.20.2.31 isAutoPull() ........................................... 180
12.20.2.32 isAutoPush() ........................................... 180
12.20.2.33 isAvoidCapitals() ..................................... 181
12.20.2.34 isAvoidNumbers() ...................................... 182
12.20.2.35 isHideContent() ......................................... 182
12.20.2.36 isHideOnClose() ......................................... 183
12.20.2.37 isHidePassword() ......................................... 184
12.20.2.38 isLessRandom() .......................................... 184
12.20.2.39 isMaximized() ........................................... 185
12.20.2.40 isStartMinimized() ...................................... 186
12.20.2.41 isTemplateAllFields() .................................. 186
12.20.2.42 isUseAutoclear() ....................................... 187
12.20.2.43 isUseAutoclearPanel() .................................. 188
12.20.2.44 isUseGit() ................................................ 188
12.20.2.45 isUseOtp() ................................................ 189
12.20.2.46 isUsePass() .............................................. 190
12.20.2.47 isUsePwgen() ............................................ 191
12.20.2.48 isUseQrencode() ........................................ 191
12.20.2.49 isUseSelection() ....................................... 192
12.20.2.50 isUseSymbols() ......................................... 192
12.20.2.51 isUseTemplate() ......................................... 193
12.20.2.52 isUseTrayIcon() ......................................... 194
12.20.2.53 isUseWebDav() .......................................... 195
12.20.2.54 setAddGPGId() .......................................... 195
12.20.2.55 setAlwaysOnTop() ...................................... 196
12.20.2.56 setAutoclearPanelSeconds() ............................ 196
12.20.2.57 setAutoclearSeconds() .................................. 197
12.20.2.58 setAutoPull() ............................................ 197
12.20.2.59 setAutoPush() ............................................ 198
12.20.2.60 setAvoidCapitals() ..................................... 198
12.20.2.61 setAvoidNumbers() ...................................... 198
12.20.2.62 setClipBoardType() ..................................... 199
12.20.2.63 setGeometry() ........................................... 199
12.20.2.64 setGitExecutable() ..................................... 200
12.20.2.65 setGpgExecutable() .................................... 200
12.20.2.66 setHideContent() ....................................... 201
12.20.2.67 setHideOnClose() ....................................... 201
12.20.2.68 setHidePassword() ...................................... 202
12.20.2.69 setLessRandom() ....................................... 202
12.20.2.70 setMaximized() .......................................... 203
12.20.2.71 setPassExecutable() .................................... 203
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>setPassStore()</td>
<td>204</td>
</tr>
<tr>
<td>setPassTemplate()</td>
<td>204</td>
</tr>
<tr>
<td>setPasswordChars()</td>
<td>205</td>
</tr>
<tr>
<td>setPasswordCharsSelection()</td>
<td>205</td>
</tr>
<tr>
<td>setPasswordConfiguration()</td>
<td>205</td>
</tr>
<tr>
<td>setPasswordLength()</td>
<td>206</td>
</tr>
<tr>
<td>setPos()</td>
<td>206</td>
</tr>
<tr>
<td>setProfile()</td>
<td>207</td>
</tr>
<tr>
<td>setProfiles()</td>
<td>207</td>
</tr>
<tr>
<td>setPwgenExecutable()</td>
<td>208</td>
</tr>
<tr>
<td>setQrencodeExecutable()</td>
<td>208</td>
</tr>
<tr>
<td>setSaveState()</td>
<td>209</td>
</tr>
<tr>
<td>setSize()</td>
<td>209</td>
</tr>
<tr>
<td>setStartMinimized()</td>
<td>210</td>
</tr>
<tr>
<td>setTemplateAllFields()</td>
<td>210</td>
</tr>
<tr>
<td>setUseAutoclear()</td>
<td>211</td>
</tr>
<tr>
<td>setUseAutoclearPanel()</td>
<td>211</td>
</tr>
<tr>
<td>setUseGit()</td>
<td>211</td>
</tr>
<tr>
<td>setUseOtp()</td>
<td>212</td>
</tr>
<tr>
<td>setUsePass()</td>
<td>212</td>
</tr>
<tr>
<td>setUsePwgen()</td>
<td>213</td>
</tr>
<tr>
<td>setUseQrencode()</td>
<td>213</td>
</tr>
<tr>
<td>setUseSelection()</td>
<td>214</td>
</tr>
<tr>
<td>setUseSymbols()</td>
<td>214</td>
</tr>
<tr>
<td>setUseTemplate()</td>
<td>214</td>
</tr>
<tr>
<td>setUseTrayIcon()</td>
<td>215</td>
</tr>
<tr>
<td>setUseWebDav()</td>
<td>215</td>
</tr>
<tr>
<td>setVersion()</td>
<td>215</td>
</tr>
<tr>
<td>setWebDavPassword()</td>
<td>216</td>
</tr>
<tr>
<td>setWebDavUrl()</td>
<td>216</td>
</tr>
<tr>
<td>setWebDavUser()</td>
<td>217</td>
</tr>
</tbody>
</table>

### 12.21 RealPass Class Reference

#### 12.21.1 Detailed Description

#### 12.21.2 Constructor & Destructor Documentation

- **12.21.2.1 RealPass()**
- **12.21.2.2 ~RealPass()**

#### 12.21.3 Member Function Documentation

- **12.21.3.1 Copy()**
- **12.21.3.2 GitInit()**
- **12.21.3.3 GitPull()**
- **12.21.3.4 GitPull_b()**
- **12.21.3.5 GitPush()**
Chapter 1

QtPass

1.1 Introduction

QtPass is a multi-platform GUI for pass, the standard unix password manager.

https://qtpass.org/

1.2 Installation

1.2.1 Dependencies

- QtPass requires Qt 5.2 or later.
- The Linguist package is required to compile the translations.
- For use of the fallback icons the SVG library is required.

At runtime the only real dependency is gpg2 but to make the most of it, you'll need git and pass too.

1.2.2 From source

On most *nix systems all you need is:

qmake && make && make install
Chapter 2

Change Log

Unreleased

Full Changelog

Implemented enhancements:

• Set correct WM_CLASS for the qr-code popup  #506

Fixed bugs:

• Will not run on Windows 10 1903 b18362.418  #486

Closed issues:

• Icons are blurry when fractional scaling is enabled  #525
• [Request] clear search password when change profile  #524
• Copying not possible on Ubuntu 20.04  #521
• Display passwords in mono space font  #514
• QtPass 1.3.2 for Ubuntu 19.10 (eoan)  #512
• Default password visibility  #511
• Consider mentioning export abilities in migration docs, if any are present  #505
• Enable out-of-source (shadow) builds.  #501
• password visibility can't be fully hidden  #496
• Translations need updating and checking  #488
• Front-End doesn't work well with HiDPI screen  #464
• How to let QtPass use the real "pass" on windows  #458
• Fresh install of Antergos with Deepin - High DPI scaling is not working  #417
Change Log

- Strange behavior when clearing filter  #402
- Tray icon remains after quitting program  #401
- QtPass doesn't work will pass in WSL  #375
- UI is blurry on HiDPI screens on MacOS (retina) since 1.2.x  #355
- No prompt for passphrase for git key on windows.  #317
- Config dialog's Password Generation field got crowded between 1.1.3 and 1.1.6  #278

Merged pull requests:

- Translations update from Weblate  #530 (weblate)
- Clear search on profile change  #529 (cmol)
- #514 Show password with a monospace font  #528 (cmol)
- Update minimum Qt version  #527 (cmol)
- Fix blurry icons when fractional scaling is enabled  #526 (mthw0)
- Spelling: Git pull, Git push  #516 (comradekingu)
- Enable ubuntu, windows and macos based builds for CI  #508 (boppybibles)
- Enable out-of-source build  #503 (boppybibles)
- Don't base pass-otp availability decision on hardcoded /usr/lib  #499 (nh2)
- Spelling: Search for users, ,  #495 (comradekingu)
- Spelling: Keylist missing, Could not fetch, GPG  #493 (comradekingu)
- Spelling: Git, GPG, PWGen, etc.  #492 (comradekingu)
- Don't use a deprecated method  #491 (amarsman)
- Issue #402: 'deselect()' on clearing filter  #490 (petr-nehez)

v1.3.2 (2019-10-09)

Full Changelog

Fixed bugs:

- QtPass could not run on Windows7 thin  #485
- Segfault on application startup (macos)  #481
- Application crashes on empty password store  #466
- App is completely broken  #423

Closed issues:

- Edit window on Gnome has no padding around  #484
• Buttons width on RHEL 8  #483
• 'Start minimized' no longer works  #471
• Editor doesn’t wait for PGP key to decrypt  #470
• v1.3.0 Data Not Showing  #465
• Hangs on MacOS after Security Update 2019-003 10.12.6  #461
• No public key  #308

Merged pull requests:
• Don’t call QtPass::setup() from QtPass class constructor (should fix #466)  #482  (maciejsszmigiero)

v1.3.1 (2019-10-01)

Full Changelog

Implemented enhancements:
• Renaming password  #463
• [Feature Request] Edit main title field  #446

Fixed bugs:
• build: dependency issue  #467
• is running but no gui  #451

Closed issues:
• Additional lines (notes) are not shown  #474
• Bundle ID is literally  $\texttt{\$(PRODUCT\_BUNDLE\_IDENTIFIER)}\$  #448

Merged pull requests:
• Add license scan report and status  #480  (fossabot)
• Build system related fixes  #479  (maciejsszmigiero)
• Add missing overrides  #478  (amarsman)
• Main window entry details improvements  #477  (maciejsszmigiero)
• Fix html links color and NL translation building error  #476  (a-andreyev)
• Restore directories-first order of passwords tree view on non-Mac platforms  #475  (maciejsszmigiero)
• Add missing finishedShow() signal connection in PasswordDialog constructor (fixes the "Edit password" function)  #473  (maciejsszmigiero)
• Sorted profiles dropdown as in #404  #472  (Noettore)
• Add support for passwords and directories renaming as requested in #463  #469  (Noettore)
• Fix missing app ID and icon on Wayland.  #468  (lightbulbjim)
v1.3.0 (2019-08-20)

Full Changelog

Implemented enhancements:

- Localization makes commits absolutely unreadable #405
- Add otp (two factor authentication) support #327
- Open specific entry from command-line parameter #32

Fixed bugs:

- Windows sigsev issues #326
- Access to the / (root) directory form within the application window on MacOS #302
- PRNG seeding is done totally wrong #238
- Context menu on transparent fields is transparent too . . #227

Closed issues:

- various issues with Info.plist file on macOS #457
- Can not add new passwords for some reason #454
- GnuPG not found on Linux Mint #433
- How to clean up the app #429
- LAN sync request #427
- Profiles can not be removed #415
- Compilation error in (K)ubuntu 16.04.5 with sources tar.gz from version 1.2.3 #408
- Prevent from removing whole password-store directory and hidden directories and files #400
- Version information string's #398
- We should select a C++ std too #372
- We should select a minimum Qt version #371
- Problem with GNUpg not found on MacOS #362
- Compiling for Linux Mint 18 Ubuntu 16 #357
- make qtpass portable in windows #356
- Unable to see main application window (applicationn runs minimized to tray only) #286
- Startup variables and parameters #212
- [OS X] Password input dialog suddenly stopped popping up #191
- MainWindow is a giant monolithic mess #107

Merged pull requests:
• Use key fingerprint as ID instead of “long” ID.  #452 (Natureshadow)

• Typo: dialouge to dialogue.  #444 (georgjaehnig)

• Scripts and logic specific to Windows Store releases  #439 (rdoeffinger)

• For config check, check that the selected binary is available.  #438 (rdoeffinger)

• Fix character encoding issues for non-UTF-8 locales.  #435 (rdoeffinger)

• Fixes and improvements for config dialog  #432 (rdoeffinger)

• Support for using WSL binaries on Windows  #431 (rdoeffinger)

• Bugfixes and Windows compatibility improvements  #430 (mrsch)

• Semi-automatic code cleanup  #425 (annejan)

• Update to prevent the installer requesting admin  #424 (hughwilliams94)

• Display passwords as QR codes  #421 (frawi)

• Tested working on Mac Os HS with pinentry-mac  #419 (riccardocossu)

• Dutch (nl) translation improvements  #418 (equaeghe)

• Bugfixes  #413 (rdoeffinger)

• pwgen: fix inverted “Generate ... less secure passwords” checkbox  #409 (ahippo)

• Continuing refactoring  #407 (FiloSpaTeam)

• #390 make box checked when opening a folder users panel  #403 (kenji21)

v1.2.3 (2018-06-04)

Full Changelog

Closed issues:

• Consider repology badges  #396

• Unable to create new password  #391

• Duplicate prefix in installation of tests directory in v1.2.2.  #389

• Compilation error on FreeBSD member access into incomplete type  #388

• No icons on macOS  #377

Merged pull requests:

• Add support for OTP code generation on Linux as requested in #327  #394 (Noetttore)

• Revert scroll bar changes  #393 (destanyol)

• Fix High Dpi Support. Works now under Windows and KDE/Plasma.  #392 (hgraeber)
v1.2.2 (2018-05-07)

Full Changelog

Implemented enhancements:

• Cleaning #includes #364 (FiloSpaTeam)

Fixed bugs:

• Insecure Password Generation #338
• Clipboard clearing timer is not reset when new passwords are copied to the clipboard #309
• Removal of files outside of password-store #300
• Some fixes and refactoring. #376 (FiloSpaTeam)
• Fix & make clearClipboard more robust #359 (lukedirtwalker)

Closed issues:

• Multiple question marks while trying to delete password #385
• No button icons and text in “menu bar” #383
• Cannot add a new password #380
• Tiny bit of regression #379
• Running qtPass remotely not prompting for the GPG key passphrase #374
• Entire program is huge on High DPI screen on Linux #369
• Two new issues since latest refactoring #368
• Chocolatey package outdated #366
• How do I change the language? #352
• Parallel make issue in qtPass-1.2.1: ld: cannot find -ltqpass #350
• "copy" icon has disappeared in v1.2.1 #344
• No password entry prompt #343
• Can’t install on OS X Sierra #337
• No icon on MacOS #333
• Font and spacing used for URL links on right in main window absurdly large #329
• QtPass don’t display all lines with templates #273

Merged pull requests:

• 2 simple fixes #386 (FiloSpaTeam)
• Should fix #383 #384 (FiloSpaTeam)
• Move connect action to main.cpp. Default search text as parameter of...  #382 (FiloSpaTeam)
• fix #380  #381 (FiloSpaTeam)
• Small refactoring.  #378 (FiloSpaTeam)
• Sorry for last error :)  #370 (FiloSpaTeam)
• Optimizations :)  #367 (FiloSpaTeam)
• Removed comment out #includes #365 (FiloSpaTeam)
• fix for #300  #363 (FiloSpaTeam)
• Translated all missing content to Italian, created Release of transla...  #361 (FiloSpaTeam)
• Refactoring  #360 (lukedirtwalker)
• Display all fields when using template setting, fixes #273  #358 (lukedirtwalker)
• Update CONTRIBUTING.md  #354 (5bentz)
• Add two entries in FAQ about the language  #353 (5bentz)
• Fix typo in french translation  #349 (babolivier)
• New scroll bar on large files  #347 (destanyol)
• Fix nested template argument list compile error  #346 (martinburchell)
• Honor PREFIX during tests install  #345 (SpiderX)

**v1.2.1 (2018-01-04)**

Full Changelog

Closed issues:

• Question: is it possible to mass import passes?  #339
• Version 1.2.0 leaks passwords  #334
• signed release files  #332
• 2017  #330
• When importing settings from 1.1.5 or older clipboard settings revert to No Clipboard  #232

Merged pull requests:

• Insecure password generation  #342 (annejan)
• Add Catalan translation  #336 (rbuj)

Generated by Doxygen
v1.2.0 (2017-11-08)

Full Changelog

Implemented enhancements:

• Icon tray from system icon theme  #318
• Copy button for each custom field  #291
• Feature Request: Use primary selection instead of clipboard  #280
• Add primary selection as clipboard option  #281 (annejan)
• Feature: CTRL/CMD + Q closes the mainwindow #258  #259 (YoshiMan)
• Feature/testing moved sources to src added tests  #257 (annejan)
• enabled drag and drop support for passwords and passwordfolders  #245 (YoshiMan)

Fixed bugs:

• Do not hide passwords and no generator  #267
• Weird behavior when turning on git support (auto push/pull) with non-clean git dir  #128
• SingleApplication implementation buggy  #26

Closed issues:

• Tab order is wrong in password dialog  #331
• Missing icons since split to static lib  #325
• "-session XXX" upon session restore taken as search string  #320
• Instructions to install it on OSX maybe outdated  #315
• QtPass hangs when trying to decrypt entry  #313
• Unable to locate package (Linux Mint 17.3)  #310
• Git commit signing  #303
• Add to Linux brew  #301
• Pass 1.7 testing  #299
• Measure unit-test code coverage  #298
• Config dialog: Propose "Password behaviour" label change  #294
• make install currently broken.  #289
• Unable to locate package (Raspbian)  #287
• There is no git cp  #272
• pass is apparently switching out pwgen  #264
• Bugs since refactoring  #262
• pass working fine but qtprocess failure with qtpass  #260

Generated by Doxygen
• Feature: CTRL/CMD + Q closes the mainwindow  #258
• Refactoring: removal of lastDecrypt  #256
• Pass environment not set-up correctly  #250
• Make fails - std c++11 not set  #244
• Double-clicking might open previous entry instead of one double-clicked on  #243
• Clean up ConfigDialog  #235

Merged pull requests:

• Extract static library and separate main function  #324  (tezeb)
• galego actualizado  #323  (xmgz)
• Add sftp, ftps, webdav and webdavs as supported links  #322  (cgonzalez)
• Ignore cmdline arguments if -session is used.  #321  (Achimh3011)
• Finished French translation (and proof-read the already translated strings).  #311  (Marcool04)
• Once again, code coverage  #305  (tezeb)
• Fixed path of resources.qrc  #297  (sideeffect42)
• Add pt_PT translation  #295  (keitalbame)
• Update README.md  #293  (joostruis)
• small band aid fix for password generation on windows  #276  (treat1)
• Final step in process mgmt refactoring  #275  (tezeb)
• Fix pwgen and refactor Pass::finished  #271  (tezeb)
• Process specific signals for process management  #270  (tezeb)
• #239 reencrypting after a drag and drop action  #261  (YoshiMan)
• this if evaluetes ervery time to true  #255  (YoshiMan)
• executeing pass show before editpassword dialog shows up  #254  (YoshiMan)
• Minor fix for file names and git push  #251  (tezeb)
• Process management refactoring part 2  #249  (tezeb)

v1.1.6 (2016-12-02)

Full Changelog

Implemented enhancements:

• Feedback on copy button use  #229
• Clickable url's + open in default browser  #226
• Deselecting password re-opens the file  #221
• Copy password button should include tooltip to say why, when disabled  #214
• **QtPass** starts by searching for `-psn_0_12345` on macOS  #213

• Copy after timeout  #189

• Feature Request: Copy template fields with button  #133

• Cannot create top level folder  #127

• Feature: moving items (reordering folders)  #116

• Password dialog decoupling from MW  #242 (tezeb)

• Refactoring of qpushbuttonwithclipboard and timers  #241 (tezeb)

• added a copy button for each line to paste the content into the clipboard, "pass init -- path=" command with right path-parameter, lupdate qtpass.pro  #218 (YoshiMan)

**Fixed bugs:**

• Regression with new view mode when using templates and urls  #223

• Problems with high dpi screen  #217

• Hangs forever on Generate GnuPG keypair  #215

• Copy after timeout  #189

• recent change to passworddialog.cpp  #188

• Re-opening entry in QtPass on Windows does not put login or url values back in the right place  #183

**Closed issues:**

• Click does not stick  #233

• Doubleclick on Treeview does not open the edit dialogue  #228

• Windows - Enable GPG SSH Authentication  #225

• We need autotype . .  #65

**Merged pull requests:**

• refactoring - pass ifce, process mgmt  #234 (tezeb)

• Solve Doubleclick issue  #230 (jounathaen)

• refactoring, new QtPassSettings class, all settings should be read and written here  #224 (YoshiMan)

• Moved @YoshiMan’s copy buttons inside the line Edit  #222 (jounathaen)

• UI Improvements  #220 (jounathaen)

• creating password store directory, if it does not exist, de_DE translation fixes and removed obsolete translations  #216 (YoshiMan)
**v1.1.5 (2016-10-19)**

Full Changelog

**Implemented enhancements:**

- I translated for Simplified Chinese. #208
- Short fullname hangs QtPass keypair generation process for infinite time #202
- More options for password generation #98
- Git hangs on windows #71

**Fixed bugs:**

- view box is trimming whitespace #210
- Short fullname hangs QtPass keypair generation process for infinite time #202

**Closed issues:**

- PREFIX is now really a prefix #185
- QtPass, git and windows #173

**Merged pull requests:**

- Allow ssh links #211 (cgonzalez)
- Increase maximum password length to 255 #209 (vladimiroff)
- Password templates #207 (jounathaen)
- Updated German Translation #206 (jounathaen)
- Italian translation #204 (dakk)
- keygendialog email and name validation (issue 202) #203 (dakk)
- Lookup validity field to check if keys are valid #201 (thotypous)
- Fix spelling error #200 (innir)
v1.1.4 (2016-09-26)

Full Changelog

Implemented enhancements:

- Re-assign permissions when adding users  #161
- Main window immediately closes upon app launch  #139

Fixed bugs:

- German umlauts fails  #192
- Error after change configuration  #190
- Re-assign permissions when adding users  #161
- Bug: Special characters in Template  #131
- Character encoding issue with GPG key  #101
- saved password ‘§’ turns to ‘Â§’ when copied to clipboard or shown when editing  #91

Closed issues:

- Signed releases  #186
- Why it’s not listed in wikipedia.org/wiki/List_of_password_managers ?  #164
- Bitdefender blocks installation and quarantines the .exe and .ink  #138

Merged pull requests:

- issue 91 bugfix  #199 (asalamon74)
- issue 101 bugfix  #198 (asalamon74)
- ArchLinux: moved from AUR to [community]  #196 (eworm-de)
- Czech translation  #195 (svetlemody)

v1.1.3 (2016-06-10)

Full Changelog

Fixed bugs:

- edit of password broken with active "Automatically push"  #177
- Clipboard not cleared when quitting or killing application  #171
- Hide content doesn’t work when using templates  #160

Closed issues:

- Add a (small) manpage  #174
v1.1.2 (2016-06-10)

Full Changelog

Implemented enhancements:

• qtpass on windows, space in front of URL and Username  #182

Fixed bugs:

• Deletion of folder doesn't work on Debian/GNU Linux  #181

Closed issues:

• gpg: decryption failed: No secret key  #179
• "gpg-agent: command get\_passphrase failed: No such file or directory"  #156

Merged pull requests:

• add Appdata file and update desktop file  #178 (daveol)
• HTTPS everywhere  #176 (da2x)
• Fix build issues with MSVC2015 on Windows  #175 (msvi)

v1.1.1 (2016-04-04)

Full Changelog

Implemented enhancements:

• Signed binaries  #149
• Icon theme and Cinnamon  #146
• Bind a key to the clear action  #142
• Installation dependencies  #140
• All text input fields need example text & edit dialogue changes  #85
• OSX: Qt-window closed only reappears when 'active' and using tray icon  #77

Fixed bugs:

• Spelling bug: German translation of push and pull  #110
• gpg: decryption failed: No secret key  #92

Generated by Doxygen
Closed issues:

- Remove outdated Debian packaging  #165
- Same name for file and folder  #159
- Icons don't work on nixos  #157
- gpg: Sorry, we are in batchmode - can't get input  #151

Merged pull requests:

- lupdate and Russian translation  #170 (ahippo)
- Remove path to password store in commit message and a leading space.  #169 (ahippo)
- Use –secure for pwgen and add more configurable options  #168 (ahippo)
- Remove Debian packaging  #166 (innir)
- Add caskroom URL  #163 (graingert)
- update gl_Es  #162 (xmgz)
- Two UI Tweaks  #158 (lftl)
- configwindow.ui default/start tab set to "settings"  #154 (jounathaen)
- FAQ update concerning button-icons on cinnamon  #153 (jounathaen)

v1.1.0 (2016-01-25)

Full Changelog

Implemented enhancements:

- Clear text input: use system icon instead of x  #84

Closed issues:

- [resolved] Error in compiling Mac OS El capitan  #148

Merged pull requests:

- Pre 1.1 mixing  #145 (annejan)
- Futurator Keygen  #144 (annejan)
- Futurator redesign proper  #141 (annejan)
- RPM Spec file updates  #137 (muff1nman)
- swedish translations  #135 (ralphtheninja)
v1.0.6 (2016-01-03)

Full Changelog

Implemented enhancements:

- Feature: Always on top  #118
- Option to show minimized instance  #99
- System Icons on Buttons and Doubleclick on treeView  #124 (jounathaen)

Fixed bugs:

- Bug: deleted record stays in memory  #117

Closed issues:

- SIGSEGV in MainWindow::executeWrapper on clean install  #122

Merged pull requests:

- improved the German translation  #134 (retokromer)
- qrand always generating the same sequence of passwords  #129 (treat1)
- some improvements  #126 (retokromer)
- added one translation  #125 (retokromer)
- initial attempt to create a RPM spec file  #121 (bram-ivs)
- Cleanup and coding standards  #120 (annejan)
- Modified the clipboard logic to allow for on-demand copy to clipboard.  #119 (jonhanks)

v1.0.5 (2015-11-18)

Full Changelog

Fixed bugs:

- using pwgen adds carriage-return  #115
- Enhancement: color code git results  #111

Merged pull requests:

- Fix bug that prints "Unknown error" to the terminal  #113 (dvaerum)
**v1.0.4 (2015-11-03)**

Full Changelog

**Implemented enhancements:**

- Add support for RightToLeft languages  #108

**v1.0.3 (2015-10-25)**

Full Changelog

**Implemented enhancements:**

- Get PREFIX variable from environment  #106
- Password file named 'git' returns error  #105

**Fixed bugs:**

- Password file named 'git' returns error  #105

**Merged pull requests:**

- Get PREFIX variable from environment  #104 (jorti)
- spanish translations added  #103 (mrpnkt)

**v1.0.2 (2015-09-24)**

Full Changelog

**Closed issues:**

- Generate password: Floating point exception (core dumped)  #102
- A way to indicate the installation prefix is needed  #100
- IPv4 URLs are non-clickable  #97
- app crashes when "Use pwgen" is unselected, and "Generate" is clicked.  #95
- Some minor improvements on the templating part  #93
- app crashes with variant of "pwgen" app  #90
v1.0.1 (2015-08-09)

Full Changelog

Implemented enhancements:

- Users setup - key colours could be improved  #82

Closed issues:

- When QtPass starts, focus search input box  #89
- Clear the password display after some time  #86
- Auto push/pull  #83
- qtpass doesn't commit deletes to git  #81
- Always crashes while using the quick-search input  #79
- Git initialisation  #72
- Initialising new repo's doesn't work correctly  #55
- gpg: Sorry, no terminal at all requested - can't get input  #18

Merged pull requests:

- Issue 86 clear panel  #87 (karlgrz)
- Update FAQ for Yubikey NEO helper in .bashrc for Ubuntu  #80 (karlgrz)
- [WIP] Call 'pass git init' on creation of password-store when useGit  #78 (dennisdegreef)

v1.0.0 (2015-08-01)

Full Changelog

Closed issues:

- Yubikey Neo Pin entry not working properly on Ubuntu 15.04  #73

Merged pull requests:

- Updating hungarian localisation  #76 (damnlie)
- added DE translations  #74 (Friedy)
**v0.9.2 (2015-07-30)**

Full Changelog

Closed issues:

- Show expiration date in key setup  #70

**v0.9.1 (2015-07-29)**

Full Changelog

Closed issues:

- Minimize on startup.  #69
- tray icon in xfce  #58
- Git integration  #57
- Weird characters in filenames breaks loading gpg files  #10

**v0.9.0 (2015-07-17)**

Full Changelog

Closed issues:

- Request: Integrate qtpass with pwgen for generating passwords.  #68

**v0.8.6 (2015-07-17)**

Full Changelog

Closed issues:

- Copy password by Ctrl+C  #60
- Remember window size and vertical pane width  #59
- Multiline Editing  #34

Merged pull requests:

- To make building successful with Desktop Qt 5.4.0 MSVC2012 OpenGL 32bit  #67 (annejan)
v0.8.5.1 (2015-07-08)

Full Changelog

v0.8.5 (2015-07-08)

Full Changelog

Closed issues:

• Won't compile on Kubuntu 15.10   #61
• Hanging process gives weird effects  #56
• Directory separator actually broken by 208171fd09c55ad765df4fa1de9a7f0757fa72d  #53

Merged pull requests:

• Many deadlocks and other nasty bug fixes   #64 (annejan)
• Mention qt5-default package in README   #62 (lorrin)
• Some hacks I needed for portable gpg4win release   #54 (rdoeffinger)

v0.8.4 (2015-06-11)

Full Changelog

Closed issues:

• QtPass does not detect GPG installation   #50
• Cannot create new folders   #48
• Better error handling when no pass or gpg found initially   #13

Merged pull requests:

• Develop   #52 (annejan)
• Minor thingies   #51 (beefcurtains)

v0.8.3 (2015-06-09)

Full Changelog

Merged pull requests:

• Bugfixes   #49 (rdoeffinger)
v0.8.2 (2015-05-27)

Full Changelog

Closed issues:

- Typo in 37f17f3808c1c97bd72c165a530c67a4bfb82ed8?  #45
- Signing of keys from user management  #41

Merged pull requests:

- Provide more information in user list.  #47 (rdoeffinger)
- Enable C++11 and use it to simplify loops.  #46 (rdoeffinger)

v0.8.1 (2015-05-06)

Fixed bugs:

- Some items not found on first search  #8

Closed issues:

- compiling qtpass on ubuntu 15.04 - fails due to newer qmake version  #43
- QProcess::start: Process is already running  #40
- Extra line breaks seem to be added to the (html) output  #39
- Missing develop branch and release testing  #38
- Windows WebDAV broken by 24f8dec3c203921f765e923e6ae6a4069b8cf50a  #36
- .gpg-id file not added to git  #35
- Icon filenames  #31
- GNUPGHOME environment variable  #30
- Feature: webdav alternative to git  #28
- Windows: not working due to pointless use of "sh"  #16
- Windows: support static build and enable ASLR and NX  #15
- Some paths to executables are printed when starting up  #11

Merged pull requests:

- SingleApplication per user and leading newline removed from output  #44 (annejan)
- User filtering and many fixes  #42 (annejan)
• Re-enable Windows WebDAV support. #37 (rdoeffinger)
• User robustness  #33 (rdoeffinger)
• Add WebDAV support. #29 (rdoeffinger)
• Add nosingleapp config. #27 (rdoeffinger)
• Add Makefile with commands to make a binary release zip file. #25 (rdoeffinger)
• Start process only after we finished disabling UI elements etc. #24 (rdoeffinger)
• Support for editing .gpg-id via GUI with public keyring list. #23 (rdoeffinger)
• More proper support for subdirectories. #22 (rdoeffinger)
• Russian translation (+typo fixed)  #20 (mexus)
• Windows-related fixes. #17 (rdoeffinger)
• Deal with "special" characters  #14 (JiCiT)
• galician and spanish localization files created  #12 (xmgz)
• Update localization_hu_HU.ts  #9 (damnlie)
• Replace which invocations with actual path resolution code  #7 (shitbangs)
• Added Swedish and Polish localization to resources  #6 (iamtew)
• Swedish localization  #5 (iamtew)
• Update localization_hu_HU.ts  #4 (reesenemesis)
• Update localization_hu_HU.ts  #3 (reesenemesis)
• [pass](http://www.passwordstore.org/) #2 (guaka)
• Beginning of German translation  #1 (mwfc)

*This Change Log was automatically generated by*  [github_changelog_generator](https://github.com/release-detection/github_changelog_generator)
Chapter 3

Contributing

Make sure you have read the FAQ

Thank you for wanting to contribute to making QtPass awesome.

This document

This document is still in a very early stage and needs a lot more work.

Pull Request Process

1. Ensure install or build dependencies and artefacts are not committed.
2. When adding big new features or changes to the build system, update the README.md to reflect those.
3. Make sure you update all of the CI configs if need be. These are ran on every Pull Request.

1. Increase the version numbers in relevant files when applicable. The versioning scheme we use is SemVer.
2. You may merge the Pull Request in once you have the sign-off of one other developer, or if you do not have permission to do that, you may request a reviewer to merge it for you.

Translations

- Add you language to the src/src.pro file under TRANSLATIONS.
- Next run the command qmake which will create and update the localization files.
- Edit your file with (let's imagine your language is sv_SE (Swedish)) linguist localization/localization-_sv_SE.ts

Qt Linguist has very nice in-context translation options for translators

You can do online translations via Weblate
IRCThis is some text about contributing.

For questions or brainstorming about features please join #ijhack on freenode.

**Gitter**

Or if you prefer to use [gitter](http://github.com).

**License**

QtPass is released under the GNU GPL v3.0 license. [http://www.gnu.org/licenses/gpl-3.0.html](http://www.gnu.org/licenses/gpl-3.0.html)
Chapter 4

FAQ

Issues

Can’t save a password

- Is folder initialised? Easiest way is to use the [Users] button and make sure you can encrypt for someone (eg. yourself)
- Are you using git? If not, make sure it is switched off.

I have an issue with GNOME keyring

- Disable GNOME keyring
- Create a ~/.gnupg/gpg-agent.conf containing:

```bash
enable-ssh-support
write-env-file
use-standard-socket
default-cache-ttl 600
max-cache-ttl 7200
```

Also, the following is useful to add to your .bashrc if you are using Yubikey NEO on Ubuntu:

```bash
# OpenPGP applet support for YubiKey NEO
if [ ! -f /tmp/gpg-agent.env ]; then
  killall gpg-agent;
  eval $(gpg-agent --daemon --enable-ssh-support > /tmp/gpg-agent.env);
fi
. /tmp/gpg-agent.env
```

- More info:  issue 60 and issue 73

I don’t get a passphrase / PIN dialog

- You’ll need to install pinentry-qt (or -qt4 or -qt5 or even -gtk) and possibly set the full path to that executable in your ~/.gnupg/gpg-agent.conf for example: pinentry-program /usr/bin/pinentry-qt4
- On some esoteric systems it might be necessary to create a symbolic link /usr/bin/pinentry to your pinentry application of choice eg: ln -s /usr/bin/pinentry-qt5 /usr/bin/pinentry
- On MacOs pinentry-program /usr/local/bin/pinentry-mac works after installing pinentry-mac from homebrew.
I have an other issue with gpg

- Possibly you have only your key in gpg and not in gpg2.

```shell
gpg --export [ID] > public.key
gpg --export-secret-key [ID] > private.key
gpg2 --import public.key
rm public.key private.key
```

Where [ID] is your gpg key-id.

- It might be the case where it is the other way around, exchange gpg and gpg2 accordingly.

Git doesn’t work on Windows

Git for Windows comes with an ssh-askpass compatible command, `git gui--askpass` (located in `/mingw64/libexec/git-core/git-gui--askpass` on PortableGit version, presumably some similar for the installed version).

Git has issues with GPG SSH Authentication

This tutorial might resolve your issues. [https://github.com/git-for-windows/git/wiki/OpenSSH-Integration-with-Pageant](https://github.com/git-for-windows/git/wiki/OpenSSH-Integration-with-Pageant)

Where is the configuration stored?

QtPass tries to use the native config choice for the OS it’s running.

- **Linux and BSD**: `$HOME/.config/IJHack/QtPass.conf`
- **macOS**: `$HOME/Library/Preferences/com.IJHack.QtPass.plist`
- **Windows registry**: `HKEY_CURRENT_USER\Software\IJhack\QtPass`

These settings can be overruled by a `qtpass.ini` file in the folder where the application resides. So called “portable config”

There are some things to take care of when trying to sync on some systems (especially OSX, with regards to text and binary .plist files).


Where can I ask for help?

- Create an issue on github.
- Send an email to help@qtpass.org

Can I import from KeePass, LastPass or X?

- Yes, check [passwordstore.org/#migration](https://passwordstore.org/#migration) for more info.
I don't see icons on the buttons

You do not have the Qt SVG library installed. Please install using your favorite package manager.

I get icons that do not fit my (X11) default

- On some WindowManagers, Qt doesn't know what icon set to use. A trick:
  export DESKTOP_SESSION=gnome

- Another possible reason is, that the currently installed Qt Version gives problems (e.g. on Linux Mint 17.3)
  Then you'll have to install the current version via your package manager or if this is not up-to-date, download
  it from https://www.qt.io/download/ install it and run:
  /PATHTOYOURQTINSTALLATION/5.5/gcc_64/bin/qmake
  make
  (sudo) make install

  where PATHTOYOURINSTALLATION is the path you selected in the qt installer (default /home/YOURUSER/Qt/) and 5.5 has to be adapted for the Qt version you downloaded.

I don't like the design, what gives?

- It's all on github, clone, change and send a pull request.
- Open an issue and point out defects or better yet propose changes.

QtPass is not in my native language

- Unfortunately, QtPass might not support your native language, or the translations might be incomplete. Check
  if newer versions of QtPass support it.
- If translations are available but aren't working, try to set the language manually (see below) or open an issue.

How do I set the language manually?

QtPass uses the system language. Changing it depends on your system:

- on Linux: LANGUAGE=fr qtpass will run QtPass in French.

How can I help improve QtPass?

I would like to donate!

- Time:
  - Read contributing documentation.
  - Fork, clone hack and send a pull request.
  - Find an issue to work on..
  - Participate in our bug bounty, you submit an issue and help us fix it, I send you a bounty.
- Money:
  - IJhack takes donations in Bitcoin
Chapter 5

QtPass

QtPass is a GUI for pass, the standard unix password manager.

Features

- Using pass or git and gpg2 directly
- Configurable shoulder surfing protection options
- Cross platform: Linux, BSD, OS X and Windows
- Per-folder user selection for multi recipient encryption
- Multiple profiles
- Easy onboarding

Logo based on Heart-padlock by AnonMoos.

Installation

From package

OpenSUSE & Fedora yum install qtpass dnf install qtpass
Debian, Ubuntu and derivates like Mint, Kali & Raspbian apt-get install qtpass
Arch Linux pacman -S qtpass
Gentoo emerge -atv qtpass
Sabayon equo install qtpass
FreeBSD pkg install qtpass
macOS brew cask install qtpass
Windows choco install qtpass
From Source

Dependencies

- QtPass requires Qt 5.10 or later
- The Linguist package is required to compile the translations.
- For use of the fallback icons the SVG library is required.

At runtime the only real dependency is gpg2 but to make the most of it, you'll need git and pass too.

Your GPG has to be set-up with a graphical pinentry when applicable, same goes for git authentication. On Mac OS X this currently seems to only work best with pinentry-mac from homebrew, although gpgtools works too.

On most unix systems all you need is:

```
qmake && make && make install
```

Testing

This is done with `make check`

Codecoverage can be done with `make lcov, make gcov, make coveralls and/or make codecov`.

Be sure to first run: `make distclean && qmake CONFIG+=coverage qtpass.pro`

Security considerations

Using this program will not magically keep your passwords secure against compromised computers even if you use it in combination with a smartcard.

It does protect future and changed passwords though against anyone with access to your password store only but not your keys. Used with a smartcard it also protects against anyone just monitoring/copying all files/keystrokes on that machine and such an attacker would only gain access to the passwords you actually use. Once you plug in your smartcard and enter your PIN (or due to CVE-2015-3298 even without your PIN) all your passwords available to the machine can be decrypted by it, if there is malicious software targeted specifically against it installed (or at least one that knows how to use a smartcard).

To get better protection out of use with a smartcard even against a targeted attack I can think of at least two options:

- The smartcard must require explicit confirmation for each decryption operation. Or if it just provides a counter for decrypted data you could at least notice an attack afterwards, though at quite some effort on your part.
- Use a different smartcard for each (group of) key.
- If using a YubiKey or U2F module or similar that requires a "button" press for other authentication methods you can use one OTP/U2F enabled WebDAV account per password (or groups of passwords) as a quite inconvenient workaround. Unfortunately I do not know of any WebDAV service with OTP support except ownCloud (so you would have to run your own server).
Known issues

- Filtering (searching) breaks the tree/model sometimes
- Starting without a correctly set password-store folder gives weird results in the tree view

Planned features

- Plugins based on field name, plugins follow same format as password files
- Colour coding folders (possibly disabling folders you can't decrypt)
- Optional table view of decrypted folder contents
- Opening of (basic auth) urls in default browser? Possibly with helper plugin for filling out forms?
- WebDAV (configuration) support
- Some other form of remote storage that allows for accountability / auditing (web API to retrieve the .gpg files?)

Further reading

FAQ and CONTRIBUTING documentation. CHangelog

Website Source code Issue queue Chat

License

GNU GPL v3.0

View official GNU site http://www.gnu.org/licenses/gpl.html.

View the Open Source Initiative site.
Chapter 6

Todo List

Member `ConfigDialog::genKey(QString, QDialog *)`
refactor the process to not be entangled so much.

Member `ConfigDialog::wizard()`
make this thing more reliable.

" refactor when process is untangled from `MainWindow` class.
Chapter 7

Namespace Index

7.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<table>
<thead>
<tr>
<th>Enums</th>
<th></th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumerators for configuration and runtime items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ui</td>
<td></td>
<td>46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Chapter 8

## Hierarchical Index

### 8.1 Class Hierarchy

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

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>dragAndDropInfo</td>
<td>60</td>
</tr>
<tr>
<td>dragAndDropInfoPasswordStore</td>
<td>61</td>
</tr>
<tr>
<td>FileContent</td>
<td>70</td>
</tr>
<tr>
<td>NamedValue</td>
<td>101</td>
</tr>
<tr>
<td>PasswordConfiguration</td>
<td>125</td>
</tr>
<tr>
<td>QApplication</td>
<td>237</td>
</tr>
<tr>
<td>QDialog</td>
<td></td>
</tr>
<tr>
<td>ConfigDialog</td>
<td>47</td>
</tr>
<tr>
<td>KeygenDialog</td>
<td>83</td>
</tr>
<tr>
<td>PasswordDialog</td>
<td>127</td>
</tr>
<tr>
<td>UsersDialog</td>
<td>261</td>
</tr>
<tr>
<td>QList</td>
<td></td>
</tr>
<tr>
<td>NamedValues</td>
<td>102</td>
</tr>
<tr>
<td>QMainWindow</td>
<td>85</td>
</tr>
<tr>
<td>QObject</td>
<td></td>
</tr>
<tr>
<td>Executor</td>
<td>62</td>
</tr>
<tr>
<td>Pass</td>
<td>104</td>
</tr>
<tr>
<td>ImitatePass</td>
<td>73</td>
</tr>
<tr>
<td>RealPass</td>
<td>217</td>
</tr>
<tr>
<td>QtPass</td>
<td></td>
</tr>
<tr>
<td>tst_ui</td>
<td>150</td>
</tr>
<tr>
<td>tst_util</td>
<td>254</td>
</tr>
<tr>
<td>QPushButton</td>
<td></td>
</tr>
<tr>
<td>QPushButtonAsQRCode</td>
<td>143</td>
</tr>
<tr>
<td>QPushButtonShowPassword</td>
<td>146</td>
</tr>
<tr>
<td>QPushButtonWithClipboard</td>
<td>148</td>
</tr>
<tr>
<td>QSettings</td>
<td></td>
</tr>
<tr>
<td>QtPassSettings</td>
<td>158</td>
</tr>
<tr>
<td>QSortFilterProxyModel</td>
<td></td>
</tr>
<tr>
<td>StoreModel</td>
<td>241</td>
</tr>
<tr>
<td>QTreeView</td>
<td></td>
</tr>
<tr>
<td>DeselectableTreeView</td>
<td>58</td>
</tr>
<tr>
<td>QWidget</td>
<td></td>
</tr>
</tbody>
</table>
Hierarchical Index

QProgressIndicator ................................................................. 136
TrayIcon ........................................................................... 250
SettingsConstants ................................................................. 224
simpleTransaction ............................................................... 235
ImitatePass ...................................................................... 73
UserInfo ......................................................................... 257
Util ............................................................................... 264
Chapter 9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigDialog</td>
<td>The ConfigDialog handles the configuration interface</td>
<td>47</td>
</tr>
<tr>
<td>DeselectableTreeView</td>
<td>The DeselectableTreeView class loosely based on <a href="https://stackoverflow.com/questions/2761284/">http://stackoverflow.com/questions/2761284/</a> thanks to Yassir Ennazk</td>
<td>58</td>
</tr>
<tr>
<td>dragAndDropInfo</td>
<td>Holds values to share between drag and drop on the passwordstorage view</td>
<td>60</td>
</tr>
<tr>
<td>Executor</td>
<td>Executes external commands for handling password, git and other data</td>
<td>62</td>
</tr>
<tr>
<td>FileContent</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>ImitatePass</td>
<td></td>
<td>73</td>
</tr>
<tr>
<td>KeygenDialog</td>
<td>Handles GPG keypair generation</td>
<td>83</td>
</tr>
<tr>
<td>MainWindow</td>
<td>Does way too much, not only is it a switchboard, configuration handler and more, it's also the process-manager</td>
<td>85</td>
</tr>
<tr>
<td>NamedValue</td>
<td>Mostly a list of NamedValue but also has a method to take a specific NamedValue pair out of the list</td>
<td>101</td>
</tr>
<tr>
<td>PasswordConfiguration</td>
<td>Holds the Password configuration settings</td>
<td>102</td>
</tr>
<tr>
<td>PasswordDialog</td>
<td>PasswordDialog Handles the inserting and editing of passwords</td>
<td>104</td>
</tr>
<tr>
<td>QProgressIndicator</td>
<td>Lets an application display a progress indicator to show that a lengthy task is under way</td>
<td>125</td>
</tr>
<tr>
<td>QPushButtonAsQRCode</td>
<td>Stylish widget to display the field as QR Code</td>
<td>127</td>
</tr>
<tr>
<td>QPushButtonShowPassword</td>
<td></td>
<td>136</td>
</tr>
<tr>
<td>QPushButtonWithClipboard</td>
<td>Stylish widget to allow copying of password and account details</td>
<td>143</td>
</tr>
<tr>
<td>Class Index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QtPass</td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>QtPassSettings</td>
<td>Singleton that stores qtpass' settings, saves and loads config</td>
<td>158</td>
</tr>
<tr>
<td>RealPass</td>
<td>Wrapper for executing pass to handle the password-store</td>
<td>217</td>
</tr>
<tr>
<td>SettingsConstants</td>
<td>Table for the naming of configuration items</td>
<td>224</td>
</tr>
<tr>
<td>simpleTransaction</td>
<td></td>
<td>235</td>
</tr>
<tr>
<td>SingleApplication</td>
<td>Used for commandline integration</td>
<td>237</td>
</tr>
<tr>
<td>StoreModel</td>
<td>The QSortFilterProxyModel for handling filesystem searches</td>
<td>241</td>
</tr>
<tr>
<td>TrayIcon</td>
<td>Handles the systemtray icon and menu</td>
<td>250</td>
</tr>
<tr>
<td>tst_ui</td>
<td>Our first unit test</td>
<td>254</td>
</tr>
<tr>
<td>tst_util</td>
<td>Our first unit test</td>
<td>255</td>
</tr>
<tr>
<td>UserInfo</td>
<td>Stores key info lines including validity, creation date and more</td>
<td>257</td>
</tr>
<tr>
<td>UsersDialog</td>
<td>Handles listing and editing of GPG users</td>
<td>261</td>
</tr>
<tr>
<td>Util</td>
<td>Some static utilities to be used elsewhere</td>
<td>264</td>
</tr>
</tbody>
</table>
Chapter 10

File Index

10.1 File List

Here is a list of all files with brief descriptions:

- main/main.cpp .......................................................... 271
- src/configdialog.cpp ................................................. 272
- src/configdialog.h ................................................... 273
- src/debughelper.h .................................................... 274
- src/deselectabletreeview.h ......................................... 275
- src/enums.h ............................................................ 275
- src/executor.cpp ..................................................... 276
- src/executor.h ........................................................ 276
- src/filecontent.cpp .................................................. 277
- src/filecontent.h .................................................... 277
- src/imitatepass.cpp .................................................. 278
- src/imitatepass.h .................................................... 279
- src/keygendialog.cpp ............................................... 279
- src/keygendialog.h ................................................... 280
- src/mainwindow.cpp ................................................ 280
- src/mainwindow.h .................................................... 281
- src/pass.cpp .......................................................... 282
- src/pass.h ............................................................. 283
- src/passwordconfiguration.h ...................................... 283
- src/passworddialog.cpp .............................................. 284
- src/passworddialog.h ............................................... 285
- src/qprogressindicator.cpp ...................................... 286
- src/qprogressindicator.h .......................................... 286
- src/qpushbuttonasqrcode.cpp .................................... 287
- src/qpushbuttonasqrcode.h ...................................... 288
- src/qpushbuttonshowpassword.cpp ............................... 288
- src/qpushbuttonshowpassword.h .................................. 289
- src/qpushbuttonwithclipboard.cpp ............................. 290
- src/qpushbuttonwithclipboard.h ................................. 290
- src/qtpass.cpp ....................................................... 291
- src/qtpass.h .......................................................... 292
- src/qtpasssettings.cpp ............................................. 293
- src/qtpasssettings.h ............................................... 293
- src/realpass.cpp ..................................................... 294
- src/realpass.h ........................................................ 294
<table>
<thead>
<tr>
<th>File Name</th>
<th>Line No</th>
</tr>
</thead>
<tbody>
<tr>
<td>src/settingsconstants.cpp</td>
<td>295</td>
</tr>
<tr>
<td>src/settingsconstants.h</td>
<td>295</td>
</tr>
<tr>
<td>src/simpletransaction.cpp</td>
<td>296</td>
</tr>
<tr>
<td>src/simpletransaction.h</td>
<td>296</td>
</tr>
<tr>
<td>src/singleapplication.cpp</td>
<td>297</td>
</tr>
<tr>
<td>src/singleapplication.h</td>
<td>298</td>
</tr>
<tr>
<td>src/storemodel.cpp</td>
<td>298</td>
</tr>
<tr>
<td>src/storemodel.h</td>
<td>300</td>
</tr>
<tr>
<td>src/trayicon.cpp</td>
<td>300</td>
</tr>
<tr>
<td>src/trayicon.h</td>
<td>301</td>
</tr>
<tr>
<td>src/userinfo.h</td>
<td>302</td>
</tr>
<tr>
<td>src/usersdialog.cpp</td>
<td>302</td>
</tr>
<tr>
<td>src/usersdialog.h</td>
<td>303</td>
</tr>
<tr>
<td>src/util.cpp</td>
<td>304</td>
</tr>
<tr>
<td>src/util.h</td>
<td>305</td>
</tr>
<tr>
<td>tests/auto/ui/tst_ui.cpp</td>
<td>305</td>
</tr>
<tr>
<td>tests/auto/util/tst_util.cpp</td>
<td>306</td>
</tr>
</tbody>
</table>
Chapter 11

Namespace Documentation

11.1 Enums Namespace Reference

Enumerators for configuration and runtime items.

Enumerations

- enum clipBoardType { CLIPBOARD_NEVER = 0, CLIPBOARD_ALWAYS = 1, CLIPBOARD_ON_DEMAND = 2 }
- enum PROCESS {
  GIT_INIT = 0, GIT_ADD, GIT_COMMIT, GIT_RM,
  GIT_PULL, GIT_PUSH, PASS_SHOW, PASS_INSERT,
  PASS_REMOVE, PASS_INIT, GPG_GENKEYS, PASS_MOVE,
  PASS_COPY, GIT_MOVE, GIT_COPY, PROCESS_COUNT,
  INVALID, PASS_OTP_GENERATE }

11.1.1 Detailed Description

Enumerators for configuration and runtime items.

11.1.2 Enumeration Type Documentation

11.1.2.1 clipBoardType

enum Enums::clipBoardType

<table>
<thead>
<tr>
<th>Enumerator</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIPBOARD_NEVER</td>
</tr>
<tr>
<td>CLIPBOARD_ALWAYS</td>
</tr>
<tr>
<td>CLIPBOARD_ON_DEMAND</td>
</tr>
</tbody>
</table>
11.1.2.2 PROCESS

enum Enums::PROCESS

<table>
<thead>
<tr>
<th>Enumerator</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIT_INIT</td>
</tr>
<tr>
<td>GIT_ADD</td>
</tr>
<tr>
<td>GIT_COMMIT</td>
</tr>
<tr>
<td>GIT_RM</td>
</tr>
<tr>
<td>GIT_PULL</td>
</tr>
<tr>
<td>GIT_PUSH</td>
</tr>
<tr>
<td>PASS_SHOW</td>
</tr>
<tr>
<td>PASS_INSERT</td>
</tr>
<tr>
<td>PASS_REMOVE</td>
</tr>
<tr>
<td>PASS_INIT</td>
</tr>
<tr>
<td>GPG_GENKEYS</td>
</tr>
<tr>
<td>PASS_MOVE</td>
</tr>
<tr>
<td>PASS_COPY</td>
</tr>
<tr>
<td>GIT_MOVE</td>
</tr>
<tr>
<td>GIT_COPY</td>
</tr>
<tr>
<td>PROCESS_COUNT</td>
</tr>
<tr>
<td>INVALID</td>
</tr>
<tr>
<td>PASS_OTP_GENERATE</td>
</tr>
</tbody>
</table>

11.2 Ui Namespace Reference
12.1 ConfigDialog Class Reference

The ConfigDialog handles the configuration interface.

```c++
#include <configdialog.h>
```

Inheritance diagram for ConfigDialog:

```
QDialog
  ConfigDialog
```

Collaboration diagram for ConfigDialog:

```
QDialog
  ConfigDialog
```
Public Member Functions

- ConfigDialog (MainWindow *parent)
  
  ConfigDialog::ConfigDialog this sets up the configuration screen.

- ~ConfigDialog ()
  
  ConfigDialog::~ConfigDialog config destructor, makes sure the mainWindow knows about git, gpg and pass executables.

- void useSelection (bool useSelection)
  
  ConfigDialog::useSelection set the clipboard type use from MainWindow.

- void useAutoclear (bool useAutoclear)
  
  ConfigDialog::useAutoclear set the clipboard autoclear use from MainWindow.

- void useAutoclearPanel (bool useAutoclearPanel)
  
  ConfigDialog::useAutoclearPanel set the panel autoclear use from MainWindow.

- QHash<QString, QString> getProfiles ()
  
  ConfigDialog::getProfiles return profile list.

- void wizard ()
  
  ConfigDialog::wizard first-time use wizard.

- void genKey (QString, QDialog *)
  
  ConfigDialog::genKey tunnel function to make MainWindow generate a gpg key pair.

- void useTrayIcon (bool useSystray)
  
  ConfigDialog::useTrayIcon set preference for using trayicon. Enable or disable related checkboxes accordingly.

- void useGit (bool useGit)
  
  ConfigDialog::useGit set preference for using git.

- void useOtp (bool useOtp)
  
  ConfigDialog::useOtp set preference for using otp plugin.

- void useQrencode (bool useQrencode)
  
  ConfigDialog::useOtp set preference for using otp plugin.

- void setPwgenPath (QString)
  
  ConfigDialog::setPwgenPath set pwgen executable path. Enable or disable related options in the interface.

- void usePwgen (bool usePwgen)
  
  ConfigDialog::usePwgen set preference for using pwgen (can be overruled buy empty pwgenPath). enable or disable related options in the interface via ConfigDialog::on_checkBoxUsePwgen_clicked.

- void setPasswordConfiguration (const PasswordConfiguration &config)
  
  PasswordConfiguration getPasswordConfiguration ()

- void useTemplate (bool useTemplate)
  
  ConfigDialog::useTemplate set preference for using templates.

Protected Member Functions

- void closeEvent (QCloseEvent *)
  
  ConfigDialog::closeEvent close this window.

12.1.1 Detailed Description

The ConfigDialog handles the configuration interface.

This class should also take the handling from the MainWindow class.

Definition at line 24 of file configdialog.h.
12.1.2 Constructor & Destructor Documentation

12.1.2.1 ConfigDialog()

ConfigDialog::ConfigDialog ( 
    MainWindow * parent ) [explicit]

ConfigDialog::ConfigDialog this sets up the configuration screen.

Parameters

- parent

Definition at line 25 of file configdialog.cpp.
Here is the call graph for this function:

12.1.2.2 ~ConfigDialog()

ConfigDialog::~ConfigDialog ()

Generated by Doxygen
**ConfigDialog::~ConfigDialog** config destructor, makes sure the mainWindow knows about git, gpg and pass executables.

Definition at line 132 of file configdialog.cpp.

Here is the call graph for this function:

![Call Graph](image)

### 12.1.3 Member Function Documentation

#### 12.1.3.1 closeEvent()

```cpp
void ConfigDialog::closeEvent ( QCloseEvent * event ) [protected]
```

**ConfigDialog::closeEvent** close this window.

**Parameters**

- **event**

Definition at line 702 of file configdialog.cpp.

#### 12.1.3.2 genKey()

```cpp
void ConfigDialog::genKey ( QString batch, QDialog * dialog )
```

**ConfigDialog::genKey** tunnel function to make MainWindow generate a gpg key pair.

**Todo** refactor the process to not be entangled so much.

**Parameters**

- **batch**
- **dialog**
Definition at line 447 of file configdialog.cpp.
Here is the call graph for this function:

12.1.3.3 getPasswordConfiguration()

PasswordConfiguration ConfigDialog::getPasswordConfiguration ()
Definition at line 809 of file configdialog.cpp.

12.1.3.4 getProfiles()

QHash<QString, QString> ConfigDialog::getProfiles ()
ConfigDialog::getProfiles return profile list.
Returns

Definition at line 485 of file configdialog.cpp.

12.1.3.5 setPasswordConfiguration()

void ConfigDialog::setPasswordConfiguration ( const PasswordConfiguration & config )
Definition at line 800 of file configdialog.cpp.
Here is the caller graph for this function:
12.1.3.6 setPwgenPath()

void ConfigDialog::setPwgenPath (  
    QString pwgen  )

ConfigDialog::setPwgenPath set pwgen executable path. Enable or disable related options in the interface.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwgen</td>
<td></td>
</tr>
</tbody>
</table>

Definition at line 762 of file configdialog.cpp.

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> ConfigDialog::setPwgenPath
```

12.1.3.7 useAutoclear()

void ConfigDialog::useAutoclear (  
    bool useAutoclear  )

ConfigDialog::useAutoclear set the clipboard autoclear use from MainWindow.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>useAutoclear</td>
<td></td>
</tr>
</tbody>
</table>

Definition at line 409 of file configdialog.cpp.

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> ConfigDialog::useAutoclear
```
### 12.1.3.8 useAutoclearPanel()

```cpp
void ConfigDialog::useAutoclearPanel (  
    bool useAutoclearPanel )
```

*ConfigDialog::useAutoclearPanel* set the panel autoclear use from *MainWindow*.

**Parameters**

- `useAutoclearPanel`

Definition at line 419 of file configdialog.cpp.

Here is the caller graph for this function:

![Caller Graph](image)

### 12.1.3.9 useGit()

```cpp
void ConfigDialog::useGit (  
    bool useGit )
```

*ConfigDialog::useGit* set preference for using git.

**Parameters**

- `useGit`

Definition at line 711 of file configdialog.cpp.

Here is the caller graph for this function:

![Caller Graph](image)
12.1.3.10 useOtp()

```c++
void ConfigDialog::useOtp (
    bool useOtp )
```

*ConfigDialog::useOtp* set preference for using otp plugin.

**Parameters**

- `useOtp`

Definition at line 720 of file configdialog.cpp.

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog ➔ ConfigDialog::useOtp
```

12.1.3.11 usePwgen()

```c++
void ConfigDialog::usePwgen (
    bool usePwgen )
```

*ConfigDialog::usePwgen* set preference for using pwgen (can be overruled by empty pwgenPath). Enable or disable related options in the interface via `ConfigDialog::on_checkBoxUsePwgen_clicked`.

**Parameters**

- `usePwgen`

Definition at line 793 of file configdialog.cpp.

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog ➔ ConfigDialog::usePwgen
```
12.1.3.12  useQrencode()

```cpp
void ConfigDialog::useQrencode (  
    bool useQrencode  )
```

ConfigDialog::useOtp set preference for using otp plugin.

Parameters

| useOtp |

Definition at line 728 of file configdialog.cpp.

Here is the caller graph for this function:

```
// Call Graph for useQrencode
```

12.1.3.13  useSelection()

```cpp
void ConfigDialog::useSelection (  
    bool useSelection  )
```

ConfigDialog::useSelection set the clipboard type use from MainWindow.

Parameters

| useSelection |

Definition at line 399 of file configdialog.cpp.

Here is the caller graph for this function:

```
// Call Graph for useSelection
```
12.1.3.14 useTemplate()

void ConfigDialog::useTemplate (  
    bool useTemplate )

ConfigDialog::useTemplate set preference for using templates.

Parameters

| useTemplate |

Definition at line 853 of file configdialog.cpp.

Here is the caller graph for this function:

```
   ConfigDialog::ConfigDialog     ConfigDialog::useTemplate
```

12.1.3.15 useTrayIcon()

void ConfigDialog::useTrayIcon (  
    bool useSystray )

ConfigDialog::useTrayIcon set preference for using trayicon. Enable or disable related checkboxes accordingly.

Parameters

| useSystray |

Definition at line 675 of file configdialog.cpp.

Here is the caller graph for this function:

```
   ConfigDialog::ConfigDialog     ConfigDialog::useTrayIcon
```
12.1.3.16  \texttt{wizard()}

\texttt{void ConfigDialog::wizard ( )}

\texttt{ConfigDialog::wizard} first-time use wizard.

\texttt{Todo} make this thing more reliable.

Definition at line 576 of file configdialog.cpp.

Here is the call graph for this function:

\begin{center}
\includegraphics[width=0.8\textwidth]{call_graph.png}
\end{center}

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

- \texttt{src/configdialog.h}
- \texttt{src/configdialog.cpp}

12.2  \texttt{DeselectableTreeView} Class Reference

The \texttt{DeselectableTreeView} class loosely based on \url{http://stackoverflow.com/questions/2761284/}
thanks to Yassir Ennazk.

\texttt{#include <deselectabletreeview.h>}

Inheritance diagram for DeselectableTreeView:

\begin{center}
\includegraphics[width=0.4\textwidth]{inheritance_diagram.png}
\end{center}
Collaboration diagram for DeselectableTreeView:

```
Collaboration diagram for DeselectableTreeView:
```

**Signals**

- void **emptyClicked** ()
  
  *emptyClicked* event

**Public Member Functions**

- **DeselectableTreeView**(QWidget *parent)
  
  *DeselectableTreeView* standard constructor.

- virtual **~DeselectableTreeView**( )
  
  *~DeselectableTreeView* standard destructor

### 12.2.1 Detailed Description

The **DeselectableTreeView** class loosely based on [http://stackoverflow.com/questions/2761284/](http://stackoverflow.com/questions/2761284/) thanks to Yassir Ennazk.

Definition at line 14 of file deselectabletreeview.h.

### 12.2.2 Constructor & Destructor Documentation

#### 12.2.2.1 DeselectableTreeView()

```
DeselectableTreeView::DeselectableTreeView ( QWidget * parent ) [inline]
```

*DeselectableTreeView* standard constructor.
### Class Documentation

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
</tr>
</tbody>
</table>

Definition at line 22 of file deselectabletreeview.h.

#### ~DeselectableTreeView()

```cpp
virtual DeselectableTreeView::~DeselectableTreeView() [inline], [virtual]
```

~DeselectableTreeView standard destructor

Definition at line 26 of file deselectabletreeview.h.

#### Member Function Documentation

##### 12.2.3.1 emptyClicked

```cpp
void DeselectableTreeView::emptyClicked() [signal]
```

emptyClicked event

Here is the caller graph for this function:

![Caller Graph](null)

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

- src/deselectabletreeview.h

### 12.3 dragAndDropInfo Struct Reference

holds values to share between drag and drop on the passwordstorage view

```cpp
#include <storemodel.h>
```
12.3.1 Detailed Description

holds values to share between drag and drop on the password storage view

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

- `src/storemodel.h`

12.4 dragAndDropInfoPasswordStore Struct Reference

```cpp
#include <storemodel.h>
```

Public Attributes

- `bool isDir`
- `bool isFile`
- `QString path`

12.4.1 Detailed Description

Definition at line 46 of file storemodel.h.

12.4.2 Member Data Documentation

12.4.2.1 isDir

```cpp
bool dragAndDropInfoPasswordStore::isDir
```

Definition at line 47 of file storemodel.h.

12.4.2.2 isFile

```cpp
bool dragAndDropInfoPasswordStore::isFile
```

Definition at line 48 of file storemodel.h.
12.4.2.3 path

QString dragAndDropInfoPasswordStore::path

Definition at line 49 of file storemodel.h.

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

• src/storemodel.h

12.5 Executor Class Reference

Executes external commands for handling password, git and other data.

#include <executor.h>

Inheritance diagram for Executor:

Collaboration diagram for Executor:
12.5 Executor Class Reference

Signals

- void finished (int id, int exitCode, const QString &output, const QString &errout)
  
  finished signal that is emitted when process finishes

- void starting ()
  
  starting signal that is emitted when process starts

- void error (int id, int exitCode, const QString &output, const QString &errout)
  
  error signal that is emitted when process finishes with an error

Public Member Functions

- **Executor** (QObject ∗parent=0)
  
  Executor::Executor executes external applications.

- void execute (int id, const QString &app, const QStringList &args, bool readStdout, bool readStderr=true)
  
  Executor::execute executes an app.

- void execute (int id, const QString &workDir, const QString &app, const QStringList &args, bool readStdout, bool readStderr=true)
  
  Executor::execute executes an app from a workDir.

- void execute (int id, const QString &app, const QStringList &args, QString input=QString(), bool readStdout=false, bool readStderr=true)
  
  Executor::execute an app, takes input and presents it as stdin.

- void execute (int id, const QString &workDir, const QString &app, const QStringList &args, QString input=QString(), bool readStdout=false, bool readStderr=true)
  
  Executor::execute executes an app from a workDir, takes input and presents it as stdin.

- void setEnvironment (const QStringList &env)
  
  Executor::setEnvironment set environment variables for executor processes.

- int cancelNext ()
  
  Executor::cancelNext cancels execution of first process in queue if it's not already running.

Static Public Member Functions

- static int executeBlocking (QString app, const QStringList &args, QString input=QString(), QString ∗process_out=Q_NULLPTR, QString ∗process_err=Q_NULLPTR)
  
  Executor::executeBlocking blocking version of the executor, takes input and presents it as stdin.

- static int executeBlocking (QString app, const QStringList &args, QString ∗process_out, QString ∗process_err=Q_NULLPTR)
  
  Executor::executeBlocking blocking version of the executor.

12.5.1 Detailed Description

Executes external commands for handling password, git and other data.

Definition at line 12 of file executor.h.

12.5.2 Constructor & Destructor Documentation

12.5.2.1 Executor()

Executor::Executor (QObject ∗parent = 0 ) [explicit]

Executor::Executor executes external applications.
Parameters

parent

Definition at line 14 of file executor.cpp.

12.5.3 Member Function Documentation

12.5.3.1 cancelNext()

int Executor::cancelNext ()

Executor::cancelNext cancels execution of first process in queue if it’s not already running.

Returns

id of the cancelled process or -1 on error

Definition at line 222 of file executor.cpp.

12.5.3.2 error

void Executor::error ( int id, int exitCode, const QString & output, const QString & errout ) [signal]

error signal that is emitted when process finishes with an error

Parameters

<table>
<thead>
<tr>
<th>id</th>
<th>id of the process</th>
</tr>
</thead>
<tbody>
<tr>
<td>exitCode</td>
<td>return code of the process</td>
</tr>
<tr>
<td>output</td>
<td>stdout produced by the process</td>
</tr>
<tr>
<td>errout</td>
<td>stderr produced by the process</td>
</tr>
</tbody>
</table>

12.5.3.3 execute() [1/4]

void Executor::execute ( int id,
const QString & app,
const QStringList & args,
bool readStdout,
bool readStderr = true
)

Executor::execute execute an app.

Parameters

<table>
<thead>
<tr>
<th>id</th>
<th>app</th>
<th>args</th>
<th>readStdout</th>
<th>readStderr</th>
</tr>
</thead>
</table>

Definition at line 63 of file executor.cpp.

Here is the caller graph for this function:

12.5.3.4 execute() [2/4]

void Executor::execute {
    int id,
    const QString & app,
    const QStringList & args,
    QString input = QString(),
    bool readStdout = false,
    bool readStderr = true
}

Executor::execute an app, takes input and presents it as stdin.

Parameters

<table>
<thead>
<tr>
<th>id</th>
<th>app</th>
<th>args</th>
<th>input</th>
<th>readStdout</th>
<th>readStderr</th>
</tr>
</thead>
</table>

Generated by Doxygen
Definition at line 92 of file executor.cpp.

Here is the call graph for this function:

```
void Executor::execute (  
    int id,  
    const QString & workDir,  
    const QString & app,  
    const QStringList & args,  
    bool readStdout,  
    bool readStderr = true )
```

**Executor::execute** executes an app from a workDir.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td></td>
</tr>
<tr>
<td>workDir</td>
<td></td>
</tr>
<tr>
<td>app</td>
<td></td>
</tr>
<tr>
<td>args</td>
<td></td>
</tr>
<tr>
<td>readStdout</td>
<td></td>
</tr>
<tr>
<td>readStderr</td>
<td></td>
</tr>
</tbody>
</table>

Definition at line 77 of file executor.cpp.

Here is the call graph for this function:
12.5.3.6 execute() [4/4]

```cpp
void Executor::execute {
    int id,
    const QString & workDir,
    const QString & app,
    const QStringList & args,
    QString input = QString(),
    bool readStdout = false,
    bool readStderr = true }
```

**Executor::execute** executes an app from a workDir, takes input and presents it as stdin.

**Parameters**

- **id**
- **workDir**
- **app**
- **args**
- **input**
- **readStdout**
- **readStderr**

Definition at line 108 of file executor.cpp.

12.5.3.7 executeBlocking() [1/2]

```cpp
int Executor::executeBlocking {
    QString app,
    const QStringList & args,
    QString * process_out,
    QString * process_err = Q_NULLPTR ) [static]
```

**Executor::executeBlocking** blocking version of the executor.

**Parameters**

- **app**
- **args**
- **process_out**
- **process_err**

**Returns**

Definition at line 202 of file executor.cpp.
Here is the call graph for this function:

```
Executor::executeBlocking
```

12.5.3.8 executeBlocking() [2/2]

```cpp
int Executor::executeBlocking (  
    QString app,  
    const QStringList & args,  
    QString input = QString(),  
    QString * process_out = Q_NULLPTR,  
    QString * process_err = Q_NULLPTR ) [static]
```

**Executor::executeBlocking** blocking version of the executor, takes input and presents it as stdin.

**Parameters**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td></td>
</tr>
<tr>
<td>args</td>
<td></td>
</tr>
<tr>
<td>input</td>
<td></td>
</tr>
<tr>
<td>process_out</td>
<td></td>
</tr>
<tr>
<td>process_err</td>
<td></td>
</tr>
</tbody>
</table>

**Returns**

TODO(bezet): it might make sense to throw here, a lot of possible errors

Definition at line 160 of file executor.cpp.

Here is the caller graph for this function:
12.5.9 finished

```cpp
void Executor::finished ( int id,
                        int exitCode,
                        const QString & output,
                        const QString & errout ) [signal]
```

finished signal that is emitted when process finishes

**Parameters**

<table>
<thead>
<tr>
<th>id</th>
<th>id of the process</th>
</tr>
</thead>
<tbody>
<tr>
<td>exitCode</td>
<td>return code of the process</td>
</tr>
<tr>
<td>output</td>
<td>stdout produced by the process</td>
</tr>
<tr>
<td>errout</td>
<td>stderr produced by the process</td>
</tr>
</tbody>
</table>

12.5.10 setEnvironment()

```cpp
void Executor::setEnvironment ( const QStringList & env )
```

Executor::setEnvironment set environment variables for executor processes.

**Parameters**

<table>
<thead>
<tr>
<th>env</th>
<th></th>
</tr>
</thead>
</table>

Definition at line 212 of file executor.cpp.

Here is the caller graph for this function:

![Caller Graph](image)

12.5.11 starting

```cpp
void Executor::starting ( ) [signal]
```

starting signal that is emitted when process starts
Here is the caller graph for this function:

```
Executor::Executor
    
Pass::Pass
```

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

- `src/executor.h`
- `src/executor.cpp`

### 12.6 FileContent Class Reference

```cpp
#include <filecontent.h>
```

#### Public Member Functions

- `QString getPassword () const`
- `NamedValues getNamedValues () const`
- `QString getRemainingData () const`
- `QString getRemainingDataForDisplay () const`

#### Static Public Member Functions

- `static FileContent parse (const QString &fileContent, const QStringList &templateFields, bool allFields)`

  `parse` parses the given `fileContent` in a `FileContent` object. The password is accessible through `getPassword`. The named value pairs (name: value) are parsed and depending on the `templateFields` and `allFields` parameters accessible through `getNamedValues`, `getRemainingData` or `getRemainingDataForDisplay`.

### 12.6.1 Detailed Description

Definition at line 25 of file `filecontent.h`.

### 12.6.2 Member Function Documentation
12.6.2.1 getNamedValues()

**NamedValues** FileContent::getNamedValues ( ) const

Returns

the named values in the file in the order of appearance, with template values first.

Definition at line 38 of file filecontent.cpp.

Here is the caller graph for this function:

12.6.2.2 getPassword()

**QString** FileContent::getPassword ( ) const

Returns

the password from the parsed file.

Definition at line 36 of file filecontent.cpp.

Here is the caller graph for this function:
12.6.2.3 getRemainingData()

QString FileContent::getRemainingData ( ) const

Returns

the data that is not the password and not in getNamedValues.

Definition at line 40 of file filecontent.cpp.

Here is the caller graph for this function:

```
PasswordDialog::Password
PasswordDialog::setPass
PasswordDialog::setPassword
FileContent::getRemainingData
```

12.6.2.4 getRemainingDataForDisplay()

QString FileContent::getRemainingDataForDisplay ( ) const

@like getRemainingData but without data that should not be displayed (like a TOTP secret).

Definition at line 42 of file filecontent.cpp.

Here is the caller graph for this function:

```
MainWindow::passShowHandler
FileContent::getRemainingDataForDisplay
```

12.6.2.5 parse()

FileContent FileContent::parse (const QString & fileContent,
                  const QStringList & templateFields,
                  bool allFields) [static]

parse parses the given fileContent in a FileContent object. The password is accessible through getPassword. The named value pairs (name: value) are parsed and depending on the templateFields and allFields parameters accessible through getNamedValues, getRemainingData or getRemainingDataForDisplay.
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileContent</td>
<td>the file content to parse.</td>
</tr>
<tr>
<td>templateFields</td>
<td>the fields in the template. Fields in the template will always be in getNamedValues() at the beginning of the list in the same order.</td>
</tr>
<tr>
<td>allFields</td>
<td>whether all fields should be considered as named values. If set to false only templateFields are returned in getNamedValues().</td>
</tr>
</tbody>
</table>

Returns

Definition at line 7 of file filecontent.cpp.

Here is the caller graph for this function:

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

- src/filecontent.h
- src/filecontent.cpp

12.7 ImitatePass Class Reference

Imitates pass features when pass is not enabled or available.

#include <imitatepass.h>

Inheritance diagram for ImitatePass:
Collaboration diagram for ImitatePass:

Signals

- void startReencryptPath ()
- void endReencryptPath ()

Public Member Functions

- **ImitatePass ()**
  
  *ImitatePass::ImitatePass* for situations when pass is not available we imitate the behavior of pass [https://www.passwordstore.org/](https://www.passwordstore.org/).

- virtual **~ImitatePass ()**

- virtual void **GitInit ()** Q_DECL_OVERRIDE
  
  *ImitatePass::GitInit* git init wrapper.

- virtual void **GitPull ()** Q_DECL_OVERRIDE
  
  *ImitatePass::GitPull* git init wrapper.

- virtual void **GitPull_b ()** Q_DECL_OVERRIDE
  
  *ImitatePass::GitPull_b* git pull wrapper.

- virtual void **GitPush ()** Q_DECL_OVERRIDE
  
  *ImitatePass::GitPush* git init wrapper.

- virtual void **Show (QString file)** Q_DECL_OVERRIDE
  
  *ImitatePass::Show* shows content of file.

- virtual void **OtpGenerate (QString file)** Q_DECL_OVERRIDE
  
  *ImitatePass::OtpGenerate* generates an otp code.

- virtual void **Insert (QString file, QString newValue, bool overwrite=false)** Q_DECL_OVERRIDE
  
  *ImitatePass::Insert* create new file with encrypted content.

- virtual void **Remove (QString file, bool isDir=false)** Q_DECL_OVERRIDE

Generated by Doxygen
ImitatePass::Remove custom implementation of "pass remove".

- virtual void Init (QString path, const QList<UserInfo>& users) Q_DECL_OVERRIDE
  
  ImitatePass::Init initialize pass repository.
- void reencryptPath (const QString &dir)
  
  ImitatePass::reencryptPath reencrypt all files under the chosen directory.
- void Move (const QString src, const QString dest, const bool force=false) Q_DECL_OVERRIDE
- void Copy (const QString src, const QString dest, const bool force=false) Q_DECL_OVERRIDE

Protected Member Functions

- virtual void finished (int id, int exitCode, const QString &out, const QString &err) Q_DECL_OVERRIDE
  
  ImitatePass::finished this function is overloaded to ensure identical behaviour to RealPass ie. only PASS_* processes are visible inside Pass::finish, so that interface-wise it all looks the same.
- virtual void executeWrapper (PROCESS id, const QString &app, const QStringList &args, QString input, bool readStdout=true, bool readStderr=true) Q_DECL_OVERRIDE
  
  executeWrapper overridden so that every execution is a transaction

Additional Inherited Members

12.7.1 Detailed Description

Imitates pass features when pass is not enabled or available.

Definition at line 11 of file imitatepass.h.

12.7.2 Constructor & Destructor Documentation

12.7.2.1 ImitatePass()

ImitatePass::ImitatePass ( ) [default]

ImitatePass::ImitatePass for situations when pass is not available we imitate the behavior of pass  
https://www.passwordstore.org/.

12.7.2.2 ~ImitatePass()

virtual ImitatePass::~ImitatePass ( ) [inline], [virtual]

Definition at line 48 of file imitatepass.h.

12.7.3 Member Function Documentation

Generated by Doxygen
12.7.3.1 Copy()

```cpp
void ImitatePass::Copy (const QString src,
                        const QString dest,
                        const bool force = false) [virtual]
```

Implements Pass.

Definition at line 381 of file imitatepass.cpp.

Here is the call graph for this function:

![Call Graph for Copy Function](image)

12.7.3.2 endReencryptPath

```cpp
void ImitatePass::endReencryptPath () [signal]
```

12.7.3.3 executeWrapper()

```cpp
void ImitatePass::executeWrapper (PROCESS id,
                                   const QString & app,
                                   const QStringList & args,
                                   QString input,
                                   bool readStdout = true,
                                   bool readStderr = true) [protected], [virtual]
```

executeWrapper overrided so that every execution is a transaction

Parameters

<table>
<thead>
<tr>
<th>id</th>
<th>app</th>
<th>args</th>
<th>input</th>
<th>readStdout</th>
<th>readStderr</th>
</tr>
</thead>
</table>

Reimplemented from Pass.
12.7 ImitatePass Class Reference

Definition at line 480 of file imitatepass.cpp.

Here is the call graph for this function:

```
ImitatePass::executeWrapper -> Pass::executeWrapper
```

### 12.7.3.4 finished()

```cpp
defined ImitatePass::finished (  
    int id,  
    int exitCode,  
    const QString & out,  
    const QString & err ) [protected], [virtual]
```

*ImitatePass::finished* this function is overloaded to ensure identical behaviour to *RealPass* ie. only PASS_* processes are visible inside Pass::finish, so that interface-wise it all looks the same.

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td></td>
</tr>
<tr>
<td>exitCode</td>
<td></td>
</tr>
<tr>
<td>out</td>
<td></td>
</tr>
<tr>
<td>err</td>
<td></td>
</tr>
</tbody>
</table>

Reimplemented from *Pass*.

Definition at line 442 of file imitatepass.cpp.

Here is the call graph for this function:

```
ImitatePass::finished -> Pass::finished
```
12.7.3.5 GitInit()

```cpp
void ImitatePass::GitInit() [virtual]
```

ImitatePass::GitInit git init wrapper.

Implements Pass.

Definition at line 35 of file imitatepass.cpp.

12.7.3.6 GitPull()

```cpp
void ImitatePass::GitPull() [virtual]
```

ImitatePass::GitPull git pull wrapper.

Implements Pass.

Definition at line 42 of file imitatepass.cpp.

12.7.3.7 GitPull_b()

```cpp
void ImitatePass::GitPull_b() [virtual]
```

ImitatePass::GitPull_b git pull wrapper.

Implements Pass.

Definition at line 47 of file imitatepass.cpp.

Here is the call graph for this function:

```
ImitatePass::GitPull_b   QtPassSettings::getGitExecutable   QtPassSettings::getInstance
```

Generated by Doxygen
12.7.3.8 GitPush()

void ImitatePass::GitPush () [virtual]

*ImitatePass::GitPush* git init wrapper.

Implements *Pass*.

Definition at line 54 of file imitatepass.cpp.

Here is the call graph for this function:

![Call Graph for GitPush()](call_graph.png)

12.7.3.9 Init()

void ImitatePass::Init ( 
    QString path, 
    const QList< UserInfo > & users ) [virtual]

*ImitatePass::Init* initialize pass repository.

Parameters

<table>
<thead>
<tr>
<th>path</th>
<th>path in which new password-store will be created</th>
</tr>
</thead>
<tbody>
<tr>
<td>users</td>
<td>list of users who shall be able to decrypt passwords in path</td>
</tr>
</tbody>
</table>

Implements *Pass*.

Definition at line 164 of file imitatepass.cpp.

Here is the call graph for this function:

![Call Graph for Init()](call_graph.png)
### 12.7.3.10 Insert()

```cpp
void ImitatePass::Insert (QString file, QString newValue, bool overwrite = false) [virtual]
```

*ImitatePass::Insert* create new file with encrypted content.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>file</code></td>
<td>file to be created</td>
</tr>
<tr>
<td><code>newValue</code></td>
<td>value to be stored in file</td>
</tr>
<tr>
<td><code>overwrite</code></td>
<td>whether to overwrite existing file</td>
</tr>
</tbody>
</table>

Implements *Pass*.

Definition at line 89 of file imitatepass.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

### 12.7.3.11 Move()

```cpp
void ImitatePass::Move (const QString src, const QString dest, const bool force = false) [virtual]
```

Implements *Pass*.

Definition at line 343 of file imitatepass.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)
12.7.3.12 OtpGenerate()

void ImitatePass::OtpGenerate (QString file) [virtual]

ImitatePass::OtpGenerate generates an otp code.

Implements Pass.

Definition at line 73 of file imitatepass.cpp.

12.7.3.13 reencryptPath()

void ImitatePass::reencryptPath (const QString &dir)

ImitatePass::reencryptPath reencrypt all files under the chosen directory.

This is still quite experimental.

Parameters

| dir |

Definition at line 240 of file imitatepass.cpp.

Here is the call graph for this function:

12.7.3.14 Remove()

void ImitatePass::Remove (QString file, bool isDir = false) [virtual]

ImitatePass::Remove custom implementation of "pass remove".

Implements Pass.

Definition at line 134 of file imitatepass.cpp.

---

Generated by Doxygen
Here is the call graph for this function:

12.7.3.15 Show()

```cpp
void ImitatePass::Show ( QString file ) [virtual]
```

ImitatePass::Show shows content of file.

Implements Pass.

Definition at line 63 of file imitatepass.cpp.

Here is the call graph for this function:

12.7.3.16 startReencryptPath

```cpp
void ImitatePass::startReencryptPath ( ) [signal]
```

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

- src/imitatepass.h
- src/imitatepass.cpp
12.8 KeygenDialog Class Reference

Handles GPG keypair generation.

#include <keygendialog.h>

Inheritance diagram for KeygenDialog:

![Inheritance Diagram](image1.jpg)

Collaboration diagram for KeygenDialog:

![Collaboration Diagram](image2.jpg)

Public Member Functions

- KeygenDialog (ConfigDialog *parent=0)
  
  KeygenDialog::KeygenDialog basic constructor.
- ~KeygenDialog ()
  
  KeygenDialog::~KeygenDialog even more basic destructor.

Protected Member Functions

- void closeEvent (QCloseEvent *event)
  
  KeygenDialog::closeEvent we are done here.
12.8.1 Detailed Description

Handles GPG keypair generation.

Definition at line 16 of file keygendialog.h.

12.8.2 Constructor & Destructor Documentation

12.8.2.1 KeygenDialog()

KeygenDialog::KeygenDialog (  
    ConfigDialog * parent = 0 ) [explicit]

KeygenDialog::KeygenDialog basic constructor.

Parameters

- **parent**

Definition at line 15 of file keygendialog.cpp.

12.8.2.2 ~KeygenDialog()

KeygenDialog::~KeygenDialog ( )

KeygenDialog::~KeygenDialog even more basic destructor.

Definition at line 24 of file keygendialog.cpp.

12.8.3 Member Function Documentation

12.8.3.1 closeEvent()

void KeygenDialog::closeEvent (  
    QCloseEvent * event ) [protected]

KeygenDialog::closeEvent we are done here.
Parameters

Definition at line 184 of file keygendialog.cpp.

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

- src/keygendialog.h
- src/keygendialog.cpp

12.9 MainWindow Class Reference

The MainWindow class does way too much, not only is it a switchboard, configuration handler and more, it's also
the process-manager.

#include <mainwindow.h>

Inheritance diagram for MainWindow:

```
 QMainWindow
  ^
  |  
MainWindow
```

Collaboration diagram for MainWindow:

```
 QMainWindow
  ^
  |  
MainWindow
```
Public Slots

- **void deselect ()**
  
  `MainWindow::deselect` clear the selection, password and copy buffer.

- **void messageAvailable (QString message)**
  
  `MainWindow::messageAvailable` we have some text/message/search to do.

- **void critical (QString, QString)**
  
  `MainWindow::critical` critical message popup wrapper.

- **void executeWrapperStarted ()**

- **void showStatusMessage (QString msg, int timeout=2000)**
  
  Displays message in status bar.

- **void passShowHandler (const QString &)**

- **void passOtpHandler (const QString &)**

- **void onPush ()**
  
  `MainWindow::onPush` do a git push.

- **void on_treeView_clicked (const QModelIndex &index)**
  
  `MainWindow::on_treeView_clicked` read the selected password file.

- **void startReencryptPath ()**
  
  `MainWindow::startReencryptPath` disable ui elements and treeview.

- **void endReencryptPath ()**
  
  `MainWindow::endReencryptPath` re-enable ui elements.

Signals

- **void passShowHandlerFinished (QString output)**

- **void passGitInitNeeded ()**

- **void generateGPGKeyPair (QString batch)**

Public Member Functions

- **MainWindow (const QString &searchText=QString(), QWidget ∗parent=nullptr)**
  
  `MainWindow::MainWindow` handles all of the main functionality and also the main window.

- **~MainWindow ()**

- **void restoreWindow ()**

- **void generateKeyPair (QString, QDialog ∗)**
  
  `MainWindow::generateKeyPair` internal gpg keypair generator...

- **void userDialog (QString="")**
  
  `MainWindow::userDialog` see `MainWindow::onUsers()`

- **void config ()**
  
  `MainWindow::config` pops up the configuration screen and handles all inter-window communication.

- **void setUiElementsEnabled (bool state)**
  
  `MainWindow::setUiElementsEnabled` enable or disable the relevant UI elements.

- **void flashText (const QString &text, const bool isError, const bool isHtml=false)**

- **const QModelIndex getCurrentTreeViewIndex ()**

- **QDialog ∗getKeygenDialog ()**

- **void cleanKeygenDialog ()**
Protected Member Functions

- void closeEvent (QCloseEvent *event)
  
  MainWindow::closeEvent hide or quit.
- void keyPressEvent (QKeyEvent *event)
  
  MainWindow::keyPressEvent did anyone press return, enter or escape?
- void changeEvent (QEvent *event)
  
  MainWindow::changeEvent sets focus to the search box.
- bool eventFilter (QObject *obj, QEvent *event)
  
  MainWindow::eventFilter filter out some events and focus the treeview.

12.9.1 Detailed Description

The MainWindow class does way too much, not only is it a switchboard, configuration handler and more, it's also the process-manager.

This class could really do with an overhaul.

Definition at line 39 of file mainwindow.h.

12.9.2 Constructor & Destructor Documentation

12.9.2.1 MainWindow()

MainWindow::MainWindow {
    const QString & searchText = QString(),
    QWidget * parent = nullptr ) [explicit]

MainWindow::MainWindow handles all of the main functionality and also the main window.

Parameters

<table>
<thead>
<tr>
<th>searchText</th>
<th>for searching from cli</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>pointer</td>
</tr>
</tbody>
</table>

Definition at line 38 of file mainwindow.cpp.
Here is the call graph for this function:

![Call Graph Image]

12.9.2.2  ~MainWindow()

MainWindow::~MainWindow ( )

Definition at line 127 of file mainwindow.cpp.

12.9.3  Member Function Documentation

12.9.3.1  changeEvent()

void MainWindow::changeEvent ( 
    QEvent * event ) [protected]

MainWindow::changeEvent sets focus to the search box.

Parameters

<table>
<thead>
<tr>
<th>event</th>
<th></th>
</tr>
</thead>
</table>

Generated by Doxygen
12.9 MainWindow Class Reference

Definition at line 144 of file mainwindow.cpp.

### 12.9.3.2 cleanKeygenDialog()

```cpp
void MainWindow::cleanKeygenDialog() {
}
```

Definition at line 204 of file mainwindow.cpp.

### 12.9.3.3 closeEvent()

```cpp
void MainWindow::closeEvent (QCloseEvent * event) [protected]
```

**MainWindow::closeEvent** hide or quit.

**Parameters**

- `event`

Definition at line 820 of file mainwindow.cpp.

Here is the call graph for this function:

![call graph diagram]

### 12.9.3.4 config()

```cpp
void MainWindow::config() {
}
```
MainWindow::config pops up the configuration screen and handles all inter-window communication.

Definition at line 229 of file mainwindow.cpp.

Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

12.9.3.5 critical

void MainWindow::critical (  
    QString title,  
    QString msg ) [slot]

MainWindow::critical critical message popup wrapper.

Parameters

<table>
<thead>
<tr>
<th>title</th>
<th>msg</th>
</tr>
</thead>
</table>

Definition at line 1195 of file mainwindow.cpp.
12.9.3.6 deselect

void MainWindow::deselect () [slot]

MainWindow::deselect clear the selection, password and copy buffer.

Definition at line 352 of filemainwindow.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.9.3.7 endReencryptPath

void MainWindow::endReencryptPath () [slot]

MainWindow::endReencryptPath re-enable ui elements.

Definition at line 1162 of filemainwindow.cpp.

Here is the call graph for this function:
12.9.3.8 eventFilter()

```cpp
bool MainWindow::eventFilter ( 
    QObject ∗ obj, 
    QEvent ∗ event ) [protected]
```

MainWindow::eventFilter filter out some events and focus the treeview.

**Parameters**

- `obj`
- `event`

**Returns**

Definition at line 845 of file mainwindow.cpp.

12.9.3.9 executeWrapperStarted

```cpp
void MainWindow::executeWrapperStarted () [slot]
```

Definition at line 364 of file mainwindow.cpp.

Here is the call graph for this function:

```
MainWindow::executeWrapper Started
MainWindow::setUiElements Enabled
```

12.9.3.10 flashText()

```cpp
void MainWindow::flashText ( 
    const QString & text, 
    const bool isError, 
    const bool isHtml = false )
```

Definition at line 209 of file mainwindow.cpp.
12.9.3.11 generateGPGKeyPair

```cpp
void MainWindow::generateGPGKeyPair ( QString batch ) [signal]
```

Here is the caller graph for this function:

```
ConfigDialog::genKey  MainWindow::generateKeyPair  MainWindow::generateGPGKeyPair
```

12.9.3.12 generateKeyPair()

```cpp
void MainWindow::generateKeyPair ( QString batch, QDialog *keygenWindow )
```

`MainWindow::generateKeyPair` internal gpg keypair generator . .

Parameters

<table>
<thead>
<tr>
<th>batch</th>
</tr>
</thead>
<tbody>
<tr>
<td>keygenWindow</td>
</tr>
</tbody>
</table>

Definition at line 731 of filemainwindow.cpp.

Here is the caller graph for this function:

```
ConfigDialog::genKey  MainWindow::generateKeyPair
```

12.9.3.13 getCurrentTreeViewIndex()

```cpp
const QModelIndex MainWindow::getCurrentTreeViewIndex ( )
```

Definition at line 200 of filemainwindow.cpp.
12.9.3.14 getKeygenDialog()

QDialog* MainWindow::getKeygenDialog () [inline]

Definition at line 58 of file mainwindow.h.

12.9.3.15 keyPressEvent()

void MainWindow::keyPressEvent ( 
    QKeyEvent * event ) [protected]

MainWindow::keyPressEvent did anyone press return, enter or escape?

Parameters

- **event**

Definition at line 859 of file mainwindow.cpp.

Here is the call graph for this function:

12.9.3.16 messageAvailable

void MainWindow::messageAvailable ( 
    QString message ) [slot]

MainWindow::messageAvailable we have some text/message/search to do.

Parameters

- **message**

Definition at line 714 of file mainwindow.cpp.
Here is the caller graph for this function:

![Caller Graph for MainWindow::messageAvailable](image)

### 12.9.3.17 on_treeView_clicked

```cpp
void MainWindow::on_treeView_clicked ( const QModelIndex & index ) [slot]
```

**MainWindow::on_treeView_clicked** read the selected password file.

**Parameters**

- **index**

Definition at line 318 of filemainwindow.cpp.

Here is the call graph for this function:

![Call Graph for on_treeView_clicked](image)

Here is the caller graph for this function:

![Caller Graph for MainWindow::on_treeView_clicked](image)
12.9.3.18 onPush

```cpp
void MainWindow::onPush ( ) [slot]
```

**MainWindow::onPush**
do a git push.

Definition at line 288 of file mainwindow.cpp.

Here is the call graph for this function:

![Call graph for onPush function](image)

12.9.3.19 passGitInitNeeded

```cpp
void MainWindow::passGitInitNeeded ( ) [signal]
```

**MainWindow::passGitInitNeeded**

Here is the caller graph for this function:

![Caller graph for passGitInitNeeded function](image)

12.9.3.20 passOtpHandler

```cpp
void MainWindow::passOtpHandler ( const QString & p_output ) [slot]
```

**MainWindow::passOtpHandler**

Definition at line 417 of file mainwindow.cpp.

Here is the call graph for this function:

![Call graph for passOtpHandler function](image)
12.9.3.21  passShowHandler

```cpp
void MainWindow::passShowHandler (  
    const QString & p_output ) [slot]
```

Definition at line 371 of file mainwindow.cpp.

Here is the call graph for this function:

```
Mainwindow::passShowHandler
FileContent::getNamedValues
QtPassSettings::getPassTemplate
FileContent::getPassword
FileContent::getRemaining
DataForDisplay
QtPassSettings::isHideContent
QtPassSettings::isTemplate
AllFields
QtPassSettings::isUseAutoclear
Panel
QtPassSettings::isUseTemplate
FileContent::parse
QtPass::setClippedText
MainWindow::setUiElements
Enabled
QtPassSettings::getInstance
QtPass::copyTextToClipboard
QtPassSettings::getClipBoardType
QtPassSettings::isUseAutoclear
QtPassSettings::isUseSelection
MainWindow::showStatusMessage
QtPassSettings::getClipBoard
TypeRaw
```

12.9.3.22  passShowHandlerFinished

```cpp
void MainWindow::passShowHandlerFinished (  
    QString output ) [signal]
```

Here is the caller graph for this function:

```
MainWindow::passShowHandler
Finished
MainWindow::passShowHandler
FinishedMainWindow::passShowHandler
```

Generated by Doxygen
### 12.9.3.23 restoreWindow()

```cpp
tvoid MainWindow::restoreWindow()
```

Definition at line 466 of filemainwindow.cpp.

Here is the call graph for this function:

![Call Graph for restoreWindow()](image.png)

### 12.9.3.24 setUiElementsEnabled()

```cpp
tvoid MainWindow::setUiElementsEnabled(
    bool state
)
```

MainWindow::setUiElementsEnabled enable or disable the relevant UI elements.

Parameters

- **state**

Definition at line 450 of filemainwindow.cpp.
Here is the caller graph for this function:

```
loopScan: endReencryptPath
MainWindow::executeWrapper
MainWindow::MainWindow
MainWindow::passOtpHandler
MainWindow::passShowHandler
MainWindow::startReencryptPath
```

### 12.9.3.25 showStatusMessage

```cpp
def MainWindow::showStatusMessage (
    const QString &msg,
    int timeout = 2000 ) [slot]
```

Displays message in status bar.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>msg</td>
<td>text to be displayed</td>
</tr>
<tr>
<td>timeout</td>
<td>time for which msg shall be visible</td>
</tr>
</tbody>
</table>

Definition at line 1147 of file mainwindow.cpp.

Here is the caller graph for this function:
12.9.3.26 startReencryptPath

void MainWindow::startReencryptPath ( ) [slot]

MainWindow::startReencryptPath disable ui elements and treeview.
Definition at line 1154 of filemainwindow.cpp.

Here is the call graph for this function:

```
MainWindow::startReencryptPath  MainWindow::setUiElements Enabled
```

12.9.3.27 userDialog()

void MainWindow::userDialog ( 
    QString dir = ""
)

MainWindow::userDialog see MainWindow::onUsers()

Parameters

- **dir** folder to edit users for.

Definition at line 687 of filemainwindow.cpp.

Here is the caller graph for this function:

```
ConfigDialog::wizard  MainWindow::userDialog
```

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

- src/mainwindow.h
- src/mainwindow.cpp
#include <filecontent.h>

## Public Attributes

- QString name
- QString value

### 12.10.1 Detailed Description

Definition at line 8 of file filecontent.h.

### 12.10.2 Member Data Documentation

#### 12.10.2.1 name

QString NamedValue::name

Definition at line 9 of file filecontent.h.

#### 12.10.2.2 value

QString NamedValue::value

Definition at line 10 of file filecontent.h.

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

- src/filecontent.h

Generated by Doxygen
12.11 NamedValues Class Reference

The NamedValues class is mostly a list of NamedValue but also has a method to take a specific NamedValue pair out of the list.

```cpp
#include <filecontent.h>
```

Inheritance diagram for NamedValues:

![Inheritance Diagram](image)

Collaboration diagram for NamedValues:

![Collaboration Diagram](image)

Public Member Functions

- NamedValues ()
- NamedValues (std::initializer_list<NamedValue> values)
- QString takeValue (const QString &name)

12.11.1 Detailed Description

The NamedValues class is mostly a list of NamedValue but also has a method to take a specific NamedValue pair out of the list.

Definition at line 17 of file filecontent.h.
12.11 NamedValues Class Reference

12.11.2 Constructor & Destructor Documentation

12.11.2.1 NamedValues() [1/2]

NamedValues::NamedValues ( )

Definition at line 54 of file filecontent.cpp.

12.11.2.2 NamedValues() [2/2]

NamedValues::NamedValues ( std::initializer_list< NamedValue > values )

Definition at line 56 of file filecontent.cpp.

12.11.3 Member Function Documentation

12.11.3.1 takeValue()

QString NamedValues::takeValue ( const QString & name )

Definition at line 59 of file filecontent.cpp.

Here is the caller graph for this function:

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

- src/filecontent.h
- src/filecontent.cpp

Generated by Doxygen
12.12 Pass Class Reference

Acts as an abstraction for pass or pass imitation.

`#include <pass.h>`

Inheritance diagram for Pass:

![Inheritance Diagram for Pass]

Collaboration diagram for Pass:

![Collaboration Diagram for Pass]

**Signals**

- void `error` (QProcess::ProcessError)
- void `startingExecuteWrapper` ()
- void `statusMsg` (QString, int)
• void **critical** (QString, QString)
• void **processErrorExit** (int exitCode, const QString &err)
• void **finishedAny** (const QString &
const QString &)
• void **finishedGitInit** (const QString &
const QString &)
• void **finishedGitPull** (const QString &
const QString &)
• void **finishedGitPush** (const QString &
const QString &)
• void **finishedShow** (const QString &)
• void **finishedOtpGenerate** (const QString &)
• void **finishedInsert** (const QString &
const QString &)
• void **finishedRemove** (const QString &
const QString &)
• void **finishedInit** (const QString &
const QString &)
• void **finishedMove** (const QString &
const QString &)
• void **finishedCopy** (const QString &
const QString &)
• void **finishedGenerate** (const QString &
const QString &)
• void **finishedGenerateGPGKeys** (const QString &
const QString &)

**Public Member Functions**

• **Pass ()**
  
  *Pass::Pass* wrapper for using either pass or the pass imitation.
• void **init ()**
• virtual **~Pass ()**
• virtual void **GitInit ()=0**
• virtual void **GitPull ()=0**
• virtual void **GitPull_b ()=0**
• virtual void **GitPush ()=0**
• virtual void **Show (QString file)=0**
• virtual void **OtpGenerate (QString file)=0**
• virtual void **Insert (QString file, QString value, bool force)=0**
• virtual void **Remove (QString file, bool isDir)=0**
• virtual void **Move (const QString srcDir, const QString dest, const bool force=false)=0**
• virtual void **Copy (const QString srcDir, const QString dest, const bool force=false)=0**
• virtual void **Init (QString path, const QList <UserInfo> &users)=0**
  
  *Pass::Generate* use either pwgen or internal password generator.
• void **GenerateGPGKeys (QString batch)**
  
  *Pass::GenerateGPGKeys* internal gpg keypair generator . .
• QList <UserInfo> **listKeys (QStringList keystings, bool secret=false)**
  
  *Pass::listKeys* list users.
• QList <UserInfo> **listKeys (QString keystring="", bool secret=false)**
  
  *Pass::listKeys* list users.
• void **updateEnv ()**
  
  *Pass::updateEnv* update the execution environment (used when switching profiles)

**Static Public Member Functions**

• static QStringList **getRecipientList (QString for_file)**
  
  *Pass::getRecipientList* return list of gpg-id's to encrypt for.
• static QStringList **getRecipientString (QString for_file, QString separator=" ", int *count=NULL)**
  
  *Pass::getRecipientString* formated string for use with GPG.
Protected Types

- typedef Enums::PROCESS PROCESS

Protected Slots

- virtual void finished (int id, int exitCode, const QString &out, const QString &err)
  
  Pass::processFinished reemits specific signal based on what process has finished.

Protected Member Functions

- void executeWrapper (PROCESS id, const QString &app, const QStringList &args, bool readStdout=true, bool readStderr=true)
- QString generateRandomPassword (const QString &charset, unsigned int length)
- quint32 boundedRandom (quint32 bound)
- virtual void executeWrapper (PROCESS id, const QString &app, const QStringList &args, QString input, bool readStdout=true, bool readStderr=true)

Protected Attributes

- Executor exec

12.12.1 Detailed Description

Acts as an abstraction for pass or pass imitation.

Definition at line 25 of file pass.h.

12.12.2 Member Typedef Documentation

12.12.2.1 PROCESS

typedef Enums::PROCESS Pass::PROCESS [protected]

Definition at line 34 of file pass.h.

12.12.3 Constructor & Destructor Documentation

Generated by Doxygen
12.12.3.1 Pass()

Pass::Pass ( )

Pass::Pass wrapper for using either pass or the pass imitation.

Definition at line 15 of file pass.cpp.

Here is the call graph for this function:

![Call Graph for Pass::Pass](image)

12.12.3.2 ~Pass()

virtual Pass::~Pass ( ) [inline], [virtual]

Definition at line 40 of file pass.h.

12.12.4 Member Function Documentation

12.12.4.1 boundedRandom()

quint32 Pass::boundedRandom ( 
    quint32 bound ) [protected]

Definition at line 309 of file pass.cpp.

Here is the caller graph for this function:

![Caller Graph for Pass::boundedRandom](image)
12.12.4.2 Copy()

virtual void Pass::Copy {
    const QString srcDir,
    const QString dest,
    const bool force = false } [pure virtual]

Implemented in ImitatePass, and RealPass.

Here is the caller graph for this function:

```
StoreModel::dropMimeData -> Pass::Copy
```

12.12.4.3 critical

void Pass::critical {
    QString ,
    QString } [signal]

Here is the caller graph for this function:

```
Pass::Generate_b -> Pass::critical
```

12.12.4.4 error

void Pass::error {
    QProcess::ProcessError } [signal]
12.12.4.5 `executeWrapper()` [1/2]

```cpp
void Pass::executeWrapper (  
    PROCESS id,  
    const QString & app,  
    const QStringList & args,  
    bool readStdout = true,  
    bool readStderr = true ) [protected]
```

Definition at line 29 of file pass.cpp.

Here is the caller graph for this function:

```
ImitatePass::executeWrapper
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass::executeWrapper</td>
<td></td>
</tr>
<tr>
<td>Pass::GenerateGPGKeys</td>
<td></td>
</tr>
</tbody>
</table>
```

12.12.4.6 `executeWrapper()` [2/2]

```cpp
void Pass::executeWrapper (  
    PROCESS id,  
    const QString & app,  
    const QStringList & args,  
    QString input,  
    bool readStdout = true,  
    bool readStderr = true ) [protected], [virtual]
```

Reimplemented in `ImitatePass`.

Definition at line 35 of file pass.cpp.

Here is the call graph for this function:

```
Pass::executeWrapper
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Executor::execute</td>
<td></td>
</tr>
<tr>
<td>QtPassSettings::getPassStore</td>
<td>Pass::executeWrapper</td>
</tr>
<tr>
<td>QtPassSettings::getInstance</td>
<td>QtPassSettings::getInstance</td>
</tr>
</tbody>
</table>
```

Generated by Doxygen
12.12.4.7 finished

```cpp
void Pass::finished (  
    int id,             
    int exitCode,      
    const QString & out, 
    const QString & err ) [protected], [virtual], [slot]
```

Pass::processFinished reemits specific signal based on what process has finished.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>id of Pass process that was scheduled and finished</td>
</tr>
<tr>
<td>exitCode</td>
<td>return code of a process</td>
</tr>
<tr>
<td>out</td>
<td>output generated by process(if capturing was requested, empty otherwise)</td>
</tr>
<tr>
<td>err</td>
<td>error output generated by process(if capturing was requested, or error occurred)</td>
</tr>
</tbody>
</table>

Reimplemented in ImitatePass.

Definition at line 192 of file pass.cpp.

Here is the caller graph for this function:

```
ImitatePass::finished
    Pass::finished
    Pass::Pass
```

12.12.4.8 finishedAny

```cpp
void Pass::finishedAny (  
    const QString & out,  
    const QString & err ) [signal]
```

Generated by Doxygen
12.12.4.9 finishedCopy

```cpp
void Pass::finishedCopy ( 
    const QString & , 
    const QString & ) [signal]
```

Here is the caller graph for this function:

![Caller Graph](image-url)

12.12.4.10 finishedGenerate

```cpp
void Pass::finishedGenerate ( 
    const QString & , 
    const QString & ) [signal]
```

12.12.4.11 finishedGenerateGPGKeys

```cpp
void Pass::finishedGenerateGPGKeys ( 
    const QString & , 
    const QString & ) [signal]
```

12.12.4.12 finishedGitInit

```cpp
void Pass::finishedGitInit ( 
    const QString & , 
    const QString & ) [signal]
```

Here is the caller graph for this function:

![Caller Graph](image-url)
12.12.4.13 finishedGitPull

```c
class Pass {
public:
    // Constructor
    Pass();

    // Destructor
    ~Pass();

    // Signal
    void finishedGitPull (const QString &, const QString &); // [signal]

private:
    // Member variables
    QString m_string1;
    QString m_string2;
};
```

Here is the caller graph for this function:

![Caller Graph for finishedGitPull](image)

12.12.4.14 finishedGitPush

```c
class Pass {
public:
    // Constructor
    Pass();

    // Destructor
    ~Pass();

    // Signal
    void finishedGitPush (const QString &, const QString &); // [signal]

private:
    // Member variables
    QString m_string1;
    QString m_string2;
};
```

Here is the caller graph for this function:

![Caller Graph for finishedGitPush](image)

12.12.4.15 finishedInit

```c
class Pass {
public:
    // Constructor
    Pass();

    // Destructor
    ~Pass();

    // Signal
    void finishedInit (const QString &, const QString &); // [signal]

private:
    // Member variables
    QString m_string1;
    QString m_string2;
};
```
Here is the caller graph for this function:

12.12.4.16 finishedInsert

```c++
void Pass::finishedInsert (
    const QString &, 
    const QString & ) [signal]
```

Here is the caller graph for this function:

12.12.4.17 finishedMove

```c++
void Pass::finishedMove (
    const QString &, 
    const QString & ) [signal]
```

Here is the caller graph for this function:
12.12.4.18  finishedOtpGenerate

void Pass::finishedOtpGenerate ( const QString & ) [signal]

Here is the caller graph for this function:

![Caller Graph for finishedOtpGenerate](image)

12.12.4.19  finishedRemove

void Pass::finishedRemove ( const QString &, const QString & ) [signal]

Here is the caller graph for this function:

![Caller Graph for finishedRemove](image)

12.12.4.20  finishedShow

void Pass::finishedShow ( const QString & ) [signal]

Here is the caller graph for this function:

![Caller Graph for finishedShow](image)
12.12.4.21  Generate_b()

QString Pass::Generate_b (  
    unsigned int length,  
    const QString & charset ) [virtual]

Pass::Generate use either pwgen or internal password generator.

Parameters

<table>
<thead>
<tr>
<th>length</th>
<th>of the desired password</th>
</tr>
</thead>
<tbody>
<tr>
<td>charset</td>
<td>to use for generation</td>
</tr>
</tbody>
</table>

Returns

the password

Definition at line 69 of file pass.cpp.

Here is the call graph for this function:

![Call Graph Diagram]

12.12.4.22  GenerateGPGKeys()

void Pass::GenerateGPGKeys (  
    QString batch )

Pass::GenerateGPGKeys internal gpg keypair generator . .
Parameters

| batch | GnuPG style configuration string |

Definition at line 114 of file pass.cpp.

Here is the call graph for this function:

```
Pass::GenerateGPGKeys
    Pass::executeWrapper
    QtPassSettings::getGpgExecutable
    QtPassSettings::getInstance
```

12.12.4.23 generateRandomPassword()

```cpp
QString Pass::generateRandomPassword (  
    const QString & charset,  
    unsigned int length ) [protected]
```

Definition at line 335 of file pass.cpp.

Here is the call graph for this function:

```
Pass::generateRandomPassword
    Pass::boundedRandom
```

Here is the caller graph for this function:

```
Pass::Generate_b
    Pass::generateRandomPassword
```

12.12.4.24 getRecipientList()

```cpp
QStringList Pass::getRecipientList (  
    QString for_file ) [static]
```

Pass::getRecipientList return list of gpg-id's to encrypt for.
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>for_file</code></td>
<td>which file (folder) would you like recipients for</td>
</tr>
</tbody>
</table>

Returns

- recipients gpg-id contents

Definition at line 263 of file pass.cpp.

Here is the call graph for this function:

```
Pass::getRecipientList QtPassSettings::getPassStore QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
UsersDialog::UsersDialog Pass::getRecipientList Pass::getRecipientString ImitatePass::Insert
```

### 12.12.4.25 getRecipientString()

```
QStringList Pass::getRecipientString ( 
    QString `for_file`, 
    QString separator = " ", 
    int * count = NULL ) [static]
```

Pass::getRecipientString formated string for use with GPG.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>for_file</code></td>
<td>which file (folder) would you like recipients for</td>
</tr>
<tr>
<td><code>separator</code></td>
<td>formating separator eg: &quot; -r &quot;</td>
</tr>
<tr>
<td><code>count</code></td>
<td></td>
</tr>
</tbody>
</table>

Generated by Doxygen
Returns

recipient string

Definition at line 299 of file pass.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.12.4.26 GitInit()

virtual void Pass::GitInit ( ) [pure virtual]

Implemented in ImitatePass, and RealPass.

12.12.4.27 GitPull()

virtual void Pass::GitPull ( ) [pure virtual]

Implemented in ImitatePass, and RealPass.

12.12.4.28 GitPull_b()

virtual void Pass::GitPull_b ( ) [pure virtual]

Implemented in ImitatePass, and RealPass.
12.12.4.29 GitPush()

virtual void Pass::GitPush ( ) [pure virtual]

Implemented in ImitatePass, and RealPass.

Here is the caller graph for this function:

```
MainWindow::onPush          Pass::GitPush
```

12.12.4.30 init()

void Pass::init ( )

Definition at line 45 of file pass.cpp.

Here is the call graph for this function:

```
Pass::init          QtPassSettings::getGpgHome          QtPassSettings::getInstance
```

Here is the caller graph for this function:
12.12.4.31  Init()

virtual void Pass::Init (  
    QString path,  
    const QList< UserInfo > & users ) [pure virtual]

Implemented in ImitatePass, and RealPass.

Here is the caller graph for this function:

![Caller Graph]

12.12.4.32  Insert()

virtual void Pass::Insert (  
    QString file,  
    QString value,  
    bool force ) [pure virtual]

Implemented in ImitatePass, and RealPass.

12.12.4.33  listKeys() [1/2]

QList< UserInfo > Pass::listKeys (  
    QString keystring = "",  
    bool secret = false )

Pass::listKeys list users.

Parameters

<table>
<thead>
<tr>
<th>keystring</th>
<th>list private keys</th>
</tr>
</thead>
</table>

Returns

QList< UserInfo > users

Definition at line 178 of file pass.cpp.
Here is the call graph for this function:

```
Pass::listKeys
    Executor::executeBlocking
    QtPassSettings::getGpgExecutable
    QtPassSettings::getInstance
```

### 12.12.4.34 listKeys() [2/2]

```cpp
QList<UserInfo> Pass::listKeys ( 
    QStringList keystrings,
    bool secret = false )
```

**Pass::listKeys** list users.

**Parameters**

- **keystrings**
- **secret** list private keys

**Returns**

`QList<UserInfo>` users

Definition at line 128 of file pass.cpp.

Here is the call graph for this function:

```
Pass::listKeys
    Executor::executeBlocking
    QtPassSettings::getGpgExecutable
    QtPassSettings::getInstance
```
Here is the caller graph for this function:

```
Pass::listKeys
     |                  |
     |  UsersDialog::UsersDialog  |
     v
Pass::listKeys
```

### 12.12.4.35 Move()

```
virtual void Pass::Move ( 
    const QString srcDir, 
    const QString dest, 
    const bool force = false ) [pure virtual]
```

Implemented in *ImitatePass*, and *RealPass*.

Here is the caller graph for this function:

```
StoreModel::dropMimeData
     |                  |
     |  Pass::Move  |
```

### 12.12.4.36 OtpGenerate()

```
virtual void Pass::OtpGenerate ( 
    QString file ) [pure virtual]
```

Implemented in *ImitatePass*, and *RealPass*. 
12.12.4.37  \texttt{processErrorExit}

```cpp
void Pass::processErrorExit (  
    int exitCode,  
    const QString & err ) [signal]
```

Here is the caller graph for this function:

![Caller Graph for processErrorExit](image)

12.12.4.38  \texttt{Remove()}

```cpp
virtual void Pass::Remove (  
    QString file,  
    bool isDir ) [pure virtual]
```

Implemented in \texttt{ImitatePass}, and \texttt{RealPass}.

12.12.4.39  \texttt{Show()}

```cpp
virtual void Pass::Show (  
    QString file ) [pure virtual]
```

Implemented in \texttt{ImitatePass}, and \texttt{RealPass}.

Here is the caller graph for this function:

![Caller Graph for Show](image)
12.12.4.40 startingExecuteWrapper

void Pass::startingExecuteWrapper ( ) [signal]

Here is the caller graph for this function:

```
Pass::Pass  Pass::startingExecuteWrapper
```

12.12.4.41 statusMsg

void Pass::statusMsg ( QString ,
                       int ) [signal]

12.12.4.42 updateEnv()

void Pass::updateEnv ( )

Pass::updateEnv update the execution environment (used when switching profiles)

Definition at line 242 of file pass.cpp.

Here is the call graph for this function:

```
Pass::updateEnv
    QtPassSettings::getInstance
        QtPassSettings::getPassStore
            Executor::setEnvironment
```

Here is the caller graph for this function:

```
MainWindow::config
    MainWindow::MainWindow
        QtPass::init
            QtPass::updateEnv
```

Generated by Doxygen
12.13 PasswordConfiguration Struct Reference

Holds the Password configuration settings.

```
#include <passwordconfiguration.h>
```

**Public Types**

- enum `characterSet` {
  `ALLCHARS = 0, ALPHABETICAL, ALPHANUMERIC, CUSTOM, CHARSETS_COUNT` 
  
  *The selected character set.*

**Public Member Functions**

- `PasswordConfiguration()`

**Public Attributes**

- enum `PasswordConfiguration::characterSet` selected
- int `length`

  *Length of the password.*

  - `QString Characters [CHARSETS_COUNT]`

  *The different character sets.*

12.13.1 Detailed Description

Holds the Password configuration settings.

Definition at line 10 of file passwordconfiguration.h.

12.13.2 Member Enumeration Documentation

12.13.2.1 `characterSet`

enum `PasswordConfiguration::characterSet`

The selected character set.
### Enumerator

| ALLCHARS   | ALPHABETICAL | ALPHANUMERIC | CUSTOM | CHARSETS_COUNT |

Definition at line 14 of file passwordconfiguration.h.

### 12.13.3 Constructor & Destructor Documentation

#### 12.13.3.1 PasswordConfiguration()

PasswordConfiguration::PasswordConfiguration () [inline]

Definition at line 29 of file passwordconfiguration.h.

### 12.13.4 Member Data Documentation

#### 12.13.4.1 Characters

QString PasswordConfiguration::Characters[CHARSETS_COUNT]

The different character sets.

Definition at line 28 of file passwordconfiguration.h.

#### 12.13.4.2 length

int PasswordConfiguration::length

Length of the password.

Definition at line 24 of file passwordconfiguration.h.
12.14 PasswordDialog Class Reference

PasswordDialog Handles the inserting and editing of passwords.

#include <passworddialog.h>

Inheritance diagram for PasswordDialog:

```
QDialog

PasswordDialog
```

Collaboration diagram for PasswordDialog:

```
QDialog

PasswordDialog
```

Public Slots

- void setPass (const QString &output)
Public Member Functions

- **PasswordDialog** (const PasswordConfiguration &passConfig, QWidget *parent=nullptr)
  
  PasswordDialog::PasswordDialog basic constructor.

- **PasswordDialog** (const QString &file, const bool &isNew, QWidget *parent=nullptr)
  
  PasswordDialog::PasswordDialog complete constructor.

- ~PasswordDialog ()
  
  Pass[[]]wordDialog::~PasswordDialog basic destructor.

- void setPassword (QString password)
  
  Sets content in the password field in the interface.

- QString getPassword ()
  
  Returns the password as set in the password field in the interface.

- void setTemplate (QString rawFields, bool useTemplate)
  
  Sets content in the template for the interface.

- void templateAll (bool templateAll)
  
  PasswordDialog::templateAll basic setter for use in PasswordDialog::setPassword templating all tokenisable lines.

- void setLength (int l)
  
  PasswordDialog::setLength PasswordDialog::setLength password length.

- void setPasswordCharTemplate (int t)
  
  PasswordDialog::setPasswordCharTemplate PasswordDialog::setPasswordCharTemplate chose the template style.

- void usePwgen (bool usePwgen)
  
  PasswordDialog::usePwgen PasswordDialog::usePwgen don't use own password generator.

12.14.1 Detailed Description

**PasswordDialog** Handles the inserting and editing of passwords.

Includes templated views.

Definition at line 20 of file passworddialog.h.

12.14.2 Constructor & Destructor Documentation

12.14.2.1 PasswordDialog() [1/2]

PasswordDialog::PasswordDialog {
    const PasswordConfiguration & passConfig,
    QWidget * parent = nullptr } [explicit]

PasswordDialog::PasswordDialog basic constructor.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passConfig</td>
<td>configuration constant</td>
</tr>
<tr>
<td>parent</td>
<td></td>
</tr>
</tbody>
</table>
Definition at line 20 of file passworddialog.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

### 12.14.2.2 PasswordDialog() [2/2]

PasswordDialog::PasswordDialog (  
    const QString & file,  
    const bool & isNew,  
    QWidget * parent = nullptr )

**PasswordDialog::PasswordDialog** complete constructor.

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td></td>
</tr>
<tr>
<td>isNew</td>
<td></td>
</tr>
<tr>
<td>parent</td>
<td>pointer</td>
</tr>
</tbody>
</table>
Here is the call graph for this function:

12.14.2.3 ~PasswordDialog()

PasswordDialog::~PasswordDialog ( )
Pass[]wordDialog::~PasswordDialog basic destructor.
Definition at line 72 of file passworddialog.cpp.

12.14.3 Member Function Documentation

12.14.3.1 getPassword()

QString PasswordDialog::getPassword ( )
Returns the password as set in the password field in the interface.
PasswordDialog::getPassword join the (templated) fields to a QString for writing back.
Returns
password as a QString
See also
setPassword
Returns
collapsed password.
Definition at line 156 of file passworddialog.cpp.
12.14.3.2  

setLength()

```cpp
void PasswordDialog::setLength (  
    int l  )
```

PasswordDialog::setLength  

PasswordDialog::setLength  

password length.

Parameters

- `l`

Definition at line 208 of file passworddialog.cpp.

Here is the caller graph for this function:

![Caller Graph of setLength Function](diagram)

12.14.3.3  

setPass

```cpp
void PasswordDialog::setPass (  
    const QString & output  ) [slot]
```

Definition at line 229 of file passworddialog.cpp.

Here is the call graph for this function:

![Call Graph of setPass Function](diagram)
Here is the caller graph for this function:

12.14.3.4 setPassword()

```cpp
void PasswordDialog::setPassword (  
    QString password  
)
```

Sets content in the password field in the interface.

**PasswordDialog::setPassword** populate the (templated) fields.

**Parameters**

- `password` the password as a QString

**See also**

- getPassword

**Parameters**

- `password`
Here is the call graph for this function:

```plaintext
diagram not rendered here
```

Here is the caller graph for this function:

```plaintext
diagram not rendered here
```

12.14.3.5 `setPasswordCharTemplate()`

```c
void PasswordDialog::setPasswordCharTemplate ( int t )
```

`PasswordDialog::setPasswordCharTemplate` chose the template style.

Parameters

- `t`

Definition at line 215 of file passworddialog.cpp.
Here is the caller graph for this function:

```
PasswordDialog::Password
    Dialog
       PasswordDialog::setPassword
          CharTemplate
```

### 12.14.3.6 setTemplate()

```cpp
define PasswordDialog::setTemplate (  
    QString rawFields,  
    bool useTemplate  
)
```

Sets content in the template for the interface.

**PasswordDialog::setTemplate** set the template and create the fields.

**Parameters**

<table>
<thead>
<tr>
<th>rawFields</th>
<th>is the template as a QString</th>
</tr>
</thead>
<tbody>
<tr>
<td>useTemplate</td>
<td>whether the template is used</td>
</tr>
</tbody>
</table>

Definition at line 174 of file passworddialog.cpp.

Here is the caller graph for this function:

```
PasswordDialog::Password
    Dialog
       PasswordDialog::setTemplate
```

### 12.14.3.7 templateAll()

```cpp
define PasswordDialog::templateAll (  
    bool templateAll  
)
```

**PasswordDialog::templateAll** basic setter for use in **PasswordDialog::setPassword** templating all tokenisable lines.
Parameters

Definition at line 199 of file passworddialog.cpp.

Here is the caller graph for this function:

PasswordDialog::Password
Dialog

PasswordDialog::templateAll

12.14.3.8 usePwgen()

void PasswordDialog::usePwgen ( bool usePwgen )

PasswordDialog::usePwgen PasswordDialog::usePwgen don't use own password generator.

Parameters

Definition at line 224 of file passworddialog.cpp.

Here is the caller graph for this function:

PasswordDialog::Password
Dialog

PasswordDialog::usePwgen

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

- src/passworddialog.h
- src/passworddialog.cpp
12.15 QProgressIndicator Class Reference

The QProgressIndicator class lets an application display a progress indicator to show that a lengthy task is under way.

#include <qprogressindicator.h>

Inheritance diagram for QProgressIndicator:

```
   QWidget
   ^
   |  
QProgressIndicator
```

Collaboration diagram for QProgressIndicator:

```
   QWidget
   ^
   |  
QProgressIndicator
```

Public Slots

- void startAnimation ()
  Starts the spin animation.
- void stopAnimation ()
  Stops the spin animation.
- void setAnimationDelay (int delay)
  Sets the delay between animation steps.
- void setDisplayedWhenStopped (bool state)
  Sets whether the component hides itself when it is not animating.
- void setColor (const QColor &color)
  Sets the color of the components to the given color.
Public Member Functions

- **QProgressIndicator** (QWidget *parent=0)
  
  QProgressIndicator::QProgressIndicator constructor.

- **int animationDelay () const**
  
  Returns the delay between animation steps.

- **bool isAnimated () const**
  
  Returns a Boolean value indicating whether the component is currently animated.

- **bool isDisplayedWhenStopped () const**
  
  Returns a Boolean value indicating whether the receiver shows itself even when it is not animating.

- **const QColor & color () const**
  
  Returns the color of the component.

- **virtual QSize sizeHint () const**
  
  QProgressIndicator::sizeHint default minimum size.

- **int heightForWidth (int w) const**
  
  QProgressIndicator::heightForWidth square ratio.

Protected Member Functions

- **virtual void timerEvent (QTimerEvent *event)**
  
  QProgressIndicator::timerEvent do the actual animation.

- **virtual void paintEvent (QPaintEvent *event)**
  
  QProgressIndicator::paintEvent draw the spinner.

Properties

- **int delay**
  
  QProgressIndicator::delay in milliseconds.

- **bool displayedWhenStopped**
  
  QProgressIndicator::displayedWhenStopped render when not spinning.

- **QColor color**
  
  QProgressIndicator::color of the component.

12.15.1 Detailed Description

The **QProgressIndicator** class lets an application display a progress indicator to show that a lengthy task is under way.

Progress indicators are indeterminate and do nothing more than spin to show that the application is busy.

See also

QProgressBar

Definition at line 16 of file qprogressindicator.h.

12.15.2 Constructor & Destructor Documentation

12.15.2.1 QProgressIndicator()

QProgressIndicator::QProgressIndicator (QWidget *parent = 0) [explicit]

QProgressIndicator::QProgressIndicator constructor.
Parameters

parent widget the indicator is placed in.

Definition at line 8 of file qprogressindicator.cpp.

### 12.15.3 Member Function Documentation

#### 12.15.3.1 animationDelay()

```cpp
int QProgressIndicator::animationDelay ( ) const [inline]
```

Returns the delay between animation steps.

Returns

The number of milliseconds between animation steps. By default, the animation delay is set to 40 milliseconds.

See also

setColor

Definition at line 40 of file qprogressindicator.h.

#### 12.15.3.2 color()

```cpp
const QColor& QProgressIndicator::color ( ) const [inline]
```

Returns the color of the component.

See also

setColor

Definition at line 60 of file qprogressindicator.h.

#### 12.15.3.3 heightForWidth()

```cpp
int QProgressIndicator::heightForWidth ( int w ) const
```

QProgressIndicator::heightForWidth square ratio.
### Parameters

- **w** requested width

### Returns

- w returned height

Definition at line 70 of file qprogressindicator.cpp.

#### 12.15.3.4 isAnimated()

```cpp
bool QProgressIndicator::isAnimated ( ) const
```

Returns a Boolean value indicating whether the component is currently animated.

Returns

- Animation state.

See also

- startAnimation stopAnimation

Definition at line 15 of file qprogressindicator.cpp.

Here is the caller graph for this function:

![Caller graph for isAnimated()](image)

#### 12.15.3.5 isDisplayedWhenStopped()

```cpp
bool QProgressIndicator::isDisplayedWhenStopped ( ) const
```

Returns a Boolean value indicating whether the receiver shows itself even when it is not animating.

Returns

- Return true if the progress indicator shows itself even when it is not animating. By default, it returns false.

See also

- setDisplayedWhenStopped

Definition at line 23 of file qprogressindicator.cpp.
12.15.3.6 paintEvent()

```cpp
void QProgressIndicator::paintEvent (  
    QPaintEvent * event ) [protected], [virtual]
```

QProgressIndicator::paintEvent draw the spinner.

Definition at line 84 of file qprogressindicator.cpp.

Here is the call graph for this function:

```
QProgressIndicator::paintEvent -> QProgressIndicator::isAnimated
```

12.15.3.7 setAnimationDelay

```cpp
void QProgressIndicator::setAnimationDelay (  
    int delay ) [slot]
```

Sets the delay between animation steps.

Setting the `delay` to a value larger than 40 slows the animation, while setting the `delay` to a smaller value speeds it up.

Parameters

- `delay` The delay, in milliseconds.

See also

- animationDelay

Definition at line 43 of file qprogressindicator.cpp.

12.15.3.8 setColor

```cpp
void QProgressIndicator::setColor (  
    const QColor & color ) [slot]
```

Sets the color of the components to the given color.
12.15 QProgressIndicator Class Reference

See also

color

Definition at line 53 of file qprogressindicator.cpp.

12.15.3.9  setDisplayedWhenStopped

void QProgressIndicator::setDisplayedWhenStopped ( bool state ) [slot]

Sets whether the component hides itself when it is not animating.

Parameters

state  The animation state. Set false to hide the progress indicator when it is not animating; otherwise true.

See also

isDisplayedWhenStopped

Definition at line 17 of file qprogressindicator.cpp.

12.15.3.10  sizeHint()

QSize QProgressIndicator::sizeHint ( ) const [virtual]

QProgressIndicator::sizeHint default minimum size.

Returns

QSize(20, 20)

Definition at line 63 of file qprogressindicator.cpp.

12.15.3.11  startAnimation

void QProgressIndicator::startAnimation ( ) [slot]

Starts the spin animation.

See also

stopAnimation isAnimated

Definition at line 27 of file qprogressindicator.cpp.
12.15.3.12 stopAnimation

void QProgressIndicator::stopAnimation () [slot]

Stops the spin animation.

See also

   startAnimation isAnimated

Definition at line 34 of file qprogressindicator.cpp.

12.15.3.13 timerEvent()

void QProgressIndicator::timerEvent (QTimerEvent * event) [protected], [virtual]

QProgressIndicator::timerEvent do the actual animation.

Definition at line 75 of file qprogressindicator.cpp.

12.15.4 Property Documentation

12.15.4.1 color

QColor QProgressIndicator::color [read], [write]

QProgressIndicator::color of the component..

Definition at line 1 of file qprogressindicator.h.

12.15.4.2 delay

int QProgressIndicator::delay [read], [write]

QProgressIndicator::delay in milliseconds.

Definition at line 1 of file qprogressindicator.h.
12.15.4.3 displayedWhenStopped

```cpp
bool QProgressIndicator::displayedWhenStopped [read], [write]
```

QProgressIndicator::displayedWhenStopped render when not spinning.

Definition at line 1 of file qprogressindicator.h.

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

- src/qprogressindicator.h
- src/qprogressindicator.cpp

12.16 QPushButtonAsQRCode Class Reference

Stylish widget to display the field as QR Code.

```cpp
#include <QPushButtonAsQRCode.h>
```

Inheritance diagram for QPushButtonAsQRCode:

```
QPushButton

QPushButtonAsQRCode
```

Collaboration diagram for QPushButtonAsQRCode:
Signals

• void clicked (QString)

Public Member Functions

• QPushButtonAsQRCode (const QString &textToCopy="", QWidget *parent=nullptr)
  QPushButtonAsQRCode::QPushButtonAsQRCode basic constructor.
• QString getTextToCopy () const
  QPushButtonAsQRCode::getTextToCopy returns the text of associated text field.
• void setTextToCopy (const QString &value)
  QPushButtonAsQRCode::setTextToCopy sets text from associated text field.

12.16.1 Detailed Description

Stylish widget to display the field as QR Code.

Definition at line 11 of file qpushbuttonasqrcode.h.

12.16.2 Constructor & Destructor Documentation

12.16.2.1 QPushButtonAsQRCode()

QPushButtonAsQRCode::QPushButtonAsQRCode ( 
    const QString & textToCopy = "", 
    QWidget * parent = nullptr ) [explicit]

QPushButtonAsQRCode::QPushButtonAsQRCode basic constructor.

Parameters

<table>
<thead>
<tr>
<th>textToCopy</th>
<th>the text to display as qrcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>the parent window</td>
</tr>
</tbody>
</table>

Definition at line 12 of file qpushbuttonasqrcode.cpp.

12.16.3 Member Function Documentation

12.16.3.1 clicked

void QPushButtonAsQRCode::clicked ( 
    QString ) [signal]
Here is the caller graph for this function:

![Caller Graph](image)

### 12.16.3.2 getTextToCopy()

```cpp
QString QPushButtonAsQRCode::getTextToCopy ( ) const
```

`QPushButtonAsQRCode::getTextToCopy` returns the text of associated text field.

#### Returns

- `QString textToCopy`

Definition at line 25 of file qpushbuttonasqrcode.cpp.

### 12.16.3.3 setTextToCopy()

```cpp
void QPushButtonAsQRCode::setTextToCopy ( const QString & value )
```

`QPushButtonAsQRCode::setTextToCopy` sets text from associated text field.

#### Parameters

- `value` : `QString text to be copied`

Definition at line 32 of file qpushbuttonasqrcode.cpp.

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

- `src/qpushbuttonasqrcode.h`
- `src/qpushbuttonasqrcode.cpp`

Generated by Doxygen
12.17 QPushButtonShowPassword Class Reference

#include <qpushbuttonshowpassword.h>

Inheritance diagram for QPushButtonShowPassword:

Collaboration diagram for QPushButtonShowPassword:

**Signals**

- void clicked (QString)

**Public Member Functions**

- QPushButtonShowPassword (QLineEdit *line, QWidget *parent=nullptr)

  QPushButtonAsQRCode::QPushButtonAsQRCode basic constructor.

12.17.1 Detailed Description

Definition at line 12 of file qpushbuttonshowpassword.h.
12.17.2 Constructor & Destructor Documentation

12.17.2.1 QPushButtonShowPassword()

QPushButtonShowPassword::QPushButtonShowPassword ( 
  QLineEdit * line,
  QWidget * parent = nullptr ) [explicit]

QPushButtonAsQRCode::QPushButtonAsQRCode basic constructor.

Parameters

<table>
<thead>
<tr>
<th>textToCopy</th>
<th>the text to display as qrcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>the parent window</td>
</tr>
</tbody>
</table>

Definition at line 12 of file qpushbuttonshowpassword.cpp.

12.17.3 Member Function Documentation

12.17.3.1 clicked

void QPushButtonShowPassword::clicked ( 
  QString ) [signal]

Here is the caller graph for this function:

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

- src/qpushbuttonshowpassword.h
- src/qpushbuttonshowpassword.cpp

Generated by Doxygen
12.18 QPushButtonWithClipboard Class Reference

Stylish widget to allow copying of password and account details.

```cpp
#include <qpushbuttonwithclipboard.h>
```

Inheritance diagram for QPushButtonWithClipboard:

```
QPushButtonWithClipboard
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>QPushButton</td>
</tr>
</tbody>
</table>
| QPushButtonWithClipboard
```

Collaboration diagram for QPushButtonWithClipboard:

```
QPushButton
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
| QPushButtonWithClipboard
```

**Signals**

- void **clicked** (QString)

**Public Member Functions**

- **QPushButtonWithClipboard** (const QString &textToCopy="", QWidget *parent=nullptr)
  
  `QPushButtonWithClipboard::QPushButtonWithClipboard` basic constructor.

- QString **getTextToCopy** () const
  
  `QPushButtonWithClipboard::getTextToCopy` returns the text of associated text field.

- void **setTextToCopy** (const QString &value)
  
  `QPushButtonWithClipboard::setTextToCopy` sets text from associated text field.
12.18 QPushButtonWithClipboard Class Reference

12.18.1 Detailed Description

Stylish widget to allow copying of password and account details.

Definition at line 11 of file qpushbuttonwithclipboard.h.

12.18.2 Constructor & Destructor Documentation

12.18.2.1 QPushButtonWithClipboard()

QPushButtonWithClipboard::QPushButtonWithClipboard (const QString & textToCopy = "", QWidget * parent = nullptr) [explicit]

QPushButtonWithClipboard::QPushButtonWithClipboard basic constructor.

Parameters

<table>
<thead>
<tr>
<th>textToCopy</th>
<th>the text to paste into the clipboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent</td>
<td>the parent window</td>
</tr>
</tbody>
</table>

Definition at line 12 of file qpushbuttonwithclipboard.cpp.

12.18.3 Member Function Documentation

12.18.3.1 clicked

void QPushButtonWithClipboard::clicked (QString ) [signal]

Here is the caller graph for this function:

![Caller Graph](image-url)

Generated by Doxygen
12.18.3.2 getTextToCopy()

QString QPushButtonWithClipboard::getTextToCopy ( ) const

QPushButtonWithClipboard::getTextToCopy returns the text of associated text field.

Returns

QString textToCopy

Definition at line 27 of file qpushbuttonwithclipboard.cpp.

12.18.3.3 setTextToCopy()

void QPushButtonWithClipboard::setTextToCopy ( const QString & value )

QPushButtonWithClipboard::setTextToCopy sets text from associated text field.

Parameters

| value | QString text to be copied |

Definition at line 34 of file qpushbuttonwithclipboard.cpp.

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

- src/qpushbuttonwithclipboard.h
- src/qpushbuttonwithclipboard.cpp

12.19 QtPass Class Reference

#include <qtpass.h>

Inheritance diagram for QtPass:
Collaboration diagram for QtPass:

```
QObject
  ↑
QtPass
```

### Public Slots

- void **clearClipboard ()**
  
  `MainWindow::clearClipboard` remove clipboard contents.

- void **copyTextToClipboard (const QString &text)**
  
  `MainWindow::copyTextToClipboard` copies text to your clipboard.

- void **showTextAsQRCode (const QString &text)**
  
  Displays the text as qrcode

### Public Member Functions

- **QtPass (MainWindow *mainWindow)**
- **~QtPass ()**
  
  `QtPass::~QtPass` destroy!

- bool **init ()**
  
  `QtPass::init` make sure we are ready to go as soon as possible.

- void **setClippedText (const QString &text, const QString &p_output=QString())**

- void **clearClippedText ()**

- void **setClipboardTimer ()**

- bool **isFreshStart ()**

- void **setFreshStart (const bool &fs)**

### 12.19.1 Detailed Description

Definition at line 10 of file qtpass.h.

### 12.19.2 Constructor & Destructor Documentation

Generated by Doxygen
12.19.2.1 QtPass()

QtPass::QtPass (  
    MainWindow * mainWindow  )

Definition at line 27 of file qtpass.cpp.

Here is the call graph for this function:

![Call graph for QtPass()](image)

12.19.2.2 ~QtPass()

QtPass::~QtPass (  )

QtPass::~QtPass destroy!

Definition at line 43 of file qtpass.cpp.

Here is the call graph for this function:

![Call graph for ~QtPass()](image)

12.19.3 Member Function Documentation
12.19.3.1 clearClipboard

```cpp
void QtPass::clearClipboard ( ) [slot]
```

`MainWindow::clearClipboard` remove clipboard contents.

Definition at line 373 of file qtpass.cpp.

Here is the call graph for this function:

![Call Graph for clearClipboard](image1.png)

Here is the caller graph for this function:

![Caller Graph for clearClipboard](image2.png)

12.19.3.2 clearClippedText()

```cpp
void QtPass::clearClippedText ( )
```

Definition at line 364 of file qtpass.cpp.

Here is the caller graph for this function:

![Caller Graph for clearClippedText](image3.png)

12.19.3.3 copyTextToClipboard

```cpp
void QtPass::copyTextToClipboard ( const QString & text ) [slot]
```

`MainWindow::copyTextToClipboard` copies text to your clipboard.

Generated by Doxygen
Parameters

<table>
<thead>
<tr>
<th>text</th>
</tr>
</thead>
</table>

Definition at line 397 of file qtpass.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

### 12.19.3.4 init()

```cpp
bool QtPass::init ()
```

*QtPass::init* make sure we are ready to go as soon as possible.

Definition at line 60 of file qtpass.cpp.
Here is the call graph for this function:

![Call Graph Image]

Here is the caller graph for this function:

![Caller Graph Image]

### 12.19.3.5 isFreshStart()

```cpp
bool QtPass::isFreshStart() [inline]
```

Definition at line 21 of file qtpass.h.

Generated by Doxygen
Here is the caller graph for this function:

```
MainWindow::MainWindow -> QtPass::init -> MainWindow::config -> QtPass::isFreshStart
```

### 12.19.3.6 `setClipboardTimer()`

```cpp
template

void QtPass::setClipboardTimer ()
```

Definition at line 366 of file qtpass.cpp.

Here is the call graph for this function:

```
QtPass::setClipboardTimer -> QtPassSettings::getAutoclear
```

Here is the caller graph for this function:

```
MainWindow::MainWindow -> QtPass::init -> MainWindow::config -> QtPass::isFreshStart
```

### 12.19.3.7 `setClippedText()`

```cpp
template

void QtPass::setClippedText (const QString &password, const QString &p_output = QString())
```

Definition at line 356 of file qtpass.cpp.
Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

12.19.3.8 setFreshStart()

```cpp
class QtPass
{
public:
    void setFreshStart (const bool & fs) [inline]

    // Definition at line 22 of file qtpass.h.
}
```

Here is the caller graph for this function:

![Caller Graph]

12.19.3.9 showTextAsQRCode

```cpp
class QtPass
{
public:
    void showTextAsQRCode (const QString & text) [slot]

    // displays the text as qrcode
}
```
12.20 QtPassSettings Class Reference

Singleton that stores qtpass' settings, saves and loads config.

#include <qtpasssettings.h>

Inheritance diagram for QtPassSettings:
Collaboration diagram for QtPassSettings:

![Collaboration Diagram](image)

**Static Public Member Functions**

- static `QtPassSettings * getInstance ()`
- static `QString getVersion (const QString &defaultValue=QVariant().toString())`
- static `void setVersion (const QString &version)`
- static `QByteArray getGeometry (const QByteArray &defaultValue=QVariant().toByteArray())`
- static `void setGeometry (const QByteArray &geometry)`
- static `QByteArray getSavestate (const QByteArray &defaultValue=QVariant().toByteArray())`
- static `void setSavestate (const QByteArray &saveState)`
- static `QPoint getPos (const QPoint &defaultValue=QVariant().toPoint())`
- static `void setPos (const QPoint &pos)`
- static `QSize getSize (const QSize &defaultValue=QVariant().toSize())`
- static `void setSize (const QSize &size)`
- static `bool isMaximized (const bool &defaultValue=QVariant().toBool())`
- static `void setMaximized (const bool &maximized)`
- static `bool isUsePass (const bool &defaultValue=QVariant().toBool())`
- static `void setUsePass (const bool &usePass)`
- static `int getClipBoardTypeRaw (const Enums::clipBoardType &defaultvalue=Enums::CLIPBOARD_NEVER)`
- static `Enums::clipBoardType getClipBoardType (const Enums::clipBoardType &defaultvalue=Enums::CLIPBOARD_NEVER)`
- static `void setClipBoardType (const int &clipBoardType)`
- static `bool isUseSelection (const bool &defaultValue=QVariant().toBool())`
- static `void setUseSelection (const bool &useSelection)`
- static `bool isUseAutoclear (const bool &defaultValue=QVariant().toBool())`
- static `void setUseAutoclear (const bool &useAutoclear)`
- static `int getAutoclearSeconds (const int &defaultValue=QVariant().toInt())`
- static `void setAutoclearSeconds (const int &autoClearSeconds)`
- static `bool isUseAutoclearPanel (const bool &defaultValue=QVariant().toBool())`
- static `void setUseAutoclearPanel (const bool &useAutoclearPanel)`
- static `int getAutoclearPanelSeconds (const int &defaultValue=QVariant().toInt())`
- static `void setAutoclearPanelSeconds (const int &autoClearPanelSeconds)`
- static `bool isHidePassword (const bool &defaultValue=QVariant().toBool())`
- static `void setHidePassword (const bool &hidePassword)`
- static `bool isHideContent (const bool &defaultValue=QVariant().toBool())`
- static `void setHideContent (const bool &hideContent)`
- static `bool isAddGPGId (const bool &defaultValue=QVariant().toBool())`
- static `void setAddGPGId (const bool &addGPGId)`
- static `QString getPassStore (const QString &defaultValue=QVariant().toString())`
• static void **setPassStore** (const QString &passStore)
• static void **initExecutables** ()
• static QString **getPassExecutable** (const QString &defaultValue=QVariant().toString())
• static void **setPassExecutable** (const QString &passExecutable)
• static QString **getGitExecutable** (const QString &defaultValue=QVariant().toString())
• static void **setGitExecutable** (const QString &gitExecutable)
• static QString **getGpgExecutable** (const QString &defaultValue=QVariant().toString())
• static void **setGpgExecutable** (const QString &gpgExecutable)
• static QString **getPwgenExecutable** (const QString &defaultValue=QVariant().toString())
• static void **setPwgenExecutable** (const QString &pwgenExecutable)
• static QString **getGpgHome** (const QString &defaultValue=QVariant().toString())
• static bool **isUseWebDav** (const bool &defaultValue=QVariant().toBool())
• static void **setUseWebDav** (const bool &useWebDav)
• static QString **getWebDavUrl** (const QString &defaultValue=QVariant().toString())
• static void **setWebDavUrl** (const QString &webDavUrl)
• static QString **getWebDavUser** (const QString &defaultValue=QVariant().toString())
• static void **setWebDavUser** (const QString &webDavUser)
• static QString **getWebDavPassword** (const QString &defaultValue=QVariant().toString())
• static void **setWebDavPassword** (const QString &webDavPassword)
• static QString **getProfile** (const QString &defaultValue=QVariant().toString())
• static void **setProfile** (const QString &profile)
• static bool **isUseGit** (const bool &defaultValue=QVariant().toBool())
• static void **setUseGit** (const bool &useGit)
• static bool **isUseOtp** (const bool &defaultValue=QVariant().toBool())
• static void **setUseOtp** (const bool &useOtp)
• static bool **isUseQrencode** (const bool &defaultValue=QVariant().toBool())
• static void **setUseQrencode** (const bool &useQrencode)
• static QString **getQrencodeExecutable** (const QString &defaultValue=QVariant().toString())
• static void **setQrencodeExecutable** (const QString &qrencodeExecutable)
• static bool **isUsePwgen** (const bool &defaultValue=QVariant().toBool())
• static void **setUsePwgen** (const bool &usePwgen)
• static bool **isAvoidCapitals** (const bool &defaultValue=QVariant().toBool())
• static void **setAvoidCapitals** (const bool &avoidCapitals)
• static bool **isAvoidNumbers** (const bool &defaultValue=QVariant().toBool())
• static void **setAvoidNumbers** (const bool &avoidNumbers)
• static bool **isLessRandom** (const bool &defaultValue=QVariant().toBool())
• static void **setLessRandom** (const bool &lessRandom)
• static bool **isUseSymbols** (const bool &defaultValue=QVariant().toBool())
• static void **setUseSymbols** (const bool &useSymbols)
• static PasswordConfiguration **getPasswordConfiguration** ()
• static void **setPasswordConfiguration** (const PasswordConfiguration &config)
• static void **setPasswordLength** (const int &passwordLength)
• static void **setPasswordCharsselection** (const int &passwordCharsselection)
• static void **setPasswordChars** (const QString &passwordChars)
• static bool **isUseTrayIcon** (const bool &defaultValue=QVariant().toBool())
• static void **setUseTrayIcon** (const bool &useTrayIcon)
• static bool **isHideOnClose** (const bool &defaultValue=QVariant().toBool())
• static void **setHideOnClose** (const bool &hideOnClose)
• static bool **isStartMinimized** (const bool &defaultValue=QVariant().toBool())
• static void **setStartMinimized** (const bool &startMinimized)
• static bool **isAlwaysOnTop** (const bool &defaultValue=QVariant().toBool())
• static void **setAlwaysOnTop** (const bool &alwaysOnTop)
• static bool **isAutoPull** (const bool &defaultValue=QVariant().toBool())
• static void **setAutoPull** (const bool &autoPull)
• static bool **isAutoPush** (const bool &defaultValue=QVariant().toBool())
• static void **setAutoPush** (const bool &autoPush)
• static void setAutoPush (const bool &autoPush)
• static QString getPassTemplate (const QString &defaultValue=QVariant().toString())
• static void setPassTemplate (const QString &passTemplate)
• static bool isUseTemplate (const bool &defaultValue=QVariant().toBool())
• static void setUseTemplate (const bool &useTemplate)
• static bool isTemplateAllFields (const bool &defaultValue=QVariant().toBool())
• static void setTemplateAllFields (const bool &templateAllFields)
• static QHash<QString, QString> getProfiles ()
• static void setProfiles (const QHash<QString, QString> &profiles)
• static Pass * getPass ()
• static RealPass * getRealPass ()
• static ImitatePass * getImitatePass ()

12.20.1 Detailed Description

Singleton that stores qtpass' settings, saves and loads config.

Definition at line 21 of file qtpasssettings.h.

12.20.2 Member Function Documentation

12.20.2.1 getAutoclearPanelSeconds()

```
int QtPassSettings::getAutoclearPanelSeconds (const int &defaultValue = QVariant().toInt()) [static]
```

Definition at line 216 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getAutoclearPanelSeconds QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::config QtPass::init MainWindow::MainWindow ConfigDialog::ConfigDialog
```

Generated by Doxygen
12.20.2.2 `getAutoclearSeconds()`

```cpp
int QtPassSettings::getAutoclearSeconds {
    const int & defaultValue = QVariant().toInt() } [static]
```

Definition at line 196 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

Here is the caller graph for this function:

![Caller Graph](caller_graph.png)

12.20.2.3 `getClipBoardType()`

```cpp
Enums::clipBoardType QtPassSettings::getClipBoardType {
    const Enums::clipBoardType & defaultValue = Enums::CLIPBOARD_NEVER } [static]
```

Definition at line 171 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

Here is the caller graph for this function:

![Caller Graph](caller_graph.png)
12.20.2.4 getClipBoardTypeRaw()

```cpp
int QtPassSettings::getClipBoardTypeRaw (const Enums::clipBoardType & defaultValue = Enums::CLIPBOARD_NEVER) [static]
```

Definition at line 163 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for getClipBoardTypeRaw()]

Here is the caller graph for this function:

![Caller Graph for getClipBoardTypeRaw()]

12.20.2.5 getGeometry()

```cpp
QByteArray QtPassSettings::getGeometry (const QByteArray & defaultValue = QVariant().toByteArray()) [static]
```

Definition at line 108 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for getGeometry()]

Here is the caller graph for this function:

![Caller Graph for getGeometry()]

Generated by Doxygen
12.20.2.6 getGitExecutable()

QString QtPassSettings::getGitExecutable (  
    const QString & defaultValue = QVariant().toString() )  [static]

Definition at line 305 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getGitExecutable -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow
ImitatePass::Init
ImitatePass::GitPull
QtPassSettings::initExecutables
QtPass::init
ConfigDialog::ConfigDialog
```

12.20.2.7 getGpgExecutable()

QString QtPassSettings::getGpgExecutable (  
    const QString & defaultValue = QVariant().toString() )  [static]

Definition at line 314 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getGpgExecutable -> QtPassSettings::getInstance
```

Generated by Doxygen
12.20 QtPassSettings Class Reference

Here is the caller graph for this function:

```plaintext
isVisible (MainWindow) → QtPassSettings::getInstance → QtPassSettings::getGpgHome

Here is the call graph for this function:

```plaintext
QtPassSettings::getGpgHome (const QString &defaultValue = QVariant().toString()) [static]

Definition at line 332 of file qtpasssettings.cpp.

Here is the call graph for this function:

```plaintext
isVisible (MainWindow) → QtPassSettings::getInstance → QtPassSettings::getGpgHome

Here is the caller graph for this function:

```plaintext
MainWindow::keyPressEvent (PasswordDialog::Password) → QtPass::init → MainWindow::MainWindow

12.20.2.8 getGpgHome()

QString QtPassSettings::getGpgHome {
    const QString &defaultValue = QVariant().toString()) [static]

Definition at line 332 of file qtpasssettings.cpp.

Here is the call graph for this function:

```plaintext
isVisible (MainWindow) → QtPassSettings::getInstance → QtPassSettings::getGpgHome

Here is the caller graph for this function:

```plaintext
MainWindow::keyPressEvent (PasswordDialog::Password) → QtPass::init → MainWindow::MainWindow

Generated by Doxygen
12.20.2.9 getImitatePass()

ImitatePass * QtPassSettings::getImitatePass ( ) [static]

Definition at line 562 of file qtpasssettings.cpp.

Here is the caller graph for this function:

12.20.2.10 getInstance()

QtPassSettings * QtPassSettings::getInstance ( ) [static]

Definition at line 19 of file qtpasssettings.cpp.
Here is the caller graph for this function:
12.20.2.12 getPassExecutable()

QString QtPassSettings::getPassExecutable (  
const QString & defaultValue = QVariant().toString() ) [static]

Definition at line 296 of file qtpasssettings.cpp.

Here is the call graph for this function:

QtPassSettings::getInstance -> QtPassSettings::getPassExecutable
Here is the caller graph for this function:

![Caller Graph]

### 12.20.2.13 getPassStore()

```cpp
QString QtPassSettings::getPassStore ( const QString & defaultValue = QVariant().toString() ) [static]
```

Definition at line 254 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

Generated by Doxygen
12.20.2.14 getPassTemplate()

QString QtPassSettings::getPassTemplate(
    const QString &defaultValue = QVariant().toString()) [static]

Definition at line 529 of file qtpasssettings.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.20.2.15 getPasswordConfiguration()

PasswordConfiguration QtPassSettings::getPasswordConfiguration() [static]

Definition at line 35 of file qtpasssettings.cpp.

Here is the call graph for this function:
Here is the caller graph for this function:

```
QtPassSettings::getPassword
Configuration
ConfigDialog::ConfigDialog
PasswordDialog::Password
Dialog
```

#### 12.20.2.16 `getPos()`

```cpp
QPoint QtPassSettings::getPos (  
    const QPoint & defaultValue = QVariant().toPoint()  )  [static]
```

Definition at line 126 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getPos
QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::restoreWindow
QtPassSettings::getPos
```
12.20.2.17  getProfile()

QString QtPassSettings::getProfile ( const QString & defaultValue = QVariant().toString() ) [static]

Definition at line 374 of file qtpasssettings.cpp.

Here is the call graph for this function:

QtPassSettings::getProfile  QtPassSettings::getInstance

Here is the caller graph for this function:

ConfigDialog::ConfigDialog  QtPassSettings::getProfile

12.20.2.18  getProfiles()

QHash< QString, QString > QtPassSettings::getProfiles ( ) [static]

Definition at line 61 of file qtpasssettings.cpp.

Here is the call graph for this function:

QtPassSettings::getProfiles  QtPassSettings::getInstance
Here is the caller graph for this function:

![Caller Graph]

### 12.20.2.19 getPwgenExecutable()

```cpp
QString QtPassSettings::getPwgenExecutable (const QString &defaultValue = QVariant().toString()) [static]
```

Definition at line 323 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

Generated by Doxygen
12.20.2.20  getQrencodeExecutable()

QString QtPassSettings::getQrencodeExecutable (const QString &defaultValue = QVariant().toString()) [static]

Definition at line 408 of file qtpasssettings.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.20.2.21  getRealPass()

RealPass ∗ QtPassSettings::getRealPass () [static]

Definition at line 557 of file qtpasssettings.cpp.

Here is the caller graph for this function:
12.20.2.22  getSavestate()

QByteArray QtPassSettings::getSavestate (  
    const QByteArray & defaultValue = QVariant().toByteArray () )  [static]

Definition at line 117 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getSavestate -------------------------- QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::restoreWindow ------------------------------- QtPassSettings::getSavestate
```

12.20.2.23  getSize()

QSize QtPassSettings::getSize (  
    const QSize & defaultValue = QVariant().toSize () )  [static]

Definition at line 133 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getSize ------------------------------ QtPassSettings::getInstance
```

Generated by Doxygen
Here is the caller graph for this function:

```
MainWindow::restoreWindow  QtPassSettings::getSize
```

### 12.20.2.24 getVersion()

```cpp
QString QtPassSettings::getVersion (  
    const QString & defaultValue = QVariant().toString()  ) [static]
```

Definition at line 99 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getVersion  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow  QtPass::init  QtPassSettings::getVersion
```

### 12.20.2.25 getWebDavPassword()

```cpp
QString QtPassSettings::getWebDavPassword (  
    const QString & defaultValue = QVariant().toString()  ) [static]
```

Definition at line 365 of file qtpasssettings.cpp.
Here is the call graph for this function:

```
QtPassSettings::getWebDav
Password

QtPassSettings::getInstance
```

### 12.20.2.26 getWebDavUrl()

```cpp
QString QtPassSettings::getWebDavUrl ( 
    const QString & defaultValue = QVariant().toString() ) [static]
```

Definition at line 347 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getWebDavUrl

QtPassSettings::getInstance
```

### 12.20.2.27 getWebDavUser()

```cpp
QString QtPassSettings::getWebDavUser ( 
    const QString & defaultValue = QVariant().toString() ) [static]
```

Definition at line 356 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getWebDavUser

QtPassSettings::getInstance
```
12.20.2.28 initExecutables()

```cpp
void QtPassSettings::initExecutables ( ) [static]
```

Definition at line 279 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for initExecutables()](call_graph_initExecutables.png)

Here is the caller graph for this function:

![Caller Graph for initExecutables()](caller_graph_initExecutables.png)

12.20.2.29 isAddGPGId()

```cpp
bool QtPassSettings::isAddGPGId ( const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 245 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for isAddGPGId()](call_graph_isAddGPGId.png)
Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog
QtPassSettings::isAddGPGid
ImitatePass::Init
```

### 12.20.2.30 isAlwaysOnTop()

```cpp
bool QtPassSettings::isAlwaysOnTop {
    const bool & defaultValue = QVariant().toBool()  [static]
}
```

Definition at line 502 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isAlwaysOnTop
QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::config
ConfigDialog::ConfigDialog
MainWindow::restoreWindow
QtPass::init
MainWindow::MainWindow
```

Generated by Doxygen
12.20.2.31 isAutoPull()

bool QtPassSettings::isAutoPull (  
    const bool & defaultValue = QVariant().toBool() ) [static]

Definition at line 511 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isAutoPull  
                           |  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog  
        |  QtPassSettings::isAutoPull
```

12.20.2.32 isAutoPush()

bool QtPassSettings::isAutoPush (  
    const bool & defaultValue = QVariant().toBool() ) [static]

Definition at line 520 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isAutoPush  
                           |  QtPassSettings::getInstance
```

Generated by Doxygen
Here is the caller graph for this function:

![Caller Graph]

### 12.20.2.33 isAvoidCapitals()

```c++
bool QtPassSettings::isAvoidCapitals ( const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 427 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]
### 12.20.2.34 isAvoidNumbers()

```cpp
bool QtPassSettings::isAvoidNumbers (const bool &defaultValue = QVariant().toBool()) [static]
```

Definition at line 436 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for isAvoidNumbers()](call_graph1.png)

Here is the caller graph for this function:

![Caller Graph for isAvoidNumbers()](caller_graph1.png)

### 12.20.2.35 isHideContent()

```cpp
bool QtPassSettings::isHideContent (const bool &defaultValue = QVariant().toBool()) [static]
```

Definition at line 236 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for isHideContent()](call_graph2.png)

Here is the caller graph for this function:

![Caller Graph for isHideContent()](caller_graph2.png)
Here is the caller graph for this function:

![Caller Graph]

12.20.2.36 isHideOnClose()

```cpp
bool QtPassSettings::isHideOnClose ( const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 484 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]
12.20.2.37  isHidePassword()

```cpp
bool QtPassSettings::isHidePassword (const bool &defaultValue = QVariant().toBool()) [static]
```

Definition at line 227 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isHidePassword -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> QtPassSettings::isHidePassword
```

12.20.2.38  isLessRandom()

```cpp
bool QtPassSettings::isLessRandom (const bool &defaultValue = QVariant().toBool()) [static]
```

Definition at line 445 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isLessRandom -> QtPassSettings::getInstance
```
Here is the caller graph for this function:

![Caller Graph](image)

### 12.20.2.39 isMaximized()

```cpp
bool QtPassSettings::isMaximized (
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 140 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph](image)

Here is the caller graph for this function:

![Caller Graph](image)
12.20.2.40  **isStartMinimized()**

```cpp
bool QtPassSettings::isStartMinimized (
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 493 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isStartMinimized  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog  QtPassSettings::isStartMinimized
MainWindow::restoreWindow
```

12.20.2.41  **isTemplateAllFields()**

```cpp
bool QtPassSettings::isTemplateAllFields (
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 547 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isTemplateAllFields  QtPassSettings::getInstance
```
12.20 QtPassSettings Class Reference

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog
MainWindow::passShowHandler
PasswordDialog::PasswordDialog
```

12.20.2.42 isUseAutoclear()

```c
bool QtPassSettings::isUseAutoclear ( const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 187 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseAutoclear
QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::passShowHandler
QtPass::copyTextToClipboard
```

Generated by Doxygen
12.20.2.43 isUseAutoclearPanel()

```cpp
bool QtPassSettings::isUseAutoclearPanel (
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 206 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseAutoclear Panel
    QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog
MainWindow::passOtpHandler
MainWindow::passShowHandler
```

12.20.2.44 isUseGit()

```cpp
bool QtPassSettings::isUseGit ( 
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 383 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseGit
    QtPassSettings::getInstance
```
Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog
ImitatePass::Copy
ImitatePass::GitPush
ImitatePass::Init
ImitatePass::Move
MainWindow::onPush
ImitatePass::Remove
QtPassSettings::isUseGit
```

### 12.20.2.45 isUseOtp()

```cpp
bool QtPassSettings::isUseOtp(
    const bool & defaultValue = QVariant().toBool()) [static]
```

Definition at line 390 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseOtp
QtPassSettings::getInstance
```
Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> QtPassSettings::isUseOtp
```

12.20.2.46 isUsePass()

```cpp
bool QtPassSettings::isUsePass (const bool &defaultValue = QVariant().toBool()) [static]
```

Definition at line 149 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUsePass -> QtPassSettings::getInstance
```

Here is the caller graph for this function:
12.20.2.47  isUsePwgen()

```cpp
bool QtPassSettings::isUsePwgen (
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 418 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUsePwgen -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> Pass::Generate_b -> QtPassSettings::isUsePwgen
PasswordDialog::PasswordDialog -> QtPassSettings::isUsePwgen
```

12.20.2.48  isUseQrencode()

```cpp
bool QtPassSettings::isUseQrencode {
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 398 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseQrencode -> QtPassSettings::getInstance
```
Here is the caller graph for this function:

![Caller Graph for isUseQrencode()](image)

12.20.2.49 **isUseSelection()**

```cpp
bool QtPassSettings::isUseSelection (const bool & defaultValue = QVariant().toBool()) [static]
```

Definition at line 178 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for isUseSelection()](image)

Here is the caller graph for this function:

![Caller Graph for isUseSelection()](image)

12.20.2.50 **isUseSymbols()**

```cpp
bool QtPassSettings::isUseSymbols (const bool & defaultValue = QVariant().toBool()) [static]
```

Definition at line 454 of file qtpasssettings.cpp.
Here is the call graph for this function:

```
QtPassSettings::isUseSymbols  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog  QtPassSettings::isUseSymbols
Pass::Generate_b
```

### 12.20.2.51 isUseTemplate()

```cpp
bool QtPassSettings::isUseTemplate ( 
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 538 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseTemplate  QtPassSettings::getInstance
```
12.20.2.52 isUseTrayIcon()

```cpp
bool QtPassSettings::isUseTrayIcon ( const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 475 of file qtpasssettings.cpp.

Here is the call graph for this function:
12.20.2.53  **isUseWebDav()**

```cpp
bool QtPassSettings::isUseWebDav (
    const bool & defaultValue = QVariant().toBool() ) [static]
```

Definition at line 338 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for isUseWebDav()](call_graph_isUseWebDav.png)

Here is the caller graph for this function:

![Caller graph for isUseWebDav()](caller_graph_isUseWebDav.png)

12.20.2.54  **setAddGPGId()**

```cpp
void QtPassSettings::setAddGPGId ( 
    const bool & addGPGId ) [static]
```

Definition at line 250 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setAddGPGId()](call_graph_setAddGPGId.png)
12.20.2.55  setAlwaysOnTop()

```c
void QtPassSettings::setAlwaysOnTop (
    const bool & alwaysOnTop ) [static]
```

Definition at line 507 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setAlwaysOnTop     QtPassSettings::getInstance
```

12.20.2.56  setAutoclearPanelSeconds()

```c
void QtPassSettings::setAutoclearPanelSeconds (
    const int & autoClearPanelSeconds ) [static]
```

Definition at line 221 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setAutoclearPanelSeconds     QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow     QtPass::init     QtPassSettings::setAutoclearPanelSeconds
```
12.20.2.57  setAutoclearSeconds()

void QtPassSettings::setAutoclearSeconds (
    const int & autoClearSeconds ) [static]

Definition at line 201 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setAutoclearSeconds()]

Here is the caller graph for this function:

![Caller Graph for setAutoclearSeconds()]

12.20.2.58  setAutoPull()

void QtPassSettings::setAutoPull (  
    const bool & autoPull ) [static]

Definition at line 516 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setAutoPull()]

Generated by Doxygen
12.20.2.59 setAutoPush()

```cpp
void QtPassSettings::setAutoPush (const bool & autoPush) [static]
```

Definition at line 525 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setAutoPush    QtPassSettings::getInstance
```

12.20.2.60 setAvoidCapitals()

```cpp
void QtPassSettings::setAvoidCapitals (const bool & avoidCapitals) [static]
```

Definition at line 432 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setAvoidCapitals    QtPassSettings::getInstance
```

12.20.2.61 setAvoidNumbers()

```cpp
void QtPassSettings::setAvoidNumbers (const bool & avoidNumbers) [static]
```

Definition at line 441 of file qtpasssettings.cpp.
12.20 QtPassSettings Class Reference

Here is the call graph for this function:

```
QtPassSettings::setAvoidNumbers
               ^
               |                        QtPassSettings::getInstance
```

12.20.2.62 setClipBoardType()

```cpp
void QtPassSettings::setClipBoardType(
    const int & clipBoardType) [static]
```

Definition at line 174 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setClipBoardType
               ^
               |                        QtPassSettings::getInstance
```

12.20.2.63 setGeometry()

```cpp
void QtPassSettings::setGeometry(
    const QByteArray & geometry) [static]
```

Definition at line 113 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setGeometry
               ^
               |                        QtPassSettings::getInstance
```
Here is the caller graph for this function:

```
MainWindow::closeEvent → QtPassSettings::setGeometry
```

### 12.20.2.64 setGitExecutable()

```cpp
void QtPassSettings::setGitExecutable (const QString &gitExecutable) [static]
```

Definition at line 310 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setGitExecutable → QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow → ConfigDialog → QtPass : init
QtPass : init → QtPassSettings : initExecutables
QtPassSettings : initExecutables → QtPassSettings : setGitExecutable
```

### 12.20.2.65 setGpgExecutable()

```cpp
void QtPassSettings::setGpgExecutable (const QString &gpgExecutable) [static]
```

Definition at line 319 of file qtpasssettings.cpp.
Here is the call graph for this function:

![Call graph for setGpgExecutable function]

Here is the caller graph for this function:

![Caller graph for setGpgExecutable function]

### 12.20.2.66 setHideContent()

```cpp
void QtPassSettings::setHideContent (const bool & hideContent) [static]
```

Definition at line 241 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setHideContent function]

### 12.20.2.67 setHideOnClose()

```cpp
void QtPassSettings::setHideOnClose (const bool & hideOnClose) [static]
```

Definition at line 489 of file qtpasssettings.cpp.

---

Generated by Doxygen
Here is the call graph for this function:

12.20.2.68 setHidePassword()

```cpp
void QtPassSettings::setHidePassword ( const bool & hidePassword ) [static]
```

Definition at line 232 of file qtpasssettings.cpp.

Here is the call graph for this function:

12.20.2.69 setLessRandom()

```cpp
void QtPassSettings::setLessRandom ( const bool & lessRandom ) [static]
```

Definition at line 450 of file qtpasssettings.cpp.

Here is the call graph for this function:
12.20.2.70 `setMaximized()`

```cpp
void QtPassSettings::setMaximized (const bool &maximized) [static]
```

Definition at line 145 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setMaximized -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::closeEvent -> QtPassSettings::setMaximized
```

12.20.2.71 `setPassExecutable()`

```cpp
void QtPassSettings::setPassExecutable (const QString &passExecutable ) [static]
```

Definition at line 301 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setPassExecutable -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
QtPass::init -> MainWindow::MainWindow -> QtPassSettings::setPassExecutable
```

Generated by Doxygen
12.20.2.72  setPassStore()

```cpp
void QtPassSettings::setPassStore (  
    const QString & passStore  ) [static]
```

Definition at line 275 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setPassStore()](image)

Here is the caller graph for this function:

![Caller Graph for setPassStore()](image)

12.20.2.73  setPassTemplate()

```cpp
void QtPassSettings::setPassTemplate (  
    const QString & passTemplate  ) [static]
```

Definition at line 534 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setPassTemplate()](image)

Here is the caller graph for this function:

![Caller Graph for setPassTemplate()](image)
12.20.2.74 setPasswordChars()

void QtPassSettings::setPasswordChars (  
    const QString & passwordChars ) [static]

Definition at line 471 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setPasswordChars()]

12.20.2.75 setPasswordCharsSelection()

void QtPassSettings::setPasswordCharsSelection (  
    const int & passwordCharsSelection ) [static]

Definition at line 466 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setPasswordCharsSelection()]

12.20.2.76 setPasswordConfiguration()

void QtPassSettings::setPasswordConfiguration (  
    const PasswordConfiguration & config ) [static]

Definition at line 52 of file qtpasssettings.cpp.

Generated by Doxygen
Here is the call graph for this function:

12.20.2.77 setPasswordLength()

```cpp
void QtPassSettings::setPasswordLength (
    const int & passwordLength ) [static]
```

Definition at line 463 of file qtpasssettings.cpp.

Here is the call graph for this function:

12.20.2.78 setPos()

```cpp
void QtPassSettings::setPos ( 
    const QPoint & pos ) [static]
```

Definition at line 129 of file qtpasssettings.cpp.

Here is the call graph for this function:
Here is the caller graph for this function:

![Caller Graph]

### 12.20.2.79 setProfile()

```cpp
definition QtPassSettings::setProfile {
    const QString & profile } [static]
```

Definition at line 379 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph]

### 12.20.2.80 setProfiles()

```cpp
definition QtPassSettings::setProfiles {
    const QHash< QString, QString > & profiles } [static]
```

Definition at line 75 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph]
12.20.2.81  setPwgenExecutable()

void QtPassSettings::setPwgenExecutable (  
    const QString & pwgenExecutable )  [static]

Definition at line 328 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setPwgenExecutable()](image)

Here is the caller graph for this function:

![Caller Graph for setPwgenExecutable()](image)

12.20.2.82  setQrencodeExecutable()

void QtPassSettings::setQrencodeExecutable (  
    const QString & qrencodeExecutable )  [static]

Definition at line 413 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setQrencodeExecutable()](image)
12.20.2.83  setSavestate()

```cpp
void QtPassSettings::setSavestate (
    const QByteArray & saveState ) [static]
```

Definition at line 122 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setSavestate -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::closeEvent -> QtPassSettings::setSavestate
```

12.20.2.84  setSize()

```cpp
void QtPassSettings::setSize (
    const QSize & size ) [static]
```

Definition at line 136 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setSize -> QtPassSettings::getInstance
```

Generated by Doxygen
Here is the caller graph for this function:

```
MainWindow::closeEvent  QtPassSettings::setSize
```

### 12.20.2.85 `setStartMinimized()`

```cpp
definition QtPassSettings::setStartMinimized (const bool & startMinimized) [static]
```

Definition at line 498 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setStartMinimized  QtPassSettings::getInstance
```

### 12.20.2.86 `setTemplateAllFields()`

```cpp
definition QtPassSettings::setTemplateAllFields (const bool & templateAllFields) [static]
```

Definition at line 552 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setTemplateAllFields  QtPassSettings::getInstance
```
12.20.2.87 setUseAutoclear()

```cpp
void QtPassSettings::setUseAutoclear ( const bool & useAutoclear ) [static]
```

Definition at line 192 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseAutoclear()](image)

12.20.2.88 setUseAutoclearPanel()

```cpp
void QtPassSettings::setUseAutoclearPanel ( const bool & useAutoclearPanel ) [static]
```

Definition at line 211 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseAutoclearPanel()](image)

12.20.2.89 setUseGit()

```cpp
void QtPassSettings::setUseGit ( const bool & useGit ) [static]
```

Definition at line 386 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseGit()](image)
12.20.2.90 setUseOtp()

```cpp
void QtPassSettings::setUseOtp (
    const bool & useOtp ) [static]
```

Definition at line 394 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseOtp()](call_graph_setUseOtp.png)

12.20.2.91 setUsePass()

```cpp
void QtPassSettings::setUsePass (
    const bool & usePass ) [static]
```

Definition at line 154 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUsePass()](call_graph_setUsePass.png)

Here is the caller graph for this function:

![Caller graph for setUsePass()](caller_graph_setUsePass.png)
12.20.2.92  setUsePwgen()

void QtPassSettings::setUsePwgen (  
        const bool & usePwgen ) [static]

Definition at line 423 of file qtpasssettings.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.20.2.93  setUseQrencode()

void QtPassSettings::setUseQrencode (  
        const bool & useQrencode ) [static]

Definition at line 404 of file qtpasssettings.cpp.

Here is the call graph for this function:
12.20.2.94 setUseSelection()

```cpp
void QtPassSettings::setUseSelection (const bool & useSelection) [static]
```

Definition at line 183 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseSelection()](#)

12.20.2.95 setUseSymbols()

```cpp
void QtPassSettings::setUseSymbols (const bool & useSymbols) [static]
```

Definition at line 459 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseSymbols()](#)

12.20.2.96 setUseTemplate()

```cpp
void QtPassSettings::setUseTemplate (const bool & useTemplate) [static]
```

Definition at line 543 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call graph for setUseTemplate()](#)
12.20.2.97 setUseTrayIcon()

```cpp
void QtPassSettings::setUseTrayIcon (  
    const bool & useTrayIcon ) [static]
```

Definition at line 480 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setUseTrayIcon()](image)

12.20.2.98 setUseWebDav()

```cpp
void QtPassSettings::setUseWebDav (  
    const bool & useWebDav ) [static]
```

Definition at line 343 of file qtpasssettings.cpp.

Here is the call graph for this function:

![Call Graph for setUseWebDav()](image)

12.20.2.99 setVersion()

```cpp
void QtPassSettings::setVersion (  
    const QString & version ) [static]
```

Definition at line 104 of file qtpasssettings.cpp.
Here is the call graph for this function:

```
QtPassSettings::setVersion  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow  QtPass::init  QtPassSettings::setVersion
```

12.20.2.100  setWebDavPassword()

```c
void QtPassSettings::setWebDavPassword ( 
    const QString & webDavPassword ) [static]
```

Definition at line 370 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setWebDavPassword  QtPassSettings::getInstance
```

12.20.2.101  setWebDavUrl()

```c
void QtPassSettings::setWebDavUrl ( 
    const QString & webDavUrl ) [static]
```

Definition at line 352 of file qtpasssettings.cpp.
Here is the call graph for this function:

```
QtPassSettings::setWebDavUrl -> QtPassSettings::getInstance
```

12.20.2.102  setWebDavUser()

```cpp
void QtPassSettings::setWebDavUser (
    const QString & webDavUser ) [static]
```

Definition at line 361 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setWebDavUser -> QtPassSettings::getInstance
```

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

- `src/qtpasssettings.h`
- `src/qtpasssettings.cpp`

12.21  RealPass Class Reference

Wrapper for executing pass to handle the password-store.

```
#include <realpass.h>
```
Inheritance diagram for RealPass:

```
QObject
         ↑
   Pass
         ↑
RealPass
```

Collaboration diagram for RealPass:

```
QObject
         ↑
   Executor
         ↓
exec
     ↑
   Pass
         ↑
RealPass
```

**Public Member Functions**

- `RealPass ()`
- virtual `~RealPass ()`
- virtual void `GitInit () Q_DECL_OVERRIDE`

  *RealPass::GitInit* pass *git init* wrapper.
12.21 RealPass Class Reference

• virtual void GitPull () Q_DECL_OVERRIDE
  
  RealPass::GitPull pass git pull wrapper.
• virtual void GitPull_b () Q_DECL_OVERRIDE
  
  RealPass::GitPull pass git pull wrapper which blocks until process finishes.
• virtual void GitPush () Q_DECL_OVERRIDE
  
  RealPass::GitPush pass git push wrapper.
• virtual void Show (QString file) Q_DECL_OVERRIDE
  
  RealPass::Show pass show.
• virtual void OtpGenerate (QString file) Q_DECL_OVERRIDE
  
  RealPass::OtpGenerate pass otp.
• virtual void Insert (QString file, QString newValue, bool overwrite=false) Q_DECL_OVERRIDE
  
  RealPass::Insert pass insert.
• virtual void Remove (QString file, bool isDir=false) Q_DECL_OVERRIDE
  
  RealPass::Remove pass remove wrapper.
• virtual void Init (QString path, const QList<UserInfo> &users) Q_DECL_OVERRIDE
  
  RealPass::Init initialize pass repository.
• void Move (const QString src, const QString dest, const bool force=false) Q_DECL_OVERRIDE
  
  RealPass::Move move a file (or folder)
• void Copy (const QString src, const QString dest, const bool force=false) Q_DECL_OVERRIDE
  
  RealPass::Copy copy a file (or folder)

Additional Inherited Members

12.21.1 Detailed Description

Wrapper for executing pass to handle the password-store.

Definition at line 10 of file realpass.h.

12.21.2 Constructor & Destructor Documentation

12.21.2.1 RealPass()

RealPass::RealPass ( ) [default]

12.21.2.2 ~RealPass()

virtual RealPass::~RealPass ( ) [inline], [virtual]

Definition at line 17 of file realpass.h.
12.21.3  Member Function Documentation

12.21.3.1  Copy()

    void RealPass::Copy (  
        const QString src,  
        const QString dest,  
        const bool force = false  )  [virtual]

RealPass::Copy copy a file (or folder)

Parameters

<table>
<thead>
<tr>
<th>src</th>
<th>source file or folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>dest</td>
<td>destination file or folder</td>
</tr>
<tr>
<td>force</td>
<td>overwrite</td>
</tr>
</tbody>
</table>

Implements Pass.

Definition at line 141 of file realpass.cpp.

Here is the call graph for this function:

```
RealPass::Copy -> QtPassSettings::getPassStore -> QtPassSettings::getInstance
```

12.21.3.2  GitInit()

    void RealPass::GitInit ( )  [virtual]

RealPass::GitInit pass git init wrapper.

Implements Pass.

Definition at line 15 of file realpass.cpp.
12.21.3.3 GitPull()

```cpp
void RealPass::GitPull() [virtual]
```

RealPass::GitPull pass git pull wrapper.
Implements Pass.
Definition at line 28 of file realpass.cpp.

12.21.3.4 GitPull_b()

```cpp
void RealPass::GitPull_b() [virtual]
```

RealPass::GitInit pass git pull wrapper which blocks until process finishes.
Implements Pass.
Definition at line 21 of file realpass.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

12.21.3.5 GitPush()

```cpp
void RealPass::GitPush() [virtual]
```

RealPass::GitPush pass git push wrapper.
Implements Pass.
Definition at line 33 of file realpass.cpp.

12.21.3.6 Init()

```cpp
void RealPass::Init(QString path,
                    const QList<UserInfo>& users) [virtual]
```

RealPass::Init initialize pass repository.

Generated by Doxygen
### Parameters

<table>
<thead>
<tr>
<th>path</th>
<th>Absolute path to new password-store</th>
</tr>
</thead>
<tbody>
<tr>
<td>users</td>
<td>list of users with ability to decrypt new password-store</td>
</tr>
</tbody>
</table>

Implements **Pass**.

Definition at line 80 of file realpass.cpp.

Here is the call graph for this function:

```
RealPass::Init -> QtPassSettings::getPassStore -> QtPassSettings::getInstance
```

#### 12.21.3.7 Insert()

```cpp
void RealPass::Insert ( 
    QString file, 
    QString newValue, 
    bool overwrite = false ) [virtual]
```

**RealPass::Insert** pass insert.

Implements **Pass**.

Definition at line 59 of file realpass.cpp.

#### 12.21.3.8 Move()

```cpp
void RealPass::Move ( 
    const QString src, 
    const QString dest, 
    const bool force = false ) [virtual]
```

**RealPass::Move** move a file (or folder)

<table>
<thead>
<tr>
<th>src</th>
<th>source file or folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>dest</td>
<td>destination file or folder</td>
</tr>
<tr>
<td>force</td>
<td>overwrite</td>
</tr>
</tbody>
</table>
12.21 RealPass Class Reference

Implements Pass.

Definition at line 100 of file realpass.cpp.

Here is the call graph for this function:

```
RealPass::Move QtPassSettings::getPassStore QtPassSettings::getInstance
```

12.21.3.9 OtpGenerate()

```c
void RealPass::OtpGenerate ( 
    QString file ) [virtual]
```

RealPass::OtpGenerate pass otp.

Parameters

- **file** file containig OTP uri

Implements Pass.

Definition at line 52 of file realpass.cpp.

12.21.3.10 Remove()

```c
void RealPass::Remove ( 
    QString file, 
    bool isDir = false ) [virtual]
```

RealPass::Remove pass remove wrapper.

Implements Pass.

Definition at line 70 of file realpass.cpp.

12.21.3.11 Show()

```c
void RealPass::Show ( 
    QString file ) [virtual]
```

RealPass::Show pass show.

Generated by Doxygen
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>file to decrypt</td>
</tr>
</tbody>
</table>

Returns

if block is set, returns exit status of internal decryption process otherwise returns QProcess::NormalExit

Implements Pass.

Definition at line 44 of file realpass.cpp.

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

- src/realpass.h
- src/realpass.cpp

### 12.22 SettingsConstants Class Reference

Table for the naming of configuration items.

```cpp
#include <settingsconstants.h>
```

**Static Public Attributes**

- static const QString version = "version"
- static const QString groupMainwindow = "mainwindow"
- static const QString geometry
- static const QString savestate
- static const QString pos
- static const QString size
- static const QString splitterLeft
- static const QString splitterRight
- static const QString maximized
- static const QString usePass = "usePass"
- static const QString useAutoclear = "useAutoclear"
- static const QString useSelection = "useSelection"
- static const QString autoclearSeconds = "autoclearSeconds"
- static const QString useAutoclearPanel = "useAutoclearPanel"
- static const QString autoclearPanelSeconds
- static const QString hidePassword = "hidePassword"
- static const QString hideContent = "hideContent"
- static const QString addGPGId = "addGPGId"
- static const QString passStore = "passStore"
- static const QString passExecutable = "passExecutable"
- static const QString gitExecutable = "gitExecutable"
- static const QString gpgExecutable = "gpgExecutable"
- static const QString pwgenExecutable = "pwgenExecutable"
- static constantQString gpgHome = "gpgHome"
- static const QString useWebDav = "useWebDav"
12.22 SettingsConstants Class Reference

- static const QString webDavUrl = "webDavUrl"
- static const QString webDavUser = "webDavUser"
- static const QString webDavPassword = "webDavPassword"
- static const QString profile = "profile"
- static const QString groupProfiles = "profiles"
- static const QString useGit = "useGit"
- static const QString useOtp = "useOtp"
- static const QString useQrencode = "useQrencode"
- static const QString qrencodeExecutable = "qrencodeExecutable"
- static const QString useClipboard = "useClipboard"
- static const QString usePwgen = "usePwgen"
- static const QString avoidCapitals = "avoidCapitals"
- static const QString avoidNumbers = "avoidNumbers"
- static const QString lessRandom = "lessRandom"
- static const QString useSymbols = "useSymbols"
- static const QString passwordLength = "passwordLength"
- static const QString passwordChars = "passwordChars"
- static const QString useTrayIcon = "useTrayIcon"
- static const QString hideOnClose = "hideOnClose"
- static const QString startMinimized = "startMinimized"
- static const QString alwaysOnTop = "alwaysOnTop"
- static const QString autoPull = "autoPull"
- static const QString autoPush = "autoPush"
- static const QString passTemplate = "passTemplate"
- static const QString useTemplate = "useTemplate"
- static const QString templateAllFields = "templateAllFields"
- static const QString clipBoardType = "clipBoardType"

12.22.1 Detailed Description

Table for the naming of configuration items.

Definition at line 10 of file settingsconstants.h.

12.22.2 Member Data Documentation

12.22.2.1 addGPGId

const QString SettingsConstants::addGPGId = "addGPGId" [static]

Definition at line 29 of file settingsconstants.h.
12.22.2.2 alwaysOnTop

const QString SettingsConstants::alwaysOnTop = "alwaysOnTop"  [static]

Definition at line 58 of file settingsconstants.h.

12.22.2.3 autoclearPanelSeconds

const QString SettingsConstants::autoclearPanelSeconds  [static]

Initial value:
=  "autoclearPanelSeconds"

Definition at line 26 of file settingsconstants.h.

12.22.2.4 autoclearSeconds

const QString SettingsConstants::autoclearSeconds = "autoclearSeconds"  [static]

Definition at line 24 of file settingsconstants.h.

12.22.2.5 autoPull

const QString SettingsConstants::autoPull = "autoPull"  [static]

Definition at line 59 of file settingsconstants.h.

12.22.2.6 autoPush

const QString SettingsConstants::autoPush = "autoPush"  [static]

Definition at line 60 of file settingsconstants.h.

12.22.2.7 avoidCapitals

const QString SettingsConstants::avoidCapitals = "avoidCapitals"  [static]

Definition at line 48 of file settingsconstants.h.
12.22.2.8 avoidNumbers

const QString SettingsConstants::avoidNumbers = "avoidNumbers" [static]

Definition at line 49 of file settingsconstants.h.

12.22.2.9 clipBoardType

const QString SettingsConstants::clipBoardType = "clipBoardType" [static]

Definition at line 64 of file settingsconstants.h.

12.22.2.10 geometry

const QString SettingsConstants::geometry [static]

Initial value:
= SettingsConstants::groupMainwindow + "/geometry"

Definition at line 14 of file settingsconstants.h.

12.22.2.11 gitExecutable

const QString SettingsConstants::gitExecutable = "gitExecutable" [static]

Definition at line 32 of file settingsconstants.h.

12.22.2.12 gpgExecutable

const QString SettingsConstants::gpgExecutable = "gpgExecutable" [static]

Definition at line 33 of file settingsconstants.h.

12.22.2.13 gpgHome

const QString SettingsConstants::gpgHome = "gpgHome" [static]

Definition at line 35 of file settingsconstants.h.
12.22.2.14 groupMainWindow

const QString SettingsConstants::groupMainWindow = "mainwindow" [static]

Definition at line 13 of file settingsconstants.h.

12.22.2.15 groupProfiles

const QString SettingsConstants::groupProfiles = "profiles" [static]

Definition at line 41 of file settingsconstants.h.

12.22.2.16 hideContent

const QString SettingsConstants::hideContent = "hideContent" [static]

Definition at line 28 of file settingsconstants.h.

12.22.2.17 hideOnClose

const QString SettingsConstants::hideOnClose = "hideOnClose" [static]

Definition at line 56 of file settingsconstants.h.

12.22.2.18 hidePassword

const QString SettingsConstants::hidePassword = "hidePassword" [static]

Definition at line 27 of file settingsconstants.h.

12.22.2.19 lessRandom

const QString SettingsConstants::lessRandom = "lessRandom" [static]

Definition at line 50 of file settingsconstants.h.
12.22.2.20 maximized

const QString SettingsConstants::maximized [static]

Initial value:
```
  SettingsConstants::groupMainWindow + "/maximized"
```

Definition at line 20 of file settingsconstants.h.

12.22.2.21 passExecutable

const QString SettingsConstants::passExecutable = "passExecutable" [static]

Definition at line 31 of file settingsconstants.h.

12.22.2.22 passStore

const QString SettingsConstants::passStore = "passStore" [static]

Definition at line 30 of file settingsconstants.h.

12.22.2.23 passTemplate

const QString SettingsConstants::passTemplate = "passTemplate" [static]

Definition at line 61 of file settingsconstants.h.

12.22.2.24 passwordChars

const QString SettingsConstants::passwordChars = "passwordChars" [static]

Definition at line 54 of file settingsconstants.h.

12.22.2.25 passwordCharsselection

const QString SettingsConstants::passwordCharsselection [static]

Initial value:
```
  "passwordCharsselection"
```

Definition at line 53 of file settingsconstants.h.
12.22.26  passwordLength

const QString SettingsConstants::passwordLength = "passwordLength"  [static]

Definition at line 52 of file settingsconstants.h.

12.22.27  pos

const QString SettingsConstants::pos  [static]

Initial value:

= SettingsConstants::groupMainwindow + "/pos"

Definition at line 16 of file settingsconstants.h.

12.22.28  profile

const QString SettingsConstants::profile = "profile"  [static]

Definition at line 40 of file settingsconstants.h.

12.22.29  pwgenExecutable

const QString SettingsConstants::pwgenExecutable = "pwgenExecutable"  [static]

Definition at line 34 of file settingsconstants.h.

12.22.30  qrencodeExecutable

const QString SettingsConstants::qrencodeExecutable = "qrencodeExecutable"  [static]

Definition at line 45 of file settingsconstants.h.

12.22.31  savestate

const QString SettingsConstants::savestate  [static]

Initial value:

= SettingsConstants::groupMainwindow + "/savestate"

Definition at line 15 of file settingsconstants.h.
12.22.2.32 size

const QString SettingsConstants::size [static]

Initial value:

   SettingsConstants::groupMainWindow + "/size"

Definition at line 17 of file settingsconstants.h.

12.22.2.33 splitterLeft

const QString SettingsConstants::splitterLeft [static]

Initial value:

   SettingsConstants::groupMainWindow + "/splitterLeft"

Definition at line 18 of file settingsconstants.h.

12.22.2.34 splitterRight

const QString SettingsConstants::splitterRight [static]

Initial value:

   SettingsConstants::groupMainWindow + "/splitterRight"

Definition at line 19 of file settingsconstants.h.

12.22.2.35 startMinimized

const QString SettingsConstants::startMinimized = "startMinimized" [static]

Definition at line 57 of file settingsconstants.h.

12.22.2.36 templateAllFields

const QString SettingsConstants::templateAllFields = "templateAllFields" [static]

Definition at line 63 of file settingsconstants.h.
12.22.2.37 useAutoclear

const QString SettingsConstants::useAutoclear = "useAutoclear" [static]

Definition at line 22 of file settingsconstants.h.

12.22.2.38 useAutoclearPanel

const QString SettingsConstants::useAutoclearPanel = "useAutoclearPanel" [static]

Definition at line 25 of file settingsconstants.h.

12.22.2.39 useClipboard

const QString SettingsConstants::useClipboard = "useClipboard" [static]

Definition at line 46 of file settingsconstants.h.

12.22.2.40 useGit

const QString SettingsConstants::useGit = "useGit" [static]

Definition at line 42 of file settingsconstants.h.

12.22.2.41 useOtp

const QString SettingsConstants::useOtp = "useOtp" [static]

Definition at line 43 of file settingsconstants.h.

12.22.2.42 usePass

const QString SettingsConstants::usePass = "usePass" [static]

Definition at line 21 of file settingsconstants.h.
12.22.2.43  usePwgen

const QString SettingsConstants::usePwgen = "usePwgen"  [static]

Definition at line 47 of file settingsconstants.h.

12.22.2.44  useQrencode

const QString SettingsConstants::useQrencode = "useQrencode"  [static]

Definition at line 44 of file settingsconstants.h.

12.22.2.45  useSelection

const QString SettingsConstants::useSelection = "useSelection"  [static]

Definition at line 23 of file settingsconstants.h.

12.22.2.46  useSymbols

const QString SettingsConstants::useSymbols = "useSymbols"  [static]

Definition at line 51 of file settingsconstants.h.

12.22.2.47  useTemplate

const QString SettingsConstants::useTemplate = "useTemplate"  [static]

Definition at line 62 of file settingsconstants.h.

12.22.2.48  useTrayIcon

const QString SettingsConstants::useTrayIcon = "useTrayIcon"  [static]

Definition at line 55 of file settingsconstants.h.
12.22.2.49 useWebDav

const QString SettingsConstants::useWebDav = "useWebDav" [static]

Definition at line 36 of file settingsconstants.h.

12.22.2.50 version

const QString SettingsConstants::version = "version" [static]

Definition at line 12 of file settingsconstants.h.

12.22.2.51 webDavPassword

const QString SettingsConstants::webDavPassword = "webDavPassword" [static]

Definition at line 39 of file settingsconstants.h.

12.22.2.52 webDavUrl

const QString SettingsConstants::webDavUrl = "webDavUrl" [static]

Definition at line 37 of file settingsconstants.h.

12.22.2.53 webDavUser

const QString SettingsConstants::webDavUser = "webDavUser" [static]

Definition at line 38 of file settingsconstants.h.

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

- src/settingsconstants.h
- src/settingsconstants.cpp
#include <simpletransaction.h>

Inheritance diagram for simpleTransaction:

```
+-------------------------------+    +-----------------+
| simpleTransaction             |    | ImitatePass      |
|                               |    +----------------+---+
|                              |    |                 |
```

## Public Member Functions

- **simpleTransaction ()**
- **void transactionStart ()**
  
  `transactionStart` this function is used to mark start of the sequence of processes that shall be treated as one operation.
- **void transactionAdd (Enums::PROCESS)**
  
  `transactionAdd` If called after call to `transactionStart()` and before `transactionEnd()`, this method marks given process as next step in transaction. Otherwise it marks given process as the only step in transaction (it’s value is treated as transaction result).
- **void transactionEnd (Enums::PROCESS)**
  
  `transactionEnd` marks end of transaction
- **Enums::PROCESS transactionIsOver (Enums::PROCESS)**
  
  `transactionIsOver` checks whether currently finished process is last in current transaction

### 12.23.1 Detailed Description

Definition at line 7 of file simpletransaction.h.

### 12.23.2 Constructor & Destructor Documentation

#### 12.23.2.1 simpleTransaction()

```
simpleTransaction::simpleTransaction ( ) [inline]
```

Definition at line 13 of file simpletransaction.h.
12.23.3 Member Function Documentation

12.23.3.1 transactionAdd()

```c++
void simpleTransaction::transactionAdd ( Enums::PROCESS id )
```

transactionAdd if called after call to `transactionStart()` and before `transactionEnd()`, this method marks given process as next step in transaction. Otherwise it marks given process as the only step in transaction(it's value is treated as transaction result).

**Parameters**

| id | process that shall be treated as part of transaction |

Definition at line 25 of file `simpletransaction.cpp`.

12.23.3.2 transactionEnd()

```c++
void simpleTransaction::transactionEnd ( Enums::PROCESS pid )
```

transactionEnd marks end of transaction

**Parameters**

| pid | value that will be used as a result of transaction |

Definition at line 40 of file `simpletransaction.cpp`.

12.23.3.3 transactionIsOver()

```c++
PROCESS simpleTransaction::transactionIsOver ( Enums::PROCESS id )
```

transactionIsOver checks whether currently finished process is last in current transaction

**Parameters**

| id | process that shall be treated as part of transaction |

Generated by Doxygen
Returns

result of transaction as set by transactionAdd or transactionEnd if the transaction is over or \texttt{PROCESS::IN\rightarrow VALID} if it's not yet over

Parameters

<table>
<thead>
<tr>
<th>id</th>
</tr>
</thead>
</table>

Returns

Definition at line 58 of file simpletransaction.cpp.

### 12.23.3.4 transactionStart()

```cpp
void simpleTransaction::transactionStart ( )
```

transactionStart this function is used to mark start of the sequence of processes that shall be treated as one operation.

simpleTransaction::transactionStart

Definition at line 14 of file simpletransaction.cpp.

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

- src/simpletransaction.h
- src/simpletransaction.cpp

### 12.24 SingleApplication Class Reference

The \texttt{SingleApplication} class is used for commandline integration.

```cpp
#include <singleapplication.h>
```

Inheritance diagram for SingleApplication:
Collaboration diagram for SingleApplication:

![Collaboration Diagram](image)

### Public Slots
- void `receiveMessage` ()
  
  *SingleApplication::receiveMessage* we have received (a command line) message.

### Signals
- void `messageAvailable` (QString message)
  
  *messageAvailable notification from commandline*

### Public Member Functions
- `SingleApplication` (int &argc, char *argv[], QString uniqueKey)
  
  *SingleApplication::SingleApplication* this replaces the QApplication allowing for local socket based communications.
- bool `isRunning` ()
  
  *SingleApplication::isRunning* is there already a QtPass instance running, to check wether to be server or client.
- bool `sendMessage` (const QString &message)
  
  *SingleApplication::sendMessage* send a message (from commandline) to an already running QtPass instance.

### 12.24.1 Detailed Description

The SingleApplication class is used for commandline integration.

This class needs a bit of work or possibly replacement.

Definition at line 14 of file singleapplication.h.

### 12.24.2 Constructor & Destructor Documentation

#### 12.24.2.1 SingleApplication()

*SingleApplication::SingleApplication* (int &argc, char *argv[], QString uniqueKey)

*SingleApplication::SingleApplication* this replaces the QApplication allowing for local socket based communications.
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>argc</td>
<td></td>
</tr>
<tr>
<td>argv</td>
<td></td>
</tr>
<tr>
<td>uniqueKey</td>
<td></td>
</tr>
</tbody>
</table>

Definition at line 15 of file singleapplication.cpp.

Here is the call graph for this function:

#### 12.24.3 Member Function Documentation

#### 12.24.3.1 isRunning()

```cpp
def SingleApplication::isRunning()
```

**SingleApplication::isRunning** is there already a QtPass instance running, to check whether to be server or client.

**Returns**

Definition at line 64 of file singleapplication.cpp.

Here is the caller graph for this function:

#### 12.24.3.2 messageAvailable

```cpp
void SingleApplication::messageAvailable(
    QString message) [signal]
```

messageAvailable notification from commandline
Parameters

- **message**: args sent to qtpass executable

Here is the caller graph for this function:

![Caller Graph for receiveMessage](image)

### 12.24.3.3 receiveMessage

```cpp
def SingleApplication::receiveMessage ()
```

**SingleApplication::receiveMessage** we have received (a command line) message.

Definition at line 44 of file singleapplication.cpp.

Here is the caller graph for this function:

![Caller Graph for sendMessage](image)

### 12.24.3.4 sendMessage()

```cpp
bool SingleApplication::sendMessage (const QString & message)
```

**SingleApplication::sendMessage** send a message (from commandline) to an already running **QtPass** instance.
Parameters

\textit{message}

Returns

Definition at line 72 of file singleapplication.cpp.

Here is the caller graph for this function:

![Caller Graph]

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

- src/singleapplication.h
- src/singleapplication.cpp

## 12.25 StoreModel Class Reference

The QSortFilterProxyModel for handling filesystem searches.

\texttt{#include <storemodel.h>}

Inheritance diagram for StoreModel:

![Inheritance Diagram]

Generated by Doxygen
Public Member Functions

- StoreModel ()
  
  StoreModel::StoreModel SubClass of QSortFilterProxyModel via http://www.qtcentre.org/threads/46471↩QTreeView-Filter.

- bool filterAcceptsRow (int, const QModelIndex &) const override
  
  StoreModel::filterAcceptsRow should row be shown, wrapper for StoreModel::ShowThis method.

- bool ShowThis (const QModelIndex) const
  
  StoreModel::ShowThis should a row be shown, based on our search criteria.

- void setModelAndStore (QFileSystemModel ∗sourceModel, QString passStore)
  
  StoreModel::setModelAndStore update the source model and store.

- QVariant data (const QModelIndex &index, int role) const override
  
  StoreModel::data don't show the .gpg at the end of a file.

- bool lessThan (const QModelIndex &source_left, const QModelIndex &source_right) const override
  
  StoreModel::lessThan.

- Qt::DropActions supportedDropActions () const override
  
  StoreModel::supportedDropActions enable drop.

- Qt::DropActions supportedDragActions () const override
  
  StoreModel::supportedDragActions enable drag.

- Qt::ItemFlags flags (const QModelIndex &index) const override
  
  StoreModel::flags.

- QStringList mimeTypes () const override
  
  StoreModel::mimeTypes.

- QMimeData ∗mimeData (const QModelIndexList &indexes) const override
  
  StoreModel::mimeData.

- bool canDropMimeData (const QMimeData ∗data, Qt::DropAction action, int row, int column, const QModelIndex &parent) const override
  
  StoreModel::canDropMimeData.

- bool dropMimeData (const QMimeData ∗data, Qt::DropAction action, int row, int column, const QModelIndex &parent) override
  
  StoreModel::dropMimeData.

12.25.1 Detailed Description

The QSortFilterProxyModel for handling filesystem searches.

Definition at line 12 of file storemodel.h.
12.25 StoreModel Class Reference

12.25.2 Constructor & Destructor Documentation

12.25.2.1 StoreModel()

StoreModel::StoreModel()

SubClass of QSortFilterProxyModel via http://www.qtcentre.org/threads/46471-
QTreeView-Filter.

Definition at line 31 of file storemodel.cpp.

12.25.3 Member Function Documentation

12.25.3.1 canDropMimeData()

bool StoreModel::canDropMimeData ( const QMimeData * data, Qt::DropAction action, int row, int column, const QModelIndex & parent ) const [override]

StoreModel::canDropMimeData.

Parameters

<table>
<thead>
<tr>
<th>data</th>
<th>action</th>
<th>row</th>
<th>column</th>
<th>parent</th>
</tr>
</thead>
</table>

Generated by Doxygen
Stores the MIME type of a file.

Definition at line 186 of file storemodel.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

12.25.3.2 data()

QVariant StoreModel::data (  
    const QModelIndex & index,  
    int role ) const [override]

StoreModel::data don't show the .gpg at the end of a file.

Parameters

<table>
<thead>
<tr>
<th>index</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
</tr>
</tbody>
</table>

Returns

Definition at line 94 of file storemodel.cpp.
Here is the caller graph for this function:

![Caller Graph](image)

### 12.25.3.3 dropMimeData()

```cpp
bool StoreModel::dropMimeData (  
    const QMimeData * data,  
    Qt::DropAction action,  
    int row,  
    int column,  
    const QModelIndex & parent ) [override]
```

**StoreModel::dropMimeData.**

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>QMimeData *</td>
</tr>
<tr>
<td>action</td>
<td>Qt::DropAction</td>
</tr>
<tr>
<td>row</td>
<td>int</td>
</tr>
<tr>
<td>column</td>
<td>int</td>
</tr>
<tr>
<td>parent</td>
<td>const QModelIndex &amp;</td>
</tr>
</tbody>
</table>

**Returns**

Definition at line 235 of file storemodel.cpp.

Here is the call graph for this function:

![Call Graph](image)
12.25.3.4 filterAcceptsRow()

```cpp
bool StoreModel::filterAcceptsRow ( 
    int sourceRow, 
    const QModelIndex & sourceParent ) const [override]
```

*StoreModel::filterAcceptsRow* should row be shown, wrapper for *StoreModel::ShowThis* method.

**Parameters**

<table>
<thead>
<tr>
<th>sourceRow</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceParent</td>
</tr>
</tbody>
</table>

**Returns**

Definition at line 40 of file storemodel.cpp.

Here is the call graph for this function:

![Call graph](image)

12.25.3.5 flags()

```cpp
Qt::ItemFlags StoreModel::flags ( 
    const QModelIndex & index ) const [override]
```

*StoreModel::flags*.

**Parameters**

| index |

**Returns**

Definition at line 131 of file storemodel.cpp.
12.25.3.6 \textbf{lessThan()}

\begin{verbatim}
bool StoreModel::lessThan (
    const QModelIndex & source_left,
    const QModelIndex & source_right ) const [override]
\end{verbatim}

\texttt{StoreModel::lessThan}.

\begin{description}
\item[Parameters]
\begin{itemize}
\item \texttt{source\_left}
\item \texttt{source\_right}
\end{itemize}
\end{description}

\textbf{Returns}

Definition at line 299 of file storemodel.cpp.

12.25.3.7 \textbf{mimeData()}

\begin{verbatim}
QMimeData * StoreModel::mimeData ( 
    const QModelIndexList & indexes ) const [override]
\end{verbatim}

\texttt{StoreModel::mimeData}.

\begin{description}
\item[Parameters]
\begin{itemize}
\item \texttt{indexes}
\end{itemize}
\end{description}

\textbf{Returns}

Definition at line 155 of file storemodel.cpp.

12.25.3.8 \textbf{mimeTypes()}

\begin{verbatim}
QStringList StoreModel::mimeTypes ( ) const [override]
\end{verbatim}

\texttt{StoreModel::mimeTypes}.

\textbf{Returns}

Definition at line 144 of file storemodel.cpp.
12.25.3.9  setModelAndStore()

```cpp
void StoreModel::setModelAndStore (  
    QFileSystemModel ∗ sourceModel,  
    QString passStore )
```

`StoreModel::setModelAndStore` update the source model and store.

**Parameters**

- `sourceModel`
- `passStore`

Definition at line 81 of file storemodel.cpp.

Here is the caller graph for this function:

```
MainWindow::MainWindow      StoreModel::setModelAndStore
```

12.25.3.10  ShowThis()

```cpp
bool StoreModel::ShowThis (  
    const QModelIndex index ) const
```

`StoreModel::ShowThis` should a row be shown, based on our search criteria.

**Parameters**

- `index`

Returns

Definition at line 52 of file storemodel.cpp.
12.25.3.11 supportedDragActions()

Qt::DropActions StoreModel::supportedDragActions ( ) const [override]

StoreModel::supportedDragActions enable drag.

Returns

Definition at line 122 of file storemodel.cpp.

12.25.3.12 supportedDropActions()

Qt::DropActions StoreModel::supportedDropActions ( ) const [override]

StoreModel::supportedDropActions enable drop.

Returns

Definition at line 114 of file storemodel.cpp.

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

- src/storemodel.h
- src/storemodel.cpp

Generated by Doxygen
12.26 TrayIcon Class Reference

Handles the system tray icon and menu.

`#include <trayicon.h>`

Inheritance diagram for TrayIcon:

```
  QWidget
     ^     
     |  
TrayIcon
```

Collaboration diagram for TrayIcon:

```
  QWidget
     ^     
     |  
TrayIcon
```

**Public Slots**

- void `showHideParent` ()
  
  `TrayIcon::showHideParent` toggle app visibility.

- void `iconActivated` (QSystemTrayIcon::ActivationReason reason)
  
  `TrayIcon::iconActivated` you clicked on the trayicon.

**Public Member Functions**

- `TrayIcon` (QMainWindow *parent)
  
  `TrayIcon::TrayIcon` use a (system) tray icon with a nice QtPass logo on it (currently) only Quits.

- void `showMessage` (const QString &title, const QString &msg, int time)
  
  `TrayIcon::showMessage` show a system tray message for notification.

- void `setVisible` (bool visible)
  
  `TrayIcon::setVisible` show or hide the icon.

- bool `getIsAllocated` ()
  
  `TrayIcon::getIsAllocated` return if TrayIcon is allocated.
12.26 TrayIcon Class Reference

12.26.1 Detailed Description

Handles the systemtray icon and menu.

Definition at line 14 of file trayicon.h.

12.26.2 Constructor & Destructor Documentation

12.26.2.1 TrayIcon()

TrayIcon::TrayIcon ( QMainWindow * parent ) [explicit]

TrayIcon use a (system) tray icon with a nice QtPass logo on it (currently) only Quits.

Parameters

- parent

Definition at line 16 of file trayicon.cpp.

Here is the call graph for this function:

12.26.3 Member Function Documentation

12.26.3.1 getIsAllocated()

bool TrayIcon::getIsAllocated ( )

TrayIcon::getIsAllocated return if TrayIcon is allocated.

Definition at line 57 of file trayicon.cpp.
12.26.3.2 iconActivated

```cpp
void TrayIcon::iconActivated ( 
    QSystemTrayIcon::ActivationReason reason ) [slot]
```

**TrayIcon::iconActivated** you clicked on the trayicon.

**Parameters**

- **reason**

Definition at line 112 of file trayicon.cpp.

Here is the call graph for this function:

![Call Graph for iconActivated]

Here is the caller graph for this function:

![Caller Graph for iconActivated]

12.26.3.3 setVisible()

```cpp
void TrayIcon::setVisible ( 
    bool visible )
```

**TrayIcon::setVisible** show or hide the icon.

**Parameters**

- **visible**
12.26.3.4  showHideParent

void TrayIcon::showHideParent ( )  [slot]

TrayIcon::showHideParent toggle app visibility.

Definition at line 101 of file trayicon.cpp.

Here is the caller graph for this function:

```
  TrayIcon::TrayIcon  
    |                 
    v                 
TrayIcon::iconActivated  
    |                 
    v                 
TrayIcon::showHideParent
```

12.26.3.5  showMessage()

void TrayIcon::showMessage (  
        const QString & title,  
        const QString & msg,  
        int time )

TrayIcon::showMessage show a systray message for notification.

Parameters

- **title**
- **msg**
- **time**

Definition at line 132 of file trayicon.cpp.

Here is the caller graph for this function:

```
  TrayIcon::TrayIcon  
    |                 
    v                 
TrayIcon::iconActivated  
    |                 
    v                 
TrayIcon::showMessage
```

The documentation for this class was generated from the following files:
The \texttt{tst\_ui} class is our first unit test.

Inheritance diagram for \texttt{tst\_ui}:

\begin{center}
\begin{tikzpicture}
  \node[draw,fill=white] {QObject} ;
  \node[draw,fill=white,below=of current bounding box] {tst\_ui} ;
  \draw[->] (QObject) -- (tst\_ui) ;
\end{tikzpicture}
\end{center}

Collaboration diagram for \texttt{tst\_ui}:

\begin{center}
\begin{tikzpicture}
  \node[draw,fill=white] {QObject} ;
  \node[draw,fill=white,below=of current bounding box] {tst\_ui} ;
  \draw[->] (QObject) -- (tst\_ui) ;
\end{tikzpicture}
\end{center}

\subsection{Detailed Description}

The \texttt{tst\_ui} class is our first unit test.

Definition at line 9 of file \texttt{tst\_ui.cpp}.

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

- \texttt{tests/auto/ui/tst\_ui.cpp}
The `tst_util` class is our first unit test.

Inheritance diagram for `tst_util`:

Collaboration diagram for `tst_util`:

**Public Slots**

- void `init()`  
  `tst_util::init` unit test init method
- void `cleanup()`  
  `tst_util::cleanup` unit test cleanup method

**Public Member Functions**

- `tst_util()`  
  `tst_util::tst_util` basic constructor
- `~tst_util()` override  
  `tst_util::~tst_util` basic destructor
12.28.1 Detailed Description

The tst_util class is our first unit test.
Definition at line 10 of file tst_util.cpp.

12.28.2 Constructor & Destructor Documentation

12.28.2.1 tst_util()

tst_util::tst_util ( ) [default]
tst_util::tst_util basic constructor

12.28.2.2 ~tst_util()

tst_util::~tst_util ( ) [override], [default]
tst_util::~tst_util basic destructor

12.28.3 Member Function Documentation

12.28.3.1 cleanup

void tst_util::cleanup ( ) [slot]
tst_util::cleanup unit test cleanup method
Definition at line 50 of file tst_util.cpp.

12.28.3.2 init

void tst_util::init ( ) [slot]
tst_util::init unit test init method
Definition at line 45 of file tst_util.cpp.

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

- tests/auto/util/tst_util.cpp
12.29 UserInfo Struct Reference

Stores key info lines including validity, creation date and more.

```cpp
#include <userinfo.h>
```

### Public Member Functions

- **UserInfo ()**
- **bool fullyValid ()**
  ```cpp
  UserInfo::fullyValid when validity is f or u. http://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git;a= blob_plain;f=doc/DETAILS.
  ```
- **bool marginallyValid ()**
  ```cpp
  UserInfo::marginallyValid when validity is m. http://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git; a=blob_ plain;f=doc/DETAILS.
  ```
- **bool isValid ()**
  ```cpp
  UserInfo::isValid when fullyValid or marginallyValid.
  ```

### Public Attributes

- **QString name**
  ```cpp
  UserInfo::name full name.
  ```
- **QString key_id**
  ```cpp
  UserInfo::key_id hexadecimal representation.
  ```
- **char validity**
  ```cpp
  UserInfo::validity GnuPG representation of validity http://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git; a=blob_ plain;f=doc/DETAILS.
  ```
- **bool have_secret**
  ```cpp
  UserInfo::have_secret secret key is available (can decrypt with this key)
  ```
- **bool enabled**
  ```cpp
  UserInfo::enabled.
  ```
- **QDateTime expiry**
  ```cpp
  UserInfo::expiry date/time key expires.
  ```
- **QDateTime created**
  ```cpp
  UserInfo::created date/time key was created.
  ```

12.29.1 Detailed Description

Stores key info lines including validity, creation date and more.

Definition at line 11 of file userinfo.h.

12.29.2 Constructor & Destructor Documentation
12.29.2.1 UserInfo()

UserInfo::UserInfo ( ) [inline]

Definition at line 12 of file userinfo.h.

12.29.3 Member Function Documentation

12.29.3.1 fullyValid()

bool UserInfo::fullyValid ( ) [inline]

UserInfo::fullyValid when validity is f or u.  http://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git;a=blob_plain;f=doc/DETAILS.

Definition at line 18 of file userinfo.h.

Here is the caller graph for this function:

12.29.3.2 isValid()

bool UserInfo::isValid ( ) [inline]

UserInfo::isValid when fullyValid or marginallyValid.

Definition at line 27 of file userinfo.h.

Here is the call graph for this function:
12.29.3.3 marginallyValid()

```cpp
bool UserInfo::marginallyValid() [inline]
```

`UserInfo::marginallyValid` when validity is m. 

Reference:
http://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git;a=blob_plain;f=doc/DETAILS.

Definition at line 23 of file userinfo.h.

Here is the caller graph for this function:

```
UserInfo::isValid → UserInfo::marginallyValid
```

12.29.4 Member Data Documentation

12.29.4.1 created

```cpp
QDateTime UserInfo::created
```

`UserInfo::created` date/time key was created.

Definition at line 58 of file userinfo.h.

12.29.4.2 enabled

```cpp
bool UserInfo::enabled
```

`UserInfo::enabled`.

Definition at line 50 of file userinfo.h.

12.29.4.3 expiry

```cpp
QDateTime UserInfo::expiry
```

`UserInfo::expiry` date/time key expires.

Definition at line 54 of file userinfo.h.
12.29.4.4  have_secret

bool UserInfo::have_secret

UserInfo::have_secret secret key is available (can decrypt with this key)

Definition at line 46 of file userinfo.h.

12.29.4.5  key_id

QString UserInfo::key_id

UserInfo::key_id hexadecimal representation.

Definition at line 36 of file userinfo.h.

12.29.4.6  name

QString UserInfo::name

UserInfo::name full name.

Definition at line 32 of file userinfo.h.

12.29.4.7  validity

char UserInfo::validity

UserInfo::validity GnuPG representation of validity

Definition at line 41 of file userinfo.h.

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

- src/userinfo.h
12.30 UsersDialog Class Reference

Handles listing and editing of GPG users.

#include <usersdialog.h>

Inheritance diagram for UsersDialog:

Collaboration diagram for UsersDialog:

Public Slots

- void accept ()
  
  UsersDialog::accept.

Public Member Functions

- UsersDialog (QString dir, QWidget *parent=nullptr)
  
  UsersDialog::UsersDialog basic constructor.
- ~UsersDialog ()
  
  UsersDialog::~UsersDialog basic destructor.
Protected Member Functions

- void `closeEvent` (QCloseEvent *event)
  
  `UsersDialog::closeEvent` might have to store size and location if that is wanted.

- void `keyPressEvent` (QKeyEvent *event)

  `UsersDialog::keyPressEvent` clear the lineEdit when escape is pressed. No action for Enter currently.

12.30.1 Detailed Description

Handles listing and editing of GPG users.

Selection of whom to encrypt to.

Definition at line 23 of file usersdialog.h.

12.30.2 Constructor & Destructor Documentation

12.30.2.1 `UsersDialog()`

`UsersDialog::UsersDialog ( 
  QString dir,
  QWidget *parent = nullptr ) [explicit]

`UsersDialog::UsersDialog` basic constructor.

Parameters

- `parent`

Definition at line 18 of file usersdialog.cpp.

Here is the call graph for this function:
12.30.2.2  ~UsersDialog()

UsersDialog:~UsersDialog ( )

UsersDialog:~UsersDialog basic destructor.
Definition at line 82 of file usersdialog.cpp.

12.30.3  Member Function Documentation

12.30.3.1  accept

void UsersDialog::accept ( )  [slot]

UsersDialog::accept.
Definition at line 89 of file usersdialog.cpp.
Here is the call graph for this function:

Here is the caller graph for this function:

12.30.3.2  closeEvent()

void UsersDialog::closeEvent ( )  [protected]

UsersDialog::closeEvent might have to store size and location if that is wanted.
12.30.3.3 keyPressEvent()

```cpp
void UserDialog::keyPressEvent ( 
    QKeyEvent * event ) [protected]
```

UserDialog::keyPressEvent clear the lineEdit when escape is pressed. No action for Enter currently.

Parameters

- event

Definition at line 110 of file usersdialog.cpp.

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

- src/usersdialog.h
- src/usersdialog.cpp

12.31 Util Class Reference

Some static utilities to be used elsewhere.

```cpp
#include <util.h>
```

Static Public Member Functions

- static QString findBinaryInPath (QString binary)  
  ```cpp
  Util::findBinaryInPath search for executables.
  ```
- static QString findPasswordStore ()  
  ```cpp
  Util::findPasswordStore look for common .password-store folder location.
  ```
- static QString normalizeFolderPath (QString path)  
  ```cpp
  Util::normalizeFolderPath let's always end folders with a QDir::separator()
  ```
- static bool checkConfig ()  
  ```cpp
  Util::checkConfig do we have prerequisite settings?
  ```
- static QString getDir (const QModelIndex &index, bool forPass, const QFileSystemModel &model, const StoreModel &storeModel)  
  ```cpp
  Util::getDir get selectd folder path.
  ```
- static void copyDir (const QString &src, const QString &dest)  
  ```cpp
  Util::copyDir.
  ```
12.31 Util Class Reference

12.31.1 Detailed Description

Some static utilities to be used elsewhere.

Definition at line 15 of file util.h.

12.31.2 Member Function Documentation

12.31.2.1 checkConfig()

bool Util::checkConfig () [static]

Util::checkConfig do we have prerequisite settings?

Returns

Definition at line 142 of file util.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:
12.31.2.2 copyDir()

void Util::copyDir (  
    const QString & src,  
    const QString & dest  ) [static]

Util::copyDir.
12.31 Util Class Reference

Parameters

| src | dest |

Definition at line 182 of file util.cpp.

12.31.2.3 findBinaryInPath()

Qstring Util::findBinaryInPath (  
  Qstring binary ) [static]

Util::findBinaryInPath search for executables.

Parameters

| binary |

Returns

Definition at line 90 of file util.cpp.

Here is the call graph for this function:

```
Util::findBinaryInPath  ->  Executor::executeBlocking
```

Here is the caller graph for this function:

```
MainWindow::MainWindow  ->  QtPass::init  ->  QtPassSettings::initExecutables  ->  Util::findBinaryInPath
```
12.31.2.4 findPasswordStore()

QString Util::findPasswordStore ( ) [static]

Util::findPasswordStore look for common .password-store folder location.

Returns

Definition at line 56 of file util.cpp.

Here is the call graph for this function:

```
Util::findPasswordStore  
```

Here is the caller graph for this function:

```
MainWindow::MainWindow  
```

12.31.2.5 getDir()

QString Util::getDir ( const QModelIndex & index, bool forPass, const QFileSystemModel & model, const StoreModel & storeModel ) [static]

Util::getDir get selectd folder path.

Parameters

<table>
<thead>
<tr>
<th>index</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>forPass</td>
<td>short or full path</td>
</tr>
<tr>
<td>model</td>
<td>the filesystem model to operate on</td>
</tr>
<tr>
<td>storeModel</td>
<td>our storemodel to operate on</td>
</tr>
</tbody>
</table>
Returns

path

Definition at line 160 of file util.cpp.

Here is the call graph for this function:

Util::getDir
QtPassSettings::getPassStore
QtPassSettings::getInstance

Here is the caller graph for this function:

MainWindow::keyPressEvent
MainWindow::on_treeView_clicked
Util::getDir

12.31.2.6 normalizeFolderPath()

QString Util::normalizeFolderPath ( QString path ) [static]

Util::normalizeFolderPath let's always end folders with a QDir::separator()

Parameters

| path |

Returns

Definition at line 79 of file util.cpp.

Here is the caller graph for this function:

MainWindow::MainWindow
Util::findPasswordStore
QtPass::init
The documentation for this class was generated from the following files:

- `src/util.h`
- `src/util.cpp`
Chapter 13

File Documentation

13.1 CHANGELOG.md File Reference

13.2 CONTRIBUTING.md File Reference

13.3 FAQ.md File Reference

13.4 main/main.cpp File Reference

#include "mainwindow.h"
#include <QApplication>
#include <QDir>
#include <QTranslator>

Include dependency graph for main.cpp:

Functions

- int main (int argc, char *argv[])
13.4.1 Function Documentation

13.4.1.1 main()

int main (  
    int argc,  
    char * argv[ ] )

Parameters

- argc
- argv

Returns

Definition at line 42 of file main.cpp.

Here is the call graph for this function:

- SingleApplication::isRunning
- MainWindow::messageAvailable
- SingleApplication::sendMessage

13.5 README.md File Reference

13.6 src/configdialog.cpp File Reference

#include "configdialog.h"
#include "keygendialog.h"
### 13.7 src/configdialog.h File Reference

```cpp
#include "enums.h"
#include "passwordconfiguration.h"
#include <QString>
```

Include dependency graph for configdialog.h:
This graph shows which files directly or indirectly include this file:

![Dependency Graph]

### Classes

- class ConfigDialog
  
  The ConfigDialog handles the configuration interface.

### Namespaces

- Ui

### 13.8 src/debughelper.h File Reference

```cpp
#include <QDebug>
```

Include dependency graph for debughelper.h:

![Dependency Graph]

### Macros

- #define dbg() qDebug() << __FILE__ ":" << __LINE__

### 13.8.1 Macro Definition Documentation
### 13.8.1.1 dbg

```c
#define dbg( ) qDebug() << __FILE__ "::" << __LINE__
```

Definition at line 8 of file debughelper.h.

### 13.9 src/deselectabletreeview.h File Reference

```c
#include <QCoreApplication>
#include <QMouseEvent>
#include <QTime>
#include <QTreeView>
```

Include dependency graph for deselectabletreeview.h:

```
src/deselectabletreeview.h
QCoreApplication QMouseEvent QTime QTreeView
```

### Classes

- **class DeselectableTreeView**
  
  The **DeselectableTreeView** class loosely based on [http://stackoverflow.com/questions/2761284/](http://stackoverflow.com/questions/2761284/) thanks to Yassir Ennazk.

### 13.10 src/enums.h File Reference

This graph shows which files directly or indirectly include this file:

```
src/enums.h
src/configdialog.h src/qtpasssettings.h
src/pass.h src/simpletransaction.h
src/configdialog.cpp src/qtpasssettings.cpp
src/imitatepass.h src/realpass.h src/simpletransaction.cpp
```

### Namespaces

- ** Enums **

  Enumerators for configuration and runtime items.
Enumerations

- **enum Enums::clipBoardType {**
  Enums::CLIPBOARD_NEVER = 0,
  Enums::CLIPBOARD_ALWAYS = 1,
  Enums::CLIPBOARD_ON_DEMAND = 2
  **}

- **enum Enums::PROCESS {**
  Enums::GIT_INIT = 0,
  Enums::GIT_ADD, Enums::GIT_COMMIT, Enums::GIT_RM,
  Enums::GIT_PULL, Enums::GIT_PUSH, Enums::PASS_SHOW, Enums::PASS_INSERT,
  Enums::PASS_REMOVE, Enums::PASS_INIT, Enums::GPG_GENKEYS, Enums::PASS_MOVE,
  Enums::PASS_COPY, Enums::GIT_MOVE, Enums::GIT_COPY, Enums::PROCESS_COUNT,
  Enums::INVALID, Enums::PASS_OTP_GENERATE
  **}

13.11 src/executor.cpp File Reference

```cpp
#include "executor.h"
#include <QCoreApplication>
#include <QDir>
#include <QTextCodec>
```

Include dependency graph for executor.cpp:

13.12 src/executor.h File Reference

```cpp
#include <QObject>
#include <QProcess>
#include <QQueue>
```

Include dependency graph for executor.h:
This graph shows which files directly or indirectly include this file:

```
#include "filecontent.h"
```

Include dependency graph for filecontent.cpp:

```
src/filecontent.cpp
filecontent.h
QList QString QStringList
```

13.14 src/filecontent.h File Reference

```
#include <QList>
#include <QString>
```

Generated by Doxygen
#include <QStringList>

Include dependency graph for filecontent.h:

This graph shows which files directly or indirectly include this file:

**Classes**

- struct NamedValue
- class NamedValues

The NamedValues class is mostly a list of NamedValue but also has a method to take a specific NamedValue pair out of the list.

- class FileContent

### 13.15 src/imitatepass.cpp File Reference

#include "imitatepass.h"
#include "qtpasssettings.h"
#include <QDirIterator>
#include <utility>

Include dependency graph for imitatepass.cpp:
13.16 src/imitatepass.h File Reference

#include "pass.h"
#include "simpletransaction.h"

Include dependency graph for imitatepass.h:

This graph shows which files directly or indirectly include this file:

Classes

• class ImitatePass

Imitates pass features when pass is not enabled or available.

13.17 src/keygendialog.cpp File Reference

#include "keygendialog.h"
#include "configdialog.h"
#include "qprogressindicator.h"
#include "ui_keygendialog.h"
#include <QMessageBox>

Include dependency graph for keygendialog.cpp:
13.18  src/keygendialog.h File Reference

#include <QDialog>
Include dependency graph for keygendialog.h:

This graph shows which files directly or indirectly include this file:

Classes

- class KeygenDialog
  Handles GPG keypair generation.

Namespaces

- Ui

13.19  src/mainwindow.cpp File Reference

#include "mainwindow.h"
#include "configdialog.h"
#include "filecontent.h"
#include "keygendialog.h"
#include "passworddialog.h"
# include "qpushbuttonasqrcode.h"
# include "qpushbuttonshowpassword.h"
# include "qpushbuttonwithclipboard.h"
# include "qtpass.h"
# include "qtpasssettings.h"
# include "settingsconstants.h"
# include "trayicon.h"
# include "ui_mainwindow.h"
# include "usersdialog.h"
# include "util.h"
# include "QCloseEvent"
# include "QDesktopServices"
# include "QDialog"
# include "QFileInfo"
# include "QInputDialog"
# include "QLabel"
# include "QMenu"
# include "QMessageBox"
# include "QShortcut"
# include "QTimer"

Include dependency graph for mainwindow.cpp:

13.20 src/mainwindow.h File Reference

#include "storemodel.h"
#include "QFileSystemModel"
#include "QItemSelectionModel"
#include "QMainWindow"
#include "QProcess"
#include "QTimer"

Include dependency graph for mainwindow.h:
This graph shows which files directly or indirectly include this file:

![Dependency Graph](image)

**Classes**
- class **MainWindow**

  The **MainWindow** class does way too much, not only is it a switchboard, configuration handler and more, it's also the process-manager.

**Namespaces**
- **Ui**

**Macros**
- `#define SingleApplication QApplication`

### 13.20.1 Macro Definition Documentation

#### 13.20.1.1 SingleApplication

```cpp
#define SingleApplication QApplication
```

Definition at line 16 of file `mainwindow.h`.

### 13.21 src/pass.cpp File Reference

```cpp
#include "pass.h"
#include "qtpasssettings.h"
#include "util.h"
#include "enums.h"
#include "executor.h"
#include "QProcess"
#include "QQueue"
#include "userinfo.h"
#include "QString"
#include "cassert"
#include "map"
#include "QRandomGenerator"
#include "QObject"
#include "QDateTime"
#include "QProcessEnvironment"
#include "QByteArray"
#include "QHash"
#include "QPoint"
#include "QSettings"
#include "QSize"
#include "QVariant"
#include "simpletransaction.h"
#include "queue"
#include "storemodel.h"
#include "QFileSystemModel"
#include "QProcessEnvironment"
#include "QSortFilterProxyModel"
```

Include dependency graph for `pass.cpp`:
13.22  src/pass.h File Reference

```c
#include "enums.h"
#include "executor.h"
#include "userinfo.h"
#include <QProcess>
#include <QQueue>
#include <QString>
#include <cassert>
#include <map>
#include <QRandomGenerator>
```

Include dependency graph for pass.h:

This graph shows which files directly or indirectly include this file:

Classes

- class `Pass`

  Acts as an abstraction for pass or pass imitation.

13.23  src/passwordconfiguration.h File Reference

```c
#include <QString>
```
Include dependency graph for passwordconfiguration.h:

This graph shows which files directly or indirectly include this file:

Classes

- struct PasswordConfiguration
  
  Holds the Password configuration settings.

13.24 src/passworddialog.cpp File Reference

```cpp
#include "passworddialog.h"
#include "filecontent.h"
#include "pass.h"
#include "passwordconfiguration.h"
#include "qtpasssettings.h"
#include "ui_passworddialog.h"
#include <QLabel>
#include <QLineEdit>
```

Include dependency graph for passworddialog.cpp:
#include "passwordconfiguration.h"
#include <QDialog>

Include dependency graph for passworddialog.h:

This graph shows which files directly or indirectly include this file:

Classes

- class PasswordDialog
  
  PasswordDialog Handles the inserting and editing of passwords.

Namespaces

- Ui
13.26 src/qprogressindicator.cpp File Reference

#include "qprogressindicator.h"
#include <QPainter>
Include dependency graph for qprogressindicator.cpp:

13.27 src/qprogressindicator.h File Reference

#include <QColor>
#include <QWidget>
Include dependency graph for qprogressindicator.h:
This graph shows which files directly or indirectly include this file:

```
src/qprogressindicator.h
src/keygendiaglog.cpp src/qprogressindicator.cpp
```

### Classes

- **class QProgressIndicator**

  *The QProgressIndicator class lets an application display a progress indicator to show that a lengthy task is under way.*

---

13.28  src/qpushbuttonasqrcode.cpp File Reference

```
#include "qpushbuttonasqrcode.h"
#include <QTimer>
```

Include dependency graph for qpushbuttonasqrcode.cpp:
13.29  src/qpushbuttonasqrcode.h File Reference

#include <QPushButton>

Include dependency graph for qpushbuttonasqrcode.h:

This graph shows which files directly or indirectly include this file:

Classes

- class QPushButtonAsQRCode
  
  Stylish widget to display the field as QR Code.

13.30  src/qpushbuttonshowpassword.cpp File Reference

#include "qpushbuttonshowpassword.h"
#include <QTimer>
Include dependency graph for qpushbuttonshowpassword.cpp:

```
#include <QLineEdit>
#include <QPushButton>
#include <QTimer>
```

Include dependency graph for qpushbuttonshowpassword.h:

```
#include <QLineEdit>
#include <QPushButton>
```
This graph shows which files directly or indirectly include this file:

![Dependency Graph](image)

**Classes**

- class `QPushButtonShowPassword`

### 13.32 src/qpushbuttonwithclipboard.cpp File Reference

```cpp
#include "qpushbuttonwithclipboard.h"
#include <QTimer>
```

Include dependency graph for `qpushbuttonwithclipboard.cpp`:

![Dependency Graph](image)

### 13.33 src/qpushbuttonwithclipboard.h File Reference

```cpp
#include <QPushButton>
```
Include dependency graph for qpushbuttonwithclipboarde.h:

This graph shows which files directly or indirectly include this file:

Classes

- class QPushButtonWithClipboard
  
  Stylish widget to allow copying of password and account details.

13.34 src/qtpass.cpp File Reference

```
#include "qtpass.h"
#include "mainwindow.h"
#include "qtpasssettings.h"
#include <QApplication>
#include <QClipboard>
#include <QDialog>
#include <QLabel>
#include <QPixmap>
#include <QInputDialog>
#include <QLineEdit>
```

Generated by Doxygen
#include <utility>
#include <QObject>
#include <QProcess>
#include <QTimer>

#include dependency graph for qtpass.h:

This graph shows which files directly or indirectly include this file:

Classes

- class QtPass
#include "qtpasssettings.h"
#include "pass.h"
#include "util.h"
#include <QCoreApplication>

Include dependency graph for qtpasssettings.cpp:

---

#include "enums.h"
#include "imitatepass.h"
#include "passwordconfiguration.h"
#include "realpass.h"
#include "settingsconstants.h"
#include <QByteArray>
#include <QHash>
#include <QPoint>
#include <QSettings>
#include <QSize>
#include <QVariant>

Include dependency graph for qtpasssettings.h:

---

This graph shows which files directly or indirectly include this file:
Classes

- class QtPassSettings
  
  Singleton that stores qtpass' settings, saves and loads config.

13.38  src/realpass.cpp File Reference

```cpp
#include "realpass.h"
#include "qtpasssettings.h"
#include <QDir>
#include <QFileInfo>
#include <utility>
```

Include dependency graph for realpass.cpp:

13.39  src/realpass.h File Reference

```cpp
#include "pass.h"
```

Include dependency graph for realpass.h:

This graph shows which files directly or indirectly include this file:
Classes

- class RealPass
  
  *Wrapper for executing pass to handle the password-store.*

13.40  src/settingsconstants.cpp File Reference

```cpp
#include "settingsconstants.h"

Include dependency graph for settingsconstants.cpp:
```

13.41  src/settingsconstants.h File Reference

```cpp
#include <QString>

Include dependency graph for settingsconstants.h:
```
This graph shows which files directly or indirectly include this file:

Classes

- class SettingConstants

  *Table for the naming of configuration items.*

13.42 src/simpletransaction.cpp File Reference

```cpp
#include "simpletransaction.h"
#include <utility>
```

Include dependency graph for simpletransaction.cpp:

```
src/simpletransaction.cpp
  \|-- simpletransaction.h
depends on:
  \|-- utility
  \|-- enums.h
  \|-- queue
```

13.43 src/simpletransaction.h File Reference

```cpp
#include "enums.h"
#include <queue>
```

Generated by Doxygen
Include dependency graph for simpletransaction.h:

```
src/simpletransaction.h
  ^
  v
enums.h
queue
```

This graph shows which files directly or indirectly include this file:

```
singleapplication.h
  ^
  v
src/singleapplication.cpp
```

**Classes**

- `class simpleTransaction`

### 13.44 src/singleapplication.cpp File Reference

```cpp
#include "singleapplication.h"
#include <QLocalSocket>
#include <utility>
```

Include dependency graph for singleapplication.cpp:

```
src/singleapplication.cpp
  ^
  v
singleapplication.h
  ^
  v
src/singleapplication.cpp
```

```
QApplication QLocalServer QSharedMemory
  ^
  v
src/singleapplication.cpp
```

```
QLocalSocket utility
  ^
  v
src/singleapplication.cpp
```

Generated by Doxygen
13.45  src/singleapplication.h File Reference

```cpp
#include <QApplication>
#include <QLocalServer>
#include <QSharedMemory>
```

Include dependency graph for singleapplication.h:

```
src/singleapplication.h
```

This graph shows which files directly or indirectly include this file:

```
QApplication  QLocalServer  QSharedMemory
```

Classes

- class SingleApplication

  The SingleApplication class is used for commandline integration.

13.46  src/storemodel.cpp File Reference

```cpp
#include "storemodel.h"
#include "qtpasssettings.h"
#include <QDebug>
#include <QMessageBox>
#include <QMimeData>
```
#include <utility>

Include dependency graph for storemodel.cpp:

```
Functions

- QDataStream & operator<< (QDataStream &out, const dragAndDropInfoPasswordStore &dragAndDropInfoPasswordStore)
- QDataStream & operator>>(QDataStream &in, dragAndDropInfoPasswordStore &dragAndDropInfoPasswordStore)

13.46.1 Function Documentation

13.46.1.1 operator<<( )

QDataStream& operator<<(QDataStream &out, const dragAndDropInfoPasswordStore &dragAndDropInfoPasswordStore )

Definition at line 9 of file storemodel.cpp.

13.46.1.2 operator>>( )

QDataStream& operator>>(QDataStream &in, dragAndDropInfoPasswordStore &dragAndDropInfoPasswordStore )

Definition at line 19 of file storemodel.cpp.

Generated by Doxygen
#include "util.h"
#include <QSortFilterProxyModel>

Include dependency graph for storemodel.h:

This graph shows which files directly or indirectly include this file:

Classes

- class StoreModel
  
  The QSortFilterProxyModel for handling filesystem searches.

- struct dragAndDropInfoPasswordStore

13.48  src/trayicon.cpp File Reference

#include "trayicon.h"
#include <QAction>
#include <QApplication>
#include <QMainWindow>
13.49 src/trayicon.h File Reference

#include <QSystemTrayIcon>
#include <QWidget>

Include dependency graph for trayicon.h:

This graph shows which files directly or indirectly include this file:
Classes

- class TrayIcon
  Handles the systemtray icon and menu.

13.50 src/userinfo.h File Reference

```
#include <QDateTime>
#include <QString>
```

Include dependency graph for userinfo.h:

```
src/userinfo.h
  QDateTime
  QString
```

This graph shows which files directly or indirectly include this file:

```
src/userinfo.h
  src/pas.h
  src/usersdialog.h
  src/imitatepass.h
  src/pass.cpp
  src/passworddialog.cpp
  src/qtpasssettings.cpp
  src/realpass.h
  src/qtpasssettings.h
  src/imitatepass.cpp
  src/configdialog.cpp
  src/mainwindow.cpp
  src/qtpass.cpp
  src/realpass.cpp
  src/storemodel.cpp
  src/usersdialog.cpp
  src/util.cpp
```

Classes

- struct UserInfo
  Stores key info lines including validity, creation date and more.

13.51 src/usersdialog.cpp File Reference

```
#include "usersdialog.h"
#include "qtpasssettings.h"
#include "ui_usersdialog.h"
#include <QCloseEvent>
#include <QKeyEvent>
#include <QMessageBox>
```

Generated by Doxygen
#include <QRegExp>
#include <QWidget>
#include <utility>

Include dependency graph for usersdialog.cpp:

### 13.52 src/usersdialog.h File Reference

```cpp
#include "userinfo.h"
#include <QDialog>
#include <QList>
#include <QList>
```

Include dependency graph for usersdialog.h:
This graph shows which files directly or indirectly include this file:

![Dependency Graph](image)

**Classes**

- class `UsersDialog`
  
  `Handles listing and editing of GPG users.`

**Namespaces**

- `Ui`

**13.53 src/util.cpp File Reference**

```cpp
#include "util.h"
#include <QDir>
#include <QFileInfo>
#include <sys/time.h>
#include "qtpasssettings.h"
#include "storemodel.h"
```

Include dependency graph for `util.cpp`:

![Dependency Graph](image)
13.54  src/util.h File Reference

#include "storemodel.h"
#include <QFileSystemModel>
#include <QProcessEnvironment>
#include <QString>

Include dependency graph for util.h:

This graph shows which files directly or indirectly include this file:

Classes

- class Util

  Some static utilities to be used elsewhere.

13.55  tests/auto/ui/tst_ui.cpp File Reference

#include "./././src/passworddialog.h"
#include "passwordconfiguration.h"
#include <QCoreApplication>
#include <QtTest>

Generated by Doxygen
#include "tst_ui.moc"

Include dependency graph for tst_ui.cpp:

Classes

- class tst_ui

  The tst_ui class is our first unit test.

13.56  tests/auto/util/tst_util.cpp File Reference

#include "../../../src/filecontent.h"
#include "../../../src/util.h"
#include <QCoreApplication>
#include <QList>
#include <QtTest>
#include "tst_util.moc"

Include dependency graph for tst_util.cpp:

Classes

- class tst_util

  The tst_util class is our first unit test.
Functions

- bool operator==(const NamedValue &a, const NamedValue &b)

13.56.1 Function Documentation

13.56.1.1 operator==()

```cpp
bool operator== (  
    const NamedValue & a,  
    const NamedValue & b  
)
```

Definition at line 28 of file tst_util.cpp.
Index

~ConfigDialog
- ConfigDialog, 50
~DeselectableTreeView
- DeselectableTreeView, 60
~ImitatePass
- ImitatePass, 75
~KeygenDialog
- KeygenDialog, 84
~MainWindow
- MainWindow, 88
~Pass
- Pass, 107
~PasswordDialog
- PasswordDialog, 130
~QtPass
- QtPass, 152
~RealPass
- RealPass, 219
~UsersDialog
- UsersDialog, 262
~tst_util
- tst_util, 256

accept
- UsersDialog, 263
addGPGId
- SettingsConstants, 225
ALLCHARS
- PasswordConfiguration, 126
ALPHABETICAL
- PasswordConfiguration, 126
ALPHANUMERIC
- PasswordConfiguration, 126
alwaysOnTop
- SettingsConstants, 225
animationDelay
- QProgressIndicator, 138
autoclearPanelSeconds
- SettingsConstants, 226
autoclearSeconds
- SettingsConstants, 226
autoPull
- SettingsConstants, 226
autoPush
- SettingsConstants, 226
avoidCapitals
- SettingsConstants, 226
avoidNumbers
- SettingsConstants, 226
boundedRandom
- Pass, 107
cancelNext
- Executor, 64
canDropMimeData
- StoreModel, 243
changeEvent
- MainWindow, 88
CHANGELOG.md, 271
Characters
- PasswordConfiguration, 126
characterSet
- PasswordConfiguration, 126
CHARSETS_COUNT
- PasswordConfiguration, 126
checkConfig
- Util, 265
cleanKeygenDialog
- MainWindow, 89
cleanup
- tst_util, 256
clearClipboard
- QtPass, 152
clearClippedText
- QtPass, 153
clicked
- QPushButtonAsQRCode, 144
- QPushButtonShowPassword, 147
- QPushButtonWithClipboard, 149
CLIPBOARD_ALWAYS
- Enums, 45
CLIPBOARD_NEVER
- Enums, 45
CLIPBOARD_ON_DEMAND
- Enums, 45
clipBoardType
- Enums, 45
- SettingsConstants, 227
closeEvent
- ConfigDialog, 51
- KeygenDialog, 84
- MainWindow, 89
- UsersDialog, 263
color
- QProgressIndicator, 138, 142
config
- MainWindow, 89
ConfigDialog
- ConfigDialog, 47
- ~ConfigDialog, 50
closeEvent, 51
ConfigDialog, 49
genKey, 51
getPasswordConfiguration, 52
getProfiles, 52
setPasswordConfiguration, 52
setPwgenPath, 52
useAutoclear, 53
useAutoclearPanel, 53
useGit, 54
useOtp, 54
usePwgen, 55
useQrencode, 56
useSelection, 56
useTemplate, 57
useTrayIcon, 57
wizard, 58
CONTRIBUTING.md, 271
Copy
ImitatePass, 75
Pass, 107
RealPass, 220
copyDir
Util, 265
copyTextToClipboard
QtPass, 153
created
UserInfo, 259
critical
MainWindow, 90
Pass, 108
CUSTOM
PasswordConfiguration, 126
data
StoreModel, 244
dbg
debughelper.h, 274
debughelper.h
dbg, 274
delay
QProgressIndicator, 142
deselect
MainWindow, 91
DeselectableTreeView, 58
¬DeselectableTreeView, 59
emptyClicked, 60
displayedWhenStopped
QProgressIndicator, 142
dragAndDropInfo, 60
dragAndDropInfoPasswordStore, 61
isDir, 61
isFile, 61
path, 61
dropMimeData
StoreModel, 245
emptyClicked
DeselectableTreeView, 60
enabled
UserInfo, 259
endReencryptPath
ImitatePass, 76
MainWindow, 91
Enums, 45
CLIPBOARD_ALWAYS, 45
CLIPBOARD_NEVER, 45
CLIPBOARD_ON_DEMAND, 45
clipBoardType, 45
GIT_ADD, 46
GIT_COMMIT, 46
GIT_COPY, 46
GIT_INIT, 46
GIT_MOVE, 46
GIT_PULL, 46
GIT_PUSH, 46
GIT_RM, 46
GPG_GENKEYS, 46
INVALID, 46
PASS_COPY, 46
PASS_INIT, 46
PASS_INSERT, 46
PASS_MOVE, 46
PASS_OTP_GENERATE, 46
PASS_REMOVE, 46
PASS_SHOW, 46
PROCESS, 46
PROCESS_COUNT, 46
error
Executor, 64
Pass, 108
eventFilter
MainWindow, 91
exec
Pass, 125
execute
Executor, 64–66
executeBlocking
Executor, 67, 68
executeWrapper
ImitatePass, 76
Pass, 108, 109
executeWrapperStarted
MainWindow, 92
Executor, 62
cancelNext, 64
error, 64
execute, 64–66
executeBlocking, 67, 68
Executor, 63
finished, 68
setEnvironment, 69
starting, 69
expiry
UserInfo, 259
FAQ.md, 271
QtPassSettings, 173
getRealPass
QtPassSettings, 174
getRecipientList
Pass, 116
getRecipientString
Pass, 117
getRemainingData
FileContent, 71
getRemainingDataForDisplay
FileContent, 72
getSavestate
QtPassSettings, 174
getSize
QtPassSettings, 175
getTextToCopy
QPushButtonAsQRCode, 145
QPushButtonWithClipboard, 149
definition
QtPassSettings, 176
getWebDavPassword
QtPassSettings, 176
getWebDavUrl
QtPassSettings, 177
getWebDavUser
QtPassSettings, 177
GIT_ADD
Enums, 46
GIT_COMMIT
Enums, 46
GIT_COPY
Enums, 46
GIT_INIT
Enums, 46
GIT_MOVE
Enums, 46
GIT_PULL
Enums, 46
GIT_PUSH
Enums, 46
GIT_RM
Enums, 46
gitExecutable
SettingsConstants, 227
GitInit
ImitatePass, 77
Pass, 118
RealPass, 220
GitPull
ImitatePass, 78
Pass, 118
RealPass, 220
GitPull_b
ImitatePass, 78
Pass, 118
RealPass, 221
GitPush
ImitatePass, 78
GPG_GENKEYS
Enums, 46
gpgExecutable
SettingsConstants, 227
gpgHome
SettingsConstants, 227
groupMainwindow
SettingsConstants, 227
groupProfiles
SettingsConstants, 228
have_secret
UserInfo, 259
heightForWidth
QProgressIndicator, 138
hideContent
SettingsConstants, 228
hideOnClose
SettingsConstants, 228
hidePassword
SettingsConstants, 228
iconActivated
TrayIcon, 251
ImitatePass, 73
~ImitatePass, 75
Copy, 75
endReencryptPath, 76
executeWrapper, 76
finished, 77
GitInit, 77
GitPull, 78
GitPull_b, 78
GitPush, 78
ImitatePass, 75
Init, 79
Insert, 79
Move, 80
OtpGenerate, 80
reencryptPath, 81
Remove, 81
Show, 82
startReencryptPath, 82
Init
ImitatePass, 79
Pass, 119
RealPass, 221
init
Pass, 119
QtPass, 154
tst_util, 256
initExecutables
QtPassSettings, 177
Insert
ImitatePass, 79
Pass, 120
RealPass, 222
setUiElementsEnabled, 98
showStatusMessage, 99
startReencryptPath, 100
userDialog, 100
mainwindow.h
   SingleApplication, 282
marginallyValid
   UserInfo, 258
maximized
   SettingsConstants, 228
messageAvailable
   MainWindow, 94
   SingleApplication, 239
mimeType
   StoreModel, 247
mimeTypes
   StoreModel, 247
Move
   ImitatePass, 80
   Pass, 122
   RealPass, 222
name
   NamedValue, 101
   UserInfo, 260
NamedValue, 101
   name, 101
   value, 101
NamedValues, 102
   NamedValues, 103
takeValue, 103
normalizeFolderPath
   Util, 269
on_treeView_clicked
   MainWindow, 95
onPush
   MainWindow, 95
operator<<
   storemodel.cpp, 299
operator>>
   storemodel.cpp, 299
operator==
   tst_util.cpp, 307
OtpGenerate
   ImitatePass, 80
   Pass, 122
   RealPass, 223
paintEvent
   QProgressIndicator, 139
parse
   FileContent, 72
Pass, 104
   ~Pass, 107
   boundedRandom, 107
   Copy, 107
critical, 108
error, 108
exec, 125
executeWrapper, 108, 109
finished, 109
finishedAny, 110
finishedCopy, 110
finishedGenerate, 111
finishedGenerateGPGKeys, 111
finishedGitInit, 111
finishedGitPull, 111
finishedGitPush, 112
finishedInit, 112
finishedInsert, 113
finishedMove, 113
finishedOtpGenerate, 113
finishedRemove, 114
finishedShow, 114
Generate_b, 115
GenerateGPGKeys, 115
generateRandomPassword, 116
getRecipientList, 116
getRecipientString, 117
GitInit, 118
GitPull, 118
GitPull_b, 118
GitPush, 118
Init, 119
init, 119
Insert, 120
listKeys, 120, 121
Move, 122
OtpGenerate, 122
Pass, 106
PROCESS, 106
processErrorExit, 122
Remove, 123
Show, 123
startingExecuteWrapper, 123
statusMsg, 124
updateEnv, 124
PASS_COPY
   Enums, 46
PASS_INIT
   Enums, 46
PASS_INSERT
   Enums, 46
PASS_MOVE
   Enums, 46
PASS_OTP_GENERATE
   Enums, 46
PASS_REMOVE
   Enums, 46
PASS_SHOW
   Enums, 46
passExecutable
   SettingsConstants, 229
passGitInitNeeded
   MainWindow, 96
passOtpHandler

Generated by Doxygen
getQrencodeExecutable, 173
getRealPass, 174
getSaveState, 174
getSize, 175
getVersion, 176
getWebDavPassword, 176
getWebDavUrl, 177
getWebDavUser, 177
initExecutables, 177
isAddGPGId, 178
isAlwaysOnTop, 179
isAutoPull, 179
isAutoPush, 180
isAvoidCapitals, 181
isAvoidNumbers, 181
isHideContent, 182
isHideOnClose, 183
isHidePassword, 183
isLessRandom, 184
isMaximized, 185
isStartMinimized, 185
isTemplateAllFields, 186
isUseAutoclear, 187
isUseAutoclearPanel, 187
isUseGit, 188
isUseOtp, 189
isUsePass, 190
isUsePwgen, 190
isUseQrencode, 191
isUseSelection, 192
isUseSymbols, 192
isUseTemplate, 193
isUseTrayIcon, 194
isUseWebDav, 194
setAddGPGId, 195
setAlwaysOnTop, 195
setAutoclearPanelSeconds, 196
setAutoclearSeconds, 196
setAutoPull, 197
setAutoPush, 197
setAvoidCapitals, 198
setAvoidNumbers, 198
setClipboardType, 199
setGeometry, 199
setGitExecutable, 200
setGpgExecutable, 200
setHideContent, 201
setHideOnClose, 201
setHidePassword, 202
setLessRandom, 202
setMaximized, 202
setPassExecutable, 203
setPassStore, 203
setPassTemplate, 204
setPasswordChars, 204
setPasswordCharsSelection, 205
setPasswordConfiguration, 205
setPasswordLength, 206
setPos, 206
setProfile, 207
setProfiles, 207
setPwgenExecutable, 207
setQrencodeExecutable, 208
setSaveState, 208
setSize, 209
setStartMinimized, 210
setTemplateAllFields, 210
setUseAutoclear, 210
setUseAutoclearPanel, 211
setUseGit, 211
setUseOtp, 211
setUsePass, 212
setUsePwgen, 212
setUseQrencode, 213
setUseSelection, 213
setUseSymbols, 214
setUseTemplate, 214
setUseTrayIcon, 214
setUseWebDav, 215
setVersion, 215
setWebDavPassword, 216
setWebDavUrl, 216
setWebDavUser, 217
README.md, 272
RealPass, 217
~RealPass, 219
Copy, 220
GitInit, 220
GitPull, 220
GitPull_b, 221
GitPush, 221
Init, 221
Insert, 222
Move, 222
OtpGenerate, 223
RealPass, 219
Remove, 223
Show, 223
receiveMessage
SingleApplication, 240
reencryptPath
ImitatePass, 81
Remove
ImitatePass, 81
Pass, 123
RealPass, 223
restoreWindow
MainWindow, 97
savestate
SettingsConstants, 230
selected
PasswordConfiguration, 126
sendMessage
SingleApplication, 240
setAddGPGId
INDEX

QtPassSettings, 195
setAlwaysOnTop
QtPassSettings, 195
setAnimationDelay
QtProgressIndicator, 140
setAutoclearPanelSeconds
QtPassSettings, 196
setAutoclearSeconds
QtPassSettings, 196
setAutoPull
QtPassSettings, 197
setAutoPush
QtPassSettings, 197
setAvoidCapitals
QtPassSettings, 198
setAvoidNumbers
QtPassSettings, 198
setClipboardTimer
QtPass, 156
setClipBoardType
QtPassSettings, 199
setClippedText
QtPass, 156
setColor
QtProgressIndicator, 140
setDisplayWhenStopped
QtProgressIndicator, 141
setEnvironment
Executor, 69
setFreshStart
QtPass, 157
setGeometry
QtPassSettings, 199
setGitExecutable
QtPassSettings, 200
setGpgExecutable
QtPassSettings, 200
setHideContent
QtPassSettings, 201
setHideOnClose
QtPassSettings, 201
setHidePassword
QtPassSettings, 202
setLength
PasswordDialog, 130
setLessRandom
QtPassSettings, 202
setMaximized
QtPassSettings, 202
setModelAndStore
StoreModel, 247
setPass
PasswordDialog, 131
setPassExecutable
QtPassSettings, 203
setPassStore
QtPassSettings, 203
setPassTemplate
QtPassSettings, 204
setPassword
PasswordDialog, 132
setPasswordChars
QtPassSettings, 204
setPasswordCharsSelection
QtPassSettings, 205
setPasswordCharTemplate
PasswordDialog, 133
setPasswordConfiguration
ConfigDialog, 52
QtPassSettings, 205
setPasswordLength
QtPassSettings, 206
setPos
QtPassSettings, 206
setProfile
QtPassSettings, 207
setProfiles
QtPassSettings, 207
setPwgenExecutable
QtPassSettings, 207
setPwgenPath
ConfigDialog, 52
setQrencodeExecutable
QtPassSettings, 208
setSavestate
QtPassSettings, 208
setSize
QtPassSettings, 208
setStartMinimized
QtPassSettings, 210
setTemplate
PasswordDialog, 134
setTemplateAllFields
QtPassSettings, 210
setTextToCopy
QPushButtonAsQRCode, 145
QPushButtonWithClipboard, 150
SettingsConstants, 224
addGPGId, 225
alwaysOnTop, 225
autoclearPanelSeconds, 226
autoclearSeconds, 226
autoPull, 226
autoPush, 226
avoidCapitals, 226
avoidNumbers, 226
clipBoardType, 227
gitExecutable, 227
gpgExecutable, 227
clipBoardType, 227
geometry, 227
gitExecutable, 227
gpgExecutable, 227
clone, 227
groupMainwindow, 227
groupProfilers, 228
hideContent, 228
hideOnClose, 228
hidePassword, 228
Generated by Doxygen
INDEX

src/imitatepass.h, 279
src/keygendialog.cpp, 279
src/keygendialog.h, 280
src/mainwindow.cpp, 280
src/mainwindow.h, 281
src/pass.cpp, 282
src/pass.h, 283
src/passwordconfiguration.h, 283
src/passworddialog.cpp, 284
src/passworddialog.h, 285
src/qprogressindicator.cpp, 286
src/qprogressindicator.h, 286
src/qpushbuttonasqrcode.cpp, 287
src/qpushbuttonasqrcode.h, 288
src/qpushbuttonshowpassword.cpp, 288
src/qpushbuttonshowpassword.h, 289
src/qpushbuttonwithclipboard.cpp, 290
src/qpushbuttonwithclipboard.h, 290
src/qtpass.cpp, 291
src/qtpass.h, 292
src/qtpasssettings.cpp, 293
src/qtpasssettings.h, 293
src/reallpass.cpp, 294
src/reallpass.h, 294
src/settingsconstants.cpp, 295
src/settingsconstants.h, 295
src/simpletransaction.cpp, 296
src/simpletransaction.h, 296
src/singleapplication.cpp, 297
src/singleapplication.h, 298
src/storemodel.cpp, 298
src/storemodel.h, 300
src/trayicon.cpp, 300
src/trayicon.h, 301
src/userinfo.h, 302
src/usersdialog.cpp, 302
src/usersdialog.h, 303
src/util.cpp, 304
src/util.h, 305
startAnimation
  QProgressIndicator, 141
starting
  Executor, 69
startingExecuteWrapper
  Pass, 123
startMinimized
  SettingsConstants, 231
startReencryptPath
  ImitatePass, 82
  MainWindow, 100
statusMsg
  Pass, 124
stopAnimation
  QProgressIndicator, 141
StoreModel, 241
  canDrop_mimeData, 243
  data, 244
  drop_mimeData, 245
filterAcceptsRow, 245
flags, 246
lessThan, 246
mimeData, 247
mimeTypes, 247
setModelAndStore, 247
ShowThis, 248
StoreModel, 243
supportedDragActions, 249
supportedDropActions, 249
storemodel.cpp
  operator <<, 299
  operator >>, 299
supportedDragActions
  StoreModel, 249
supportedDropActions
  StoreModel, 249
takeValue
  NamedValues, 103
templateAll
  PasswordDialog, 134
templateAllFields
  SettingsConstants, 231
tests/auto/ui/tst_ui.cpp, 305
tests/auto/util/tst_util.cpp, 306
timerEvent
  QProgressIndicator, 142
transactionAdd
  simpleTransaction, 236
transactionEnd
  simpleTransaction, 236
transactionIsOver
  simpleTransaction, 236
transactionStart
  simpleTransaction, 237
TrayIcon, 250
getsAllocated, 251
iconActivated, 251
setVisible, 252
showHideParent, 253
showMessage, 253
TrayIcon, 251
tst_ui, 254
tst_util, 255
~tst_util, 256
cleanup, 256
init, 256
tst_util, 256
tst_util.cpp
  operator ==, 307
Ui, 46
updateEnv
  Pass, 124
useAutoclear
  ConfigDialog, 53
  SettingsConstants, 231
useAutoclearPanel

Generated by Doxygen
ConfigDialog, 53
SettingsConstants, 232

useClipboard
SettingsConstants, 232

useGit
ConfigDialog, 54
SettingsConstants, 232

useOtp
ConfigDialog, 54
SettingsConstants, 232

usePass
SettingsConstants, 232

usePwgen
ConfigDialog, 55
PasswordDialog, 135
SettingsConstants, 232

useQrencode
ConfigDialog, 56
SettingsConstants, 233

userDialog
MainWindow, 100

UserInfo, 257
  created, 259
  enabled, 259
  expiry, 259
  fullyValid, 258
  have_secret, 259
  isValid, 258
  key_id, 260
  marginallyValid, 258
  name, 260
  UserInfo, 257
  validity, 260

UsersDialog, 261
  ~UsersDialog, 262
    accept, 263
    closeEvent, 263
    keyPressEvent, 264

  UsersDialog, 262

useSelection
ConfigDialog, 56
SettingsConstants, 233

useSymbols
SettingsConstants, 233

useTemplate
ConfigDialog, 57
SettingsConstants, 233

useTrayIcon
ConfigDialog, 57
SettingsConstants, 233

useWebDav
SettingsConstants, 233

Util, 264
  checkConfig, 265
  copyDir, 265
  findBinaryInPath, 267
  findPasswordStore, 267
  getDir, 268

normalizeFolderPath, 269

validity
  UserInfo, 260

value
  NamedValue, 101

version
  SettingsConstants, 234

webDavPassword
  SettingsConstants, 234

webDavUrl
  SettingsConstants, 234

webDavUser
  SettingsConstants, 234

wizard
  ConfigDialog, 58