<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 QtPass</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Installation</td>
<td>1</td>
</tr>
<tr>
<td>1.2.1 Dependencies</td>
<td>1</td>
</tr>
<tr>
<td>1.2.2 From source</td>
<td>1</td>
</tr>
<tr>
<td>2 Change Log</td>
<td>3</td>
</tr>
<tr>
<td>3 Contributor Covenant Code of Conduct</td>
<td>25</td>
</tr>
<tr>
<td>4 Contributing</td>
<td>27</td>
</tr>
<tr>
<td>5 FAQ</td>
<td>29</td>
</tr>
<tr>
<td>6 QtPass</td>
<td>33</td>
</tr>
<tr>
<td>7 Todo List</td>
<td>37</td>
</tr>
<tr>
<td>8 Namespace Index</td>
<td>39</td>
</tr>
<tr>
<td>8.1 Namespace List</td>
<td>39</td>
</tr>
<tr>
<td>9 Hierarchical Index</td>
<td>41</td>
</tr>
<tr>
<td>9.1 Class Hierarchy</td>
<td>41</td>
</tr>
<tr>
<td>10 Class Index</td>
<td>43</td>
</tr>
<tr>
<td>10.1 Class List</td>
<td>43</td>
</tr>
<tr>
<td>11 File Index</td>
<td>45</td>
</tr>
<tr>
<td>11.1 File List</td>
<td>45</td>
</tr>
<tr>
<td>12 Namespace Documentation</td>
<td>47</td>
</tr>
<tr>
<td>12.1 Enums Namespace Reference</td>
<td>47</td>
</tr>
<tr>
<td>12.1.1 Detailed Description</td>
<td>47</td>
</tr>
<tr>
<td>12.1.2 Enumeration Type Documentation</td>
<td>47</td>
</tr>
<tr>
<td>12.1.2.1 clipBoardType</td>
<td>47</td>
</tr>
<tr>
<td>12.1.2.2 PROCESS</td>
<td>48</td>
</tr>
<tr>
<td>12.2 Ui Namespace Reference</td>
<td>48</td>
</tr>
<tr>
<td>13 Class Documentation</td>
<td>49</td>
</tr>
<tr>
<td>13.1 ConfigDialog Class Reference</td>
<td>49</td>
</tr>
<tr>
<td>13.1.1 Detailed Description</td>
<td>50</td>
</tr>
<tr>
<td>13.1.2 Constructor &amp; Destructor Documentation</td>
<td>51</td>
</tr>
<tr>
<td>13.1.2.1 ConfigDialog()</td>
<td>51</td>
</tr>
<tr>
<td>13.1.2.2 ~ConfigDialog()</td>
<td>52</td>
</tr>
<tr>
<td>13.1.3 Member Function Documentation</td>
<td>53</td>
</tr>
<tr>
<td>13.1.3.1 closeEvent()</td>
<td>53</td>
</tr>
<tr>
<td>13.1.3.2 genKey()</td>
<td>53</td>
</tr>
</tbody>
</table>
13.5.3.9 finished ................................................. 71
13.5.3.10 setEnvironment() ................................... 71
13.5.3.11 starting .............................................. 71

13.6 FileContent Class Reference ................................ 72
  13.6.1 Detailed Description .................................. 72
  13.6.2 Member Function Documentation ...................... 72
      13.6.2.1 getNamedValues() ................................ 73
      13.6.2.2 getPassword() ..................................... 73
      13.6.2.3 getRemainingData() ............................... 74
      13.6.2.4 getRemainingDataForDisplay() ..................... 74
      13.6.2.5 parse() ........................................... 74

13.7 ImitatePass Class Reference ................................ 75
  13.7.1 Detailed Description .................................. 77
  13.7.2 Constructor & Destructor Documentation ............ 77
      13.7.2.1 ImitatePass() .................................... 77
      13.7.2.2 ~ImitatePass() .................................... 77
  13.7.3 Member Function Documentation ...................... 77
      13.7.3.1 Copy() ............................................ 78
      13.7.3.2 endReencryptPath ................................ 78
      13.7.3.3 executeWrapper() ................................ 78
      13.7.3.4 finished() ....................................... 79
      13.7.3.5 GitInit() ......................................... 80
      13.7.3.6 GitPull() ......................................... 80
      13.7.3.7 GitPull_b() ....................................... 80
      13.7.3.8 GitPush() ......................................... 81
      13.7.3.9 Init() ............................................. 81
      13.7.3.10 Insert() .......................................... 82
      13.7.3.11 Move() ........................................... 82
      13.7.3.12 OtpGenerate() .................................... 83
      13.7.3.13 reencryptPath() ................................ 83
      13.7.3.14 Remove() ......................................... 83
      13.7.3.15 Show() .......................................... 84
      13.7.3.16 startReencryptPath .............................. 84

13.8 KeygenDialog Class Reference ................................ 85
  13.8.1 Detailed Description .................................. 86
  13.8.2 Constructor & Destructor Documentation ............ 86
      13.8.2.1 KeygenDialog() .................................. 86
      13.8.2.2 ~KeygenDialog() .................................. 86
  13.8.3 Member Function Documentation ...................... 86
      13.8.3.1 closeEvent() ...................................... 86

13.9 MainWindow Class Reference ................................ 87
  13.9.1 Detailed Description .................................. 89
13.12.4.34 listKeys() [2/2] .......................................................... 123
13.12.4.35 Move() ................................................................. 124
13.12.4.36 OtpGenerate() ........................................................ 124
13.12.4.37 processErrorExit ................................................... 125
13.12.4.38 Remove() ............................................................... 125
13.12.4.39 Show() ................................................................. 125
13.12.4.40 startingExecuteWrapper ........................................... 126
13.12.4.41 statusMsg .............................................................. 126
13.12.4.42 updateEnv() ............................................................ 126
13.12.5 Member Data Documentation ........................................... 127
13.12.5.1 exec ................................................................. 127
13.13 PasswordConfiguration Struct Reference .................................. 127
13.13.1 Detailed Description .................................................... 127
13.13.2 Member Enumeration Documentation .................................. 127
13.13.2.1 characterSet ........................................................... 127
13.13.3 Constructor & Destructor Documentation ................................. 128
13.13.3.1 PasswordConfiguration() ........................................... 128
13.13.4 Member Data Documentation ........................................... 128
13.13.4.1 Characters ........................................................... 128
13.13.4.2 length ................................................................. 128
13.13.4.3 selected ............................................................... 129
13.14 PasswordDialog Class Reference ........................................... 129
13.14.1 Detailed Description .................................................... 130
13.14.2 Constructor & Destructor Documentation ................................ 130
13.14.2.1 PasswordDialog() [1/2] ............................................. 130
13.14.2.2 PasswordDialog() [2/2] ............................................. 131
13.14.2.3 ~PasswordDialog() ................................................... 132
13.14.3 Member Function Documentation ....................................... 132
13.14.3.1 getPassword() ....................................................... 132
13.14.3.2 setLength() ........................................................ 133
13.14.3.3 setPass ............................................................... 133
13.14.3.4 setPassword() ....................................................... 134
13.14.3.5 setPasswordCharTemplate() ...................................... 135
13.14.3.6 setTemplate() ....................................................... 136
13.14.3.7 templateAll() ....................................................... 136
13.14.3.8 usePwgen() ......................................................... 137
13.15 QProgressIndicator Class Reference ...................................... 138
13.15.1 Detailed Description .................................................... 139
13.15.2 Constructor & Destructor Documentation ................................ 139
13.15.2.1 QProgressIndicator() .............................................. 139
13.15.3 Member Function Documentation ....................................... 140
13.15.3.1 animationDelay() ................................................... 140
13.20.2.29 isAddGPGId() .................................................. 180
13.20.2.30 isAlwaysOnTop() ........................................... 181
13.20.2.31 isAutoPull() .................................................. 182
13.20.2.32 isAutoPush() .................................................. 182
13.20.2.33 isAvoidCapitals() .......................................... 183
13.20.2.34 isAvoidNumbers() ......................................... 184
13.20.2.35 isHideContent() ............................................. 184
13.20.2.36 isHideOnClose() ............................................. 185
13.20.2.37 isHidePassword() .......................................... 186
13.20.2.38 isLessRandom() ............................................. 186
13.20.2.39 isMaximized() ............................................... 187
13.20.2.40 isStartMinimized() ........................................ 188
13.20.2.41 isTemplateAllFields() ..................................... 188
13.20.2.42 isUseAutoclear() .......................................... 189
13.20.2.43 isUseAutoclearPanel() .................................... 190
13.20.2.44 isUseGit() ................................................... 190
13.20.2.45 isUseOtp() ................................................... 191
13.20.2.46 isUsePass() .................................................. 192
13.20.2.47 isUsePwgen() .............................................. 193
13.20.2.48 isUseQrencode() .......................................... 193
13.20.2.49 isUseSelection() .......................................... 194
13.20.2.50 isUseSymbols() ............................................ 194
13.20.2.51 isUseTemplate() ........................................... 195
13.20.2.52 isUseTrayIcon() ........................................... 196
13.20.2.53 isUseWebDav() ............................................ 197
13.20.2.54 setAddGPGId() ............................................. 197
13.20.2.55 setAlwaysOnTop() ........................................ 198
13.20.2.56 setAutoclearPanelSeconds() ............................... 198
13.20.2.57 setAutoclearSeconds() .................................... 199
13.20.2.58 setAutoPull() ............................................... 199
13.20.2.59 setAutoPush() ............................................... 200
13.20.2.60 setAvoidCapitals() ........................................ 200
13.20.2.61 setAvoidNumbers() ....................................... 200
13.20.2.62 setClipboardType() ....................................... 201
13.20.2.63 setGeometry() ............................................. 201
13.20.2.64 setGitExecutable() ........................................ 202
13.20.2.65 setGpgExecutable() ....................................... 202
13.20.2.66 setHideContent() ......................................... 203
13.20.2.67 setHideOnClose() ......................................... 203
13.20.2.68 setHidePassword() ....................................... 204
13.20.2.69 setLessRandom() ......................................... 204
13.20.2.70 setMaximized() ............................................ 205
13.25.1 StoreModel() ................................................. 245
13.25.3 Member Function Documentation ........................................ 245
  13.25.3.1 canDropMimeData() ...................................... 245
  13.25.3.2 data() .................................................. 246
  13.25.3.3 dropMimeData() ........................................ 247
  13.25.3.4 filterAcceptsRow() ..................................... 248
  13.25.3.5 flags() ................................................ 248
  13.25.3.6 lessThan() ............................................. 249
  13.25.3.7 mimeData() ............................................ 249
  13.25.3.8 mimeTypes() .......................................... 249
  13.25.3.9 setModelAndStore() .................................... 250
  13.25.3.10 ShowThis() ........................................... 250
  13.25.3.11 supportedDragActions() ................................ 251
  13.25.3.12 supportedDropActions() ................................ 251
13.26 TrayIcon Class Reference .............................................. 252
  13.26.1 Detailed Description ....................................... 253
  13.26.2 Constructor & Destructor Documentation ............................ 253
    13.26.2.1 TrayIcon() ........................................... 253
  13.26.3 Member Function Documentation ................................... 253
    13.26.3.1 getIsAllocated() ..................................... 253
    13.26.3.2 iconActivated ....................................... 254
    13.26.3.3 setVisible() ......................................... 254
    13.26.3.4 showHideParent ...................................... 255
    13.26.3.5 showMessage() ....................................... 255
13.27 tst_ui Class Reference ............................................... 256
  13.27.1 Detailed Description ....................................... 256
13.28 tst_util Class Reference ............................................. 258
  13.28.1 Detailed Description ....................................... 258
  13.28.2 Constructor & Destructor Documentation ............................ 258
    13.28.2.1 tst_util() ........................................... 258
    13.28.2.2 ∼tst_util() .......................................... 258
  13.28.3 Member Function Documentation ................................... 258
    13.28.3.1 cleanup .............................................. 258
    13.28.3.2 init ................................................ 258
13.29 UserInfo Struct Reference ............................................. 259
  13.29.1 Detailed Description ....................................... 259
  13.29.2 Constructor & Destructor Documentation ............................ 259
    13.29.2.1 UserInfo() ........................................... 260
  13.29.3 Member Function Documentation ................................... 260
    13.29.3.1 fullyValid() ........................................ 260
    13.29.3.2 isValid() ............................................ 260
    13.29.3.3 marginallyValid() .................................... 261
13.29.4 Member Data Documentation .................................................. 261
  13.29.4.1 created ................................................................. 261
  13.29.4.2 enabled ............................................................... 261
  13.29.4.3 expiry ................................................................. 261
  13.29.4.4 have_secret ........................................................... 262
  13.29.4.5 key_id ................................................................. 262
  13.29.4.6 name ................................................................. 262
  13.29.4.7 validity ............................................................... 262

13.30 UsersDialog Class Reference .................................................... 263
  13.30.1 Detailed Description .......................................................... 264

13.30.2 Constructor & Destructor Documentation ..................................... 264
  13.30.2.1 UsersDialog() .......................................................... 264
  13.30.2.2 ~UsersDialog() ......................................................... 265

13.30.3 Member Function Documentation ............................................. 265
  13.30.3.1 accept ................................................................. 265
  13.30.3.2 closeEvent() ........................................................ 265
  13.30.3.3 keyPressEvent() ..................................................... 266

13.31 Util Class Reference ............................................................... 266
  13.31.1 Detailed Description ........................................................ 267

13.31.2 Member Function Documentation ............................................. 267
  13.31.2.1 checkConfig() ......................................................... 267
  13.31.2.2 copyDir() ............................................................. 268
  13.31.2.3 findBinaryInPath() ................................................ 269
  13.31.2.4 findPasswordStore() ................................................. 270
  13.31.2.5 getDir() ............................................................. 270
  13.31.2.6 normalizeFolderPath() .............................................. 271

14 File Documentation ................................................................. 273
  14.1 CHANGELOG.md File Reference .................................................. 273
  14.2 CODE_OF_CONDUCT.md File Reference ....................................... 273
  14.3 CONTRIBUTING.md File Reference .............................................. 273
  14.4 FAQ.md File Reference .......................................................... 273
  14.5 main/main.cpp File Reference .................................................. 273
  14.5.1 Function Documentation .................................................... 274
    14.5.1.1 main() .................................................................. 274
  14.6 README.md File Reference .......................................................... 274
  14.7 src/configdialog.cpp File Reference .......................................... 274
  14.8 src/configdialog.h File Reference ............................................. 275
  14.9 src/debughelper.h File Reference ............................................. 275
  14.9.1 Macro Definition Documentation .......................................... 276
    14.9.1.1 dbg .................................................................. 277
  14.10 src/deselectabletreeview.h File Reference ................................... 277
<table>
<thead>
<tr>
<th>Section</th>
<th>File Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.11</td>
<td>src/enums.h</td>
</tr>
<tr>
<td>14.12</td>
<td>src/executor.cpp</td>
</tr>
<tr>
<td>14.13</td>
<td>src/executor.h</td>
</tr>
<tr>
<td>14.14</td>
<td>src/filecontent.cpp</td>
</tr>
<tr>
<td>14.15</td>
<td>src/filecontent.h</td>
</tr>
<tr>
<td>14.16</td>
<td>src/imitatepass.cpp</td>
</tr>
<tr>
<td>14.17</td>
<td>src/imitatepass.h</td>
</tr>
<tr>
<td>14.18</td>
<td>src/keygendialog.cpp</td>
</tr>
<tr>
<td>14.19</td>
<td>src/keygendialog.h</td>
</tr>
<tr>
<td>14.20</td>
<td>src/mainwindow.cpp</td>
</tr>
<tr>
<td>14.21</td>
<td>src/mainwindow.h</td>
</tr>
<tr>
<td>14.21.1</td>
<td>Macro Definition Documentation</td>
</tr>
<tr>
<td>14.21.1.1</td>
<td>SingleApplication</td>
</tr>
<tr>
<td>14.22</td>
<td>src/pass.cpp</td>
</tr>
<tr>
<td>14.23</td>
<td>src/pass.h</td>
</tr>
<tr>
<td>14.24</td>
<td>src/passwordconfiguration.h</td>
</tr>
<tr>
<td>14.25</td>
<td>src/passworddialog.cpp</td>
</tr>
<tr>
<td>14.26</td>
<td>src/passworddialog.h</td>
</tr>
<tr>
<td>14.27</td>
<td>src/qprogressindicator.cpp</td>
</tr>
<tr>
<td>14.28</td>
<td>src/qprogressindicator.h</td>
</tr>
<tr>
<td>14.29</td>
<td>src/qpushbuttonasqrcode.cpp</td>
</tr>
<tr>
<td>14.30</td>
<td>src/qpushbuttonasqrcode.h</td>
</tr>
<tr>
<td>14.31</td>
<td>src/qpushbuttonshowpassword.cpp</td>
</tr>
<tr>
<td>14.32</td>
<td>src/qpushbuttonshowpassword.h</td>
</tr>
<tr>
<td>14.33</td>
<td>src/qpushbuttonwithclipboard.cpp</td>
</tr>
<tr>
<td>14.34</td>
<td>src/qpushbuttonwithclipboard.h</td>
</tr>
<tr>
<td>14.35</td>
<td>src/qtpass.cpp</td>
</tr>
<tr>
<td>14.36</td>
<td>src/qtpass.h</td>
</tr>
<tr>
<td>14.37</td>
<td>src/qtpasssettings.cpp</td>
</tr>
<tr>
<td>14.38</td>
<td>src/qtpasssettings.h</td>
</tr>
<tr>
<td>14.39</td>
<td>src/reallpass.cpp</td>
</tr>
<tr>
<td>14.40</td>
<td>src/reallpass.h</td>
</tr>
<tr>
<td>14.41</td>
<td>src/settingsconstants.cpp</td>
</tr>
<tr>
<td>14.42</td>
<td>src/settingsconstants.h</td>
</tr>
<tr>
<td>14.43</td>
<td>src/simpletransaction.cpp</td>
</tr>
<tr>
<td>14.44</td>
<td>src/simpletransaction.h</td>
</tr>
<tr>
<td>14.45</td>
<td>src/singleapplication.cpp</td>
</tr>
<tr>
<td>14.46</td>
<td>src/singleapplication.h</td>
</tr>
<tr>
<td>14.47</td>
<td>src/storemodel.cpp</td>
</tr>
<tr>
<td>14.47.1</td>
<td>Function Documentation</td>
</tr>
<tr>
<td>14.47.1.1</td>
<td>operator &lt;&lt; ()</td>
</tr>
<tr>
<td>14.47.1.2</td>
<td>operator &gt;&gt; ()</td>
</tr>
</tbody>
</table>
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:

- `<tt>` ... `<tt>` included in password text  #542
- Markup tags are left in password and clipboard  #533
- Renaming passwords and directories fail  #487
- Will not run on Windows 10 1903 b18362.418  #486

Closed issues:

- Hide results on search  #551
- QtPass 1.3.2 freezes on macOS 10.15.6 when trying to display password  #544
- 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
• 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  #554 (weblate)
• Translations update from Weblate  #553 (weblate)
• Translations update from Weblate  #552 (weblate)
• Translations update from Weblate  #548 (weblate)
• Move MainWindow to the screen the cursor is on  #547 (JFK422)
• Translations update from Weblate  #541 (weblate)
• Translations update from Weblate  #535 (weblate)
• Fix issues with renaming passwords and moving folders  #532 (ChaoticEnigma)
• Translations update from Weblate  #531 (weblate)
• 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 (boppybibbles)
• Enable out-of-source build  #503 (boppybibbles)
• Use new stable version of install-qt-action.  #502 (boppybibbles)
• 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 $\{(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 (application 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 cheched 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 (Noettore)
- 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 passphrasse  #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 -lqtpass    #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)

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
- 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 stfp, 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 evaluates every 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
• I translated for Simplified Chinese. #208
• 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 doesnot exists, de.DE translation fixes and removed obsolete translations  #216 ( YoshiMan)

v1.1.5 (2016-10-19)

Full Changelog

Implemented enhancements:

- 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 ‘Å§’ 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 (svetlemodry)

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
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:

- 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

Implemented enhancements:

- Get PREFIX variable from environment #106

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
• 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 (dennisdegeef)

v1.0.0 (2015-08-01)

Full Changelog

Merged pull requests:

• Updating hungarian localisation  #76 (damnlie)
• added DE translations  #74 (Friedy)
• Develop  #52 (annejan)

v0.9.2 (2015-07-30)

Full Changelog

Closed issues:

• Yubikey Neo Pin entry not working properly on Ubuntu 15.04  #73
• Show expiration date in key setup  #70
• Initialising new repo's doesn't work correctly  #55
**v0.9.1 (2015-07-29)**

Full Changelog

Closed issues:

- Minimize on startup. #69
- tray icon in xfce #58
- Git intergration #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 wi 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:

- 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 37f17f3808c1c97bd72c165a530c67a4bf82edb? #45
• Signing of keys from user management #41

Merged pull requests:

• Provide more information in user list. #47 (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:

• Enable C++11 and use it to simplify loops. #46 (rdoeffinger)
• 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](http://www.passwordstore.org/)
Chapter 3

Contributor Covenant Code of Conduct

Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

Our Standards

Examples of behavior that contributes to creating a positive environment include:

• Using welcoming and inclusive language
• Being respectful of differing viewpoints and experiences
• Gracefully accepting constructive criticism
• Focusing on what is best for the community
• Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

• The use of sexualized language or imagery and unwelcome sexual attention or advances
• Public or private harassment
• Publishing others' private information, such as a physical or electronic address, without explicit permission
• Other conduct which could reasonably be considered inappropriate in a professional setting

Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.
Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at help@qtpass.org. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at https://www.contributor-covenant.org/version/1/4/code-of-conduct.html

For answers to common questions about this code of conduct, see https://www.contributor-covenant.org/faq
Chapter 4

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](https://semver.org/).
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 your 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.
**IRC**

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

**Gitter**

Or if you prefer to use [gitter](https://gitter.im).

**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 5

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:

```plaintext
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 you key only in gpg and not in gpg2

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 place 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

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 over-ruled 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 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:
  ```bash
  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/](https://www.qt.io/download/) install it and run:
  ```bash
  /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 6

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 (Qt 6 works too)
- 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 7

Todo List

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

**Member** ConfigDialog::wizard ()
make this thing more reliable.
Chapter 8

Namespace Index

8.1 Namespace List

Here is a list of all namespaces with brief descriptions:

- **Enums**
  - Enumerators for configuration and runtime items
  - Page 47

- **Ui**
  - Page 48
Chapter 9

Hierarchical Index

9.1 Class Hierarchy

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

dragAndDropInfo ................................................................. 62
dragAndDropInfoPasswordStore ........................................... 63
FileContent ........................................................................ 72
NamedValue ........................................................................ 103
PasswordConfiguration ......................................................... 127
QApplication 
  SingleApplication ........................................................... 239
QDialog 
  ConfigDialog ................................................................ 49
  KeygenDialog ................................................................ 85
  PasswordDialog .............................................................. 129
  UsersDialog .................................................................. 263
QList 
  NamedValues ................................................................. 104
QMainWindow 
  MainWindow .................................................................. 87
QObject 
  Executor ...................................................................... 64
  Pass ............................................................................. 106
    ImitatePass ............................................................... 75
    RealPass ....................................................................... 219
  QtPass .......................................................................... 152
tst_ui .............................................................................. 256
tst_util .......................................................................... 257
QPushButton 
  QPushButtonAsQRCode ................................................... 145
  QPushButtonShowPassword ............................................. 148
  QPushButtonWithClipboard ............................................ 150
QSettings 
  QtPassSettings ............................................................. 160
QSortFilterProxyModel 
  StoreModel ................................................................... 243
QTreeView 
  DeselectableTreeView .................................................... 60
QWidget
<table>
<thead>
<tr>
<th>Class/Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>QProgressIndicator</td>
<td>138</td>
</tr>
<tr>
<td>TrayIcon</td>
<td>252</td>
</tr>
<tr>
<td>SettingsConstants</td>
<td>226</td>
</tr>
<tr>
<td>simpleTransaction</td>
<td>237</td>
</tr>
<tr>
<td>ImitatePass</td>
<td>75</td>
</tr>
<tr>
<td>UserInfo</td>
<td>259</td>
</tr>
<tr>
<td>Util</td>
<td>266</td>
</tr>
</tbody>
</table>
Chapter 10

Class Index

10.1 Class List

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

- **ConfigDialog**
  The *ConfigDialog* handles the configuration interface ........................................ 49

- **DeselectableTreeView**
  The *DeselectableTreeView* class loosely based on [http://stackoverflow.com/questions/2761284/](http://stackoverflow.com/questions/2761284/) thanks to Yassir Ennazk ........................................ 60

- **dragAndDropInfo**
  Holds values to share between drag and drop on the password storage view ............ 62

- **dragAndDropInfoPasswordStore** ............................................................... 63

- **Executor**
  Executes external commands for handling password, git and other data ................. 64

- **FileContent** ............................................................................................... 72

- **ImitatePass**
  Imitates pass features when pass is not enabled or available ............................... 75

- **KeygenDialog**
  Handles GPG keypair generation ................................................................. 85

- **MainWindow**
  Does way too much, not only is it a switchboard, configuration handler and more, it's also the process-manager ................................................................. 87

- **NamedValue** .................................................................................................. 103

- **NamedValues**
  Mostly a list of *NamedValue* but also has a method to take a specific *NamedValue* pair out of the list ............................................. 104

- **Pass**
  Acts as an abstraction for pass or pass imitation .............................................. 106

- **PasswordConfiguration**
  Holds the Password configuration settings ...................................................... 127

- **PasswordDialog**
  PasswordDialog Handles the inserting and editing of passwords ......................... 129

- **QProgressIndicator**
  Lets an application display a progress indicator to show that a lengthy task is under way .... 138

- **QPushButtonAsQRCode**
  Stylish widget to display the field as QR Code .............................................. 145

- **QPushButtonShowPassword** ................................................................. 148

- **QPushButtonWithClipboard**
  Stylish widget to allow copying of password and account details ...................... 150
QtPass
Singleton that stores qtpass' settings, saves and loads config

RealPass
Wrapper for executing pass to handle the password-store

SettingsConstants
Table for the naming of configuration items

SimpleTransaction
Used for commandline intergration

StoreModel
The QSortFilterProxyModel for handling filesystem searches

TrayIcon
Handles the systemtray icon and menu

tst_ui
Our first unit test

tst_util
Our first unit test

UserInfo
Stores key info lines including validity, creation date and more

UsersDialog
Handles listing and editing of GPG users

Util
Some static utilities to be used elsewhere
Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

- main/main.cpp ................................................. 273
- src/configdialog.cpp .......................................... 274
- src/configdialog.h ........................................... 275
- src/debughelper.h ........................................... 276
- src/deselectabletreeview.h ................................. 277
- src/enums.h .................................................. 277
- src/executor.cpp ............................................. 278
- src/executor.h ................................................ 278
- src/filecontent.cpp .......................................... 279
- src/filecontent.h ............................................ 280
- src/imitatepass.cpp ......................................... 280
- src/imitatepass.h ........................................... 281
- src/keygendialog.cpp ........................................ 282
- src/keygendialog.h .......................................... 282
- src/mainwindow.cpp .......................................... 283
- src/mainwindow.h ............................................ 283
- src/pass.cpp .................................................. 285
- src/pass.h ..................................................... 285
- src/passwordconfiguration.h ............................... 286
- src/passworddialog.cpp ...................................... 286
- src/passworddialog.h ........................................ 287
- src/qprogressindicator.cpp ................................. 288
- src/qprogressindicator.h .................................... 288
- src/qpushbuttonasqrcode.cpp .............................. 289
- src/qpushbuttonasqrcode.h .................................. 290
- src/qpushbuttonshowpassword.cpp .......................... 290
- src/qpushbuttonshowpassword.h ......................... 291
- src/qpushbuttonwithclipboard.cpp ...................... 292
- src/qpushbuttonwithclipboard.h ......................... 292
- src/qtapp.cpp ............................................... 293
- src/qtapp.h .................................................. 294
- src/qtpasssettings.cpp ..................................... 295
- src/qtpasssettings.h ....................................... 295
- src/realpass.cpp ............................................. 296
- src/realpass.h ............................................... 296
src/settingsconstants.cpp .................................................. 297
src/settingsconstants.h .................................................... 297
src/simpletransaction.cpp .................................................. 298
src/simpletransaction.h ..................................................... 298
src/singleapplication.cpp ................................................. 299
src/singleapplication.h .................................................... 300
src/storemodel.cpp .......................................................... 300
src/storemodel.h ............................................................. 302
src/trayicon.cpp ............................................................. 302
src/trayicon.h ............................................................... 303
src/userinfo.h ............................................................... 304
src/usersdialog.cpp ........................................................ 304
src/usersdialog.h ........................................................... 305
src/util.cpp ................................................................. 306
src/util.h .................................................................... 307
tests/auto/ui/tst_ui.cpp ................................................... 307
tests/auto/util/tst_util.cpp .............................................. 308
Chapter 12

Namespace Documentation

12.1 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 }

12.1.1 Detailed Description

Enumerators for configuration and runtime items.

12.1.2 Enumeration Type Documentation

12.1.2.1 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>
12.1.2.2 PROCESS

enum Enums::PROCESS

Enumerator

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

Definition at line 16 of file enums.h.

12.2 Ui Namespace Reference
13.1 ConfigDialog Class Reference

The ConfigDialog handles the configuration interface.

#include <configdialog.h>

Inheritance diagram for ConfigDialog:

Collaboration diagram for 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 by 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 *event)`
  `ConfigDialog::closeEvent` close this window.

13.1.1 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.
13.1.2 Constructor & Destructor Documentation

13.1.2.1 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:

```
13.1.2.2 ~ConfigDialog()
```

ConfigDialog::~ConfigDialog ()
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]

13.1.3 Member Function Documentation

13.1.3.1 closeEvent()

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

ConfigDialog::closeEvent close this window.

Parameters

- **event**

Definition at line 702 of file configdialog.cpp.

13.1.3.2 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:

![Call Graph]

13.1.3.3 getPasswordConfiguration()

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

13.1.3.4 getProfiles()

QHash<QString, QString> ConfigDialog::getProfiles ( )
ConfigDialog::getProfiles return profile list.
Returns
Definition at line 485 of file configdialog.cpp.

13.1.3.5 setPasswordConfiguration()

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

![Caller Graph]
13.1.3.6 setPwgenPath()

```cpp
void ConfigDialog::setPwgenPath (QString pwgen )
```

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

**Parameters**

- **pwgen**

Definition at line 762 of file configdialog.cpp.

Here is the caller graph for this function:

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

13.1.3.7 useAutoclear()

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

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

**Parameters**

- **useAutoclear**

Definition at line 409 of file configdialog.cpp.

Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> ConfigDialog::useAutoclear
```
13.1.3.8 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:

```
ConfigDialog::ConfigDialog -> ConfigDialog::useAutoclearPanel
```

13.1.3.9 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:

```
ConfigDialog::ConfigDialog -> ConfigDialog::useGit
```
13.1.3.10 useOtp()

```cpp
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:

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

13.1.3.11 usePwgen()

```cpp
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:

![Caller Graph for usePwgen()](image)
13.1.3.12 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:

```
ConfigDialog::ConfigDialog -> ConfigDialog::useQrencode
```

13.1.3.13 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:

```
ConfigDialog::ConfigDialog -> ConfigDialog::useSelection
```
13.1.3.14  useTemplate()

```cpp
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:

![Caller Graph for useTemplate()](image)

13.1.3.15  useTrayIcon()

```cpp
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:

![Caller Graph for useTrayIcon()](image)
### 13.1.3.16  wizard()

```cpp
void ConfigDialog::wizard()
```

ConfigDialog::wizard first-time use wizard.

Todo make this thing more reliable.

Definition at line 576 of file configdialog.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

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

- `src/configdialog.h`
- `src/configdialog.cpp`

### 13.2  DeselectableTreeView Class Reference


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

Inheritance diagram for DeselectableTreeView:

![Inheritance Diagram](inheritance_diagram.png)
13.2 DeselectableTreeView Class Reference

Collaboration diagram for DeselectableTreeView:

```
Collaboration diagram for DeselectableTreeView:

DeselectableTreeView
QTreeView
```

Signals

- void emptyClicked ()
  
  emptyClicked event

Public Member Functions

- DeselectableTreeView (QWidget *parent)
  
  DeselectableTreeView standard constructor.

- virtual ~DeselectableTreeView ()
  
  ~DeselectableTreeView standard destructor

13.2.1 Detailed Description

The DeselectableTreeView class loosely based on http://stackoverflow.com/questions/2761284/ thanks to Yassir Ennazk.

Definition at line 14 of file deselectabletreeview.h.

13.2.2 Constructor & Destructor Documentation

13.2.2.1 DeselectableTreeView()

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

DeselectableTreeView standard constructor.
Parameters

parent

Definition at line 22 of file deselectabletreeview.h.

### 13.2.2.2 ~DeselectableTreeView()

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

~DeselectableTreeView standard destructor

Definition at line 26 of file deselectabletreeview.h.

### 13.2.3 Member Function Documentation

#### 13.2.3.1 emptyClicked

void DeselectableTreeView::emptyClicked () [signal]

emptyClicked event

Here is the caller graph for this function:

![Caller Graph](image)

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

- src/deselectabletreeview.h

### 13.3 dragAndDropInfo Struct Reference

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

```cpp
#include <storemodel.h>
```
13.3.1 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

13.4 dragAndDropInfoPasswordStore Struct Reference

#include <storemodel.h>

Public Attributes

- bool isDir
- bool isFile
- QString path

13.4.1 Detailed Description

Definition at line 46 of file storemodel.h.

13.4.2 Member Data Documentation

13.4.2.1 isDir

bool dragAndDropInfoPasswordStore::isDir

Definition at line 47 of file storemodel.h.

13.4.2.2 isFile

bool dragAndDropInfoPasswordStore::isFile

Definition at line 48 of file storemodel.h.
13.4.2.3  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

13.5  Executor Class Reference

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

#include <executor.h>

Inheritance diagram for Executor:

```
 QObject
  
  Executor
  
  Executor
  
  QObject
```

Collaboration diagram for Executor:
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 execute 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)
  
  Executor::executeBlocking blocking version of the executor.

13.5.1 Detailed Description

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

Definition at line 12 of file executor.h.

13.5.2 Constructor & Destructor Documentation

13.5.2.1 Executor()

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

Executor::Executor executes external applications.
13.5.3 Member Function Documentation

13.5.3.1 cancelNext()

```cpp
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 230 of file executor.cpp.

13.5.3.2 error

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

**error** signal that is emited 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>

13.5.3.3 execute() [1/4]

```cpp
void Executor::execute ( int id,
```

Definition at line 16 of file executor.cpp.
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>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>args</td>
<td>readStdout</td>
</tr>
<tr>
<td></td>
<td>readStderr</td>
</tr>
</tbody>
</table>

Definition at line 65 of file executor.cpp.

Here is the caller graph for this function:

![Caller Graph](image)

13.5.3.4 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>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>args</td>
<td>input</td>
</tr>
<tr>
<td></td>
<td>readStdout</td>
</tr>
<tr>
<td></td>
<td>readStderr</td>
</tr>
</tbody>
</table>

Generated by Doxygen
Definition at line 94 of file executor.cpp.

Here is the call graph for this function:

```
Executor::execute Executor::execute
```

13.5.3.5 execute() [3/4]

```cpp
definition at line 79 of file executor.cpp.

Here is the call graph for this function:

```
Executor::execute Executor::execute
```

**Parameters**

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

Definition at line 79 of file executor.cpp.

Here is the call graph for this function:
13.5.3.6 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 110 of file executor.cpp.

13.5.3.7 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 210 of file executor.cpp.
Here is the call graph for this function:

```
Executor::executeBlocking
```

### 13.5.3.8 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 168 of file executor.cpp.

Here is the caller graph for this function:
13.5.3.9 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>

13.5.3.10 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 220 of file executor.cpp.

Here is the caller graph for this function:

```
```

13.5.3.11 starting

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

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

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

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

### 13.6 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`.

#### 13.6.1 Detailed Description

Definition at line 25 of file filecontent.h.

#### 13.6.2 Member Function Documentation
13.6.2.1  getNamedValues()

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

Returns

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

Definition at line 38 of file filecontent.cpp.

Here is the caller graph for this function:

![Caller Graph for getNamedValues()](attachment:graph.png)

13.6.2.2  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:

![Caller Graph for getPassword()](attachment:graph.png)
13.6.2.3 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:

![Caller Graph]

13.6.2.4 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:

![Caller Graph]

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

Returns

Definition at line 7 of file `filecontent.cpp`.

Here is the caller graph for this function:

![Caller Graph]

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

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

## 13.7 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
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 overrided so that every execution is a transaction

Additional Inherited Members

13.7.1 Detailed Description

Imitates pass features when pass is not enabled or available.

Definition at line 11 of file imitatepass.h.

13.7.2 Constructor & Destructor Documentation

13.7.2.1 ImitatePass()

ImitatePass::ImitatePass ( ) [default]

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

13.7.2.2 ~ImitatePass()

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

Definition at line 48 of file imitatepass.h.

13.7.3 Member Function Documentation

Generated by Doxygen
13.7.3.1 Copy()

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

Implements Pass.

Definition at line 421 of file imitatepass.cpp.

Here is the call graph for this function:

```
  ImitatePass::Copy  QtPassSettings::isUseGit  QtPassSettings::getInstance
```

13.7.3.2 endReencryptPath

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

13.7.3.3 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.
13.7 ImitatePass Class Reference

Definition at line 520 of file imitatepass.cpp.

Here is the call graph for this function:

![Call graph for ImitatePass::executeWrapper](image)

13.7.3.4 finished()

```cpp
defined Pass::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>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>exitCode</td>
</tr>
<tr>
<td>out</td>
</tr>
<tr>
<td>err</td>
</tr>
</tbody>
</table>

Reimplemented from Pass.

Definition at line 482 of file imitatepass.cpp.

Here is the call graph for this function:

![Call graph for ImitatePass::finished](image)
13.7.3.5 GitInit()

void ImitatePass::GitInit () [virtual]

ImitatePass::GitInit git init wrapper.

Implements Pass.

Definition at line 36 of file imitatepass.cpp.

13.7.3.6 GitPull()

void ImitatePass::GitPull ( ) [virtual]

ImitatePass::GitPull git pull wrapper.

Implements Pass.

Definition at line 43 of file imitatepass.cpp.

13.7.3.7 GitPull_b()

void ImitatePass::GitPull_b ( ) [virtual]

ImitatePass::GitPull_b git pull wrapper.

Implements Pass.

Definition at line 48 of file imitatepass.cpp.

Here is the call graph for this function:

```plaintext
ImitatePass::GitPull_b  QtPassSettings::getGitExecutable  QtPassSettings::getInstance
```
13.7.3.8 GitPush()

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

**ImitatePass::GitPush** git init wrapper.

Implements **Pass**.

Definition at line 55 of file imitatepass.cpp.

Here is the call graph for this function:

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

13.7.3.9 Init()

```cpp
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 168 of file imitatepass.cpp.

Here is the call graph for this function:

![Call Graph for ImitatePass::Init](call_graph.png)
13.7.3.10  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>file</td>
<td>file to be created</td>
</tr>
<tr>
<td>newValue</td>
<td>value to be stored in file</td>
</tr>
<tr>
<td>overwrite</td>
<td>whether to overwrite existing file</td>
</tr>
</tbody>
</table>

Implements **Pass**.

Definition at line 90 of file imitatepass.cpp.

Here is the call graph for this function:

---

13.7.3.11  Move()

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

Implements **Pass**.

Definition at line 347 of file imitatepass.cpp.

Here is the call graph for this function:
13.7.3.12 OtpGenerate()

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

ImitatePass::OtpGenerate generates an otp code.

Implements Pass.

Definition at line 74 of file imitatepass.cpp.

13.7.3.13 reencryptPath()

```cpp
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 244 of file imitatepass.cpp.

Here is the call graph for this function:

```
ImitatePass::reencryptPath QtPassSettings::isAutoPull QtPassSettings::getInstance
```

13.7.3.14 Remove()

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

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

Implements Pass.

Definition at line 138 of file imitatepass.cpp.
Here is the call graph for this function:

13.7.3.15  Show()

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

*ImitatePass::Show* shows content of file.

Implements *Pass*.

Definition at line 64 of file imitatepass.cpp.

Here is the call graph for this function:

13.7.3.16  startReencryptPath

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

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

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

Handles GPG keypair generation.

#include <keygendialog.h>

Inheritance diagram for KeygenDialog:

![Inheritance Diagram]

Collaboration diagram for KeygenDialog:

![Collaboration Diagram]

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.
13.8.1 Detailed Description

Handles GPG keypair generation.

Definition at line 16 of file keygendialog.h.

13.8.2 Constructor & Destructor Documentation

13.8.2.1 KeygenDialog()

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

KeygenDialog::KeygenDialog basic constructor.

Parameters

parent

Definition at line 16 of file keygendialog.cpp.

13.8.2.2 ~KeygenDialog()

KeygenDialog::~KeygenDialog ( )

KeygenDialog::~KeygenDialog even more basic destructor.

Definition at line 25 of file keygendialog.cpp.

13.8.3 Member Function Documentation

13.8.3.1 closeEvent()

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

KeygenDialog::closeEvent we are done here.
Parameters

\begin{verbatim}
event
\end{verbatim}

Definition at line 183 of file keygendialog.cpp.

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

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

### 13.9 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:

![Inheritance diagram for MainWindow](image)

Collaboration diagram for MainWindow:

![Collaboration diagram for MainWindow](image)
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.

13.9.1 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.

13.9.2 Constructor & Destructor Documentation

13.9.2.1 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:

13.9.2.2  ~MainWindow()

MainWindow::~MainWindow ()

Definition at line 127 of file mainwindow.cpp.

13.9.3  Member Function Documentation

13.9.3.1  changeEvent()

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

MainWindow::changeEvent sets focus to the search box.

Parameters

  event
13.9.3.2 cleanKeygenDialog()

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

Definition at line 204 of file mainwindow.cpp.

13.9.3.3 closeEvent()

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

MainWindow::closeEvent hide or quit.

Parameters

- `event`

Definition at line 821 of file mainwindow.cpp.

Here is the call graph for this function:

13.9.3.4 config()

```cpp
void MainWindow::config ()
```

Generated by Doxygen
**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](call_graph.png)

Here is the caller graph for this function:

![Caller Graph](caller_graph.png)

### 13.9.3.5 critical

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

**MainWindow::critical** critical message popup wrapper.

**Parameters**

- **title**
- **msg**

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

void MainWindow::deselect ( ) [slot]

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

Definition at line 352 of file mainwindow.cpp.

Here is the call graph for this function:

![Call Graph for deselect](call_graph_deselect)

Here is the caller graph for this function:

![Caller Graph for deselect](caller_graph_deselect)

13.9.3.7  endReencryptPath

void MainWindow::endReencryptPath ( ) [slot]

`MainWindow::endReencryptPath` re-enable ui elements.

Definition at line 1162 of file mainwindow.cpp.

Here is the call graph for this function:

![Call Graph for endReencryptPath](call_graph_endReencryptPath)

Generated by Doxygen
13.9.3.8 eventFilter()

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 846 of filemainwindow.cpp.

13.9.3.9 executeWrapperStarted

void MainWindow::executeWrapperStarted ( ) [slot]

Definition at line 362 of filemainwindow.cpp.

Here is the call graph for this function:

13.9.3.10 flashText()

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

Definition at line 209 of filemainwindow.cpp.
### 13.9.3.11 generateGPGKeyPair

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

Here is the caller graph for this function:

![Caller Graph for generateGPGKeyPair](image)

### 13.9.3.12 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 732 of filemainwindow.cpp.

Here is the caller graph for this function:

![Caller Graph for generateKeyPair](image)

### 13.9.3.13 getCurrentTreeViewIndex()

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

Definition at line 200 of filemainwindow.cpp.
### 13.9.3.14 `getKeygenDialog()`

QDialog* MainWindow::getKeygenDialog () [inline]

Definition at line 58 of file mainwindow.h.

### 13.9.3.15 `keyPressEvent()`

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

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

**Parameters**

- `event`

Definition at line 860 of file mainwindow.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

### 13.9.3.16 `messageAvailable`

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

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

**Parameters**

- `message`

Definition at line 715 of file mainwindow.cpp.
Here is the caller graph for this function:

![Caller Graph](image)

### 13.9.3.17 on_treeView_clicked

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

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

#### Parameters

- **index**

Definition at line 318 of file `mainwindow.cpp`.

Here is the call graph for this function:

![Call Graph](image)

Here is the caller graph for this function:

![Caller Graph](image)
13.9.3.18 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](call_graph_onpush.png)

13.9.3.19 passGitInitNeeded

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

Here is the caller graph for this function:

![Caller graph for passGitInitNeeded function](call_graph_passgitinitneeded.png)

13.9.3.20 passOtpHandler

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

Definition at line 415 of file mainwindow.cpp.

Here is the call graph for this function:

![Call graph for passOtpHandler function](call_graph_passotphandler.png)
13.9.3.21  passShowHandler

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

Definition at line 369 of file mainwindow.cpp.

Here is the call graph for this function:

13.9.3.22  passShowHandlerFinished

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

Here is the caller graph for this function:
### 13.9.3.23 `restoreWindow()`

```cpp
void MainWindow::restoreWindow ( )
```

Definition at line 464 of file mainwindow.cpp.

Here is the call graph for this function:

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

### 13.9.3.24 `setUiElementsEnabled()`

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

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

**Parameters**

- `state`

Definition at line 448 of file mainwindow.cpp.
13.9.3.25  showStatusMessage

void MainWindow::showStatusMessage (  
    QString msg,  
    int timeout = 2000 ) [slot]

Displays message in status bar.

Parameters

<table>
<thead>
<tr>
<th>msg</th>
<th>text to be displayed</th>
</tr>
</thead>
<tbody>
<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:
### 13.9.3.26 startReencryptPath

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

**MainWindow::startReencryptPath** disable ui elements and treeview.

Definition at line 1154 of file mainwindow.cpp.

Here is the call graph for this function:

![Call Graph](call-graph.png)

### 13.9.3.27 userDialog()

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

**MainWindow::userDialog** see MainWindow::onUsers()

**Parameters**

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

Definition at line 688 of file mainwindow.cpp.

Here is the caller graph for this function:

![Caller Graph](caller-graph.png)

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

- `src/mainwindow.h`
- `src/mainwindow.cpp`
13.10 NamedValue Struct Reference

#include <filecontent.h>

Public Attributes

- QStringList name
- QStringList value

13.10.1 Detailed Description

Definition at line 8 of file filecontent.h.

13.10.2 Member Data Documentation

13.10.2.1 name

QString NamedValue::name

Definition at line 9 of file filecontent.h.

13.10.2.2 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
13.11 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.

#include <filecontent.h>

Inheritance diagram for NamedValues:

```
#include <filecontent.h>

Inheritance diagram for NamedValues:
```

Collaboration diagram for NamedValues:

```
#include <filecontent.h>

Collaboration diagram for NamedValues:
```

### Public Member Functions

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

### 13.11.1 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.
13.11.2 Constructor & Destructor Documentation

13.11.2.1 NamedValues() [1/2]

NamedValues::NamedValues ( )

Definition at line 54 of file filecontent.cpp.

13.11.2.2 NamedValues() [2/2]

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

Definition at line 56 of file filecontent.cpp.

13.11.3 Member Function Documentation

13.11.3.1 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
13.12 Pass Class Reference

Acts as an abstraction for pass or pass imitation.

#include <pass.h>

Inheritance diagram for Pass:

![Inheritance Diagram]

Collaboration diagram for Pass:

![Collaboration Diagram]

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 **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**
• virtual QString **Generate_b (unsigned int length, const QString &charset)**
  
  *Pass::Generate use either pwgen or internal password generator.*

• void **GenerateGPGKeys (QString batch)**
  
  *Pass::GenerateGPGKeys internal gpg keypair generator.*

• QList <UserInfo > **listKeys (QStringList keystrings, 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

13.12.1 Detailed Description

Acts as an abstraction for pass or pass imitation.

Definition at line 25 of file pass.h.

13.12.2 Member Typedef Documentation

13.12.2.1 PROCESS

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

Definition at line 34 of file pass.h.

13.12.3 Constructor & Destructor Documentation
13.12.3.1 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:

13.12.3.2 ~Pass()

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

Definition at line 40 of file pass.h.

13.12.4 Member Function Documentation

13.12.4.1 boundedRandom()

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

Definition at line 309 of file pass.cpp.

Here is the caller graph for this function:
13.12.4.2 Copy()

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

Implemented in RealPass, and ImitatePass.

Here is the caller graph for this function:

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

13.12.4.3 critical

void Pass::critical (QString , QString ) [signal]

Here is the caller graph for this function:

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

13.12.4.4 error

void Pass::error (QProcess::ProcessError ) [signal]
### 13.12.4.5 executeWrapper() [1/2]

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:

![Caller Graph](image1)

### 13.12.4.6 executeWrapper() [2/2]

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:

![Call Graph](image2)
13.12.4.7 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>id</th>
<th>id of Pass process that was scheduled and finished</th>
</tr>
</thead>
<tbody>
<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:

![Caller Graph](#)

13.12.4.8 finishedAny

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

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

Here is the caller graph for this function:

![Caller graph for finishedCopy](image)

13.12.4.10 finishedGenerate

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

13.12.4.11 finishedGenerateGPGKeys

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

13.12.4.12 finishedGitInit

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

Here is the caller graph for this function:

![Caller graph for finishedGitInit](image)
13.12.4.13 finishedGitPull

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

Here is the caller graph for this function:

![Caller Graph for finishedGitPull]

13.12.4.14 finishedGitPush

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

Here is the caller graph for this function:

![Caller Graph for finishedGitPush]

13.12.4.15 finishedInit

void Pass::finishedInit ( 
    const QString &, 
    const QString & ) [signal]
Here is the caller graph for this function:

```
ImitatePass::finished

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

13.12.4.16 finishedInsert

```cpp
template<typename T>
void Pass::finishedInsert ( const QString & str, const QItem & item ) [signal]
```

Here is the caller graph for this function:

```
ImitatePass::finished

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

13.12.4.17 finishedMove

```cpp
template<typename T>
void Pass::finishedMove ( const QString & str, const QItem & item ) [signal]
```

Here is the caller graph for this function:

```
ImitatePass::finished

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Generated by Doxygen
13.12.4.18 finishedOtpGenerate

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

Here is the caller graph for this function:

![Caller Graph for finishedOtpGenerate](image)

13.12.4.19 finishedRemove

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

Here is the caller graph for this function:

![Caller Graph for finishedRemove](image)

13.12.4.20 finishedShow

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

Here is the caller graph for this function:

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

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

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

Parameters

| length  | of the desired password |
| charset | to use for generation   |

Returns

the password

Definition at line 69 of file pass.cpp.

Here is the call graph for this function:

Get the call graph for this function

13.12.4.22 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
```

## 13.12.4.23 generateRandomPassword()

```cpp
definition_at_line_335_of_file_pass.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
```

Here is the caller graph for this function:

```
Pass::Generate_b
```

## 13.12.4.24 getRecipientList()

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

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

| for_file | which file (folder) would you like recipients for |

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:

```
Pass::getRecipientList
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Pass::getRecipientString</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>UsersDialog::UsersDialog</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Pass::getRecipientList</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>ImitatePass::Insert</td>
</tr>
</tbody>
</table>
```

13.12.4.25 `getRecipientString()`

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

Pass::getRecipientString formatted string for use with GPG.

Parameters

| for_file | which file (folder) would you like recipients for |
| separator | forating separator eg: " -r " |
| count | |
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:

13.12.4.26 GitInit()

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

Implemented in RealPass, and ImitatePass.

13.12.4.27 GitPull()

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

Implemented in RealPass, and ImitatePass.

13.12.4.28 GitPull_b()

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

Implemented in RealPass, and ImitatePass.
13.12.4.29 GitPush()

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

Implemented in RealPass, and ImitatePass.

Here is the caller graph for this function:

![Caller Graph for GitPush()](image1)

13.12.4.30 init()

void Pass::init ( )

Definition at line 45 of file pass.cpp.

Here is the call graph for this function:

![Call Graph for init()](image2)

Here is the caller graph for this function:

![Caller Graph for init()](image3)
13.12.4.31 Init()

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

Implemented in RealPass, and ImitatePass.

Here is the caller graph for this function:

13.12.4.32 Insert()

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

Implemented in RealPass, and ImitatePass.

13.12.4.33 listKeys() [1/2]

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

Pass::listKeys list users.

Parameters

<table>
<thead>
<tr>
<th>keystring</th>
<th>secret</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>list private keys</td>
</tr>
</tbody>
</table>

Returns

QList<UserInfo> users

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

![Call Graph](call_graph.png)

### 13.12.4.34 `listKeys()` [2/2]

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

**Pass::listKeys** list users.

**Parameters**

<table>
<thead>
<tr>
<th>keystrings</th>
<th>list private keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret</td>
<td></td>
</tr>
</tbody>
</table>

**Returns**

`QList<UserInfo>` users

Definition at line 128 of file `pass.cpp`.

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

![Call Graph](call_graph.png)

### 13.12.4.35 Move()

```cpp
class Pass {
public:
  virtual void Move(const QString srcDir, const QString dest, const bool force = false) = 0;
};
```

Implemented in `RealPass`, and `ImitatePass`.

Here is the caller graph for this function:

![Call Graph](call_graph.png)

### 13.12.4.36 OtpGenerate()

```cpp
class Pass {
public:
  virtual void OtpGenerate(const QString file) = 0;
};
```

Implemented in `RealPass`, and `ImitatePass`. 
13.12.4.37 processErrorExit

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

Here is the caller graph for this function:

13.12.4.38 Remove()

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

Implemented in RealPass, and ImitatePass.

13.12.4.39 Show()

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

Implemented in RealPass, and ImitatePass.

Here is the caller graph for this function:
13.12.4.40 startingExecuteWrapper

```cpp
void Pass::startingExecuteWrapper ( ) [signal]
```

Here is the caller graph for this function:

![Caller Graph for startingExecuteWrapper](image)

13.12.4.41 statusMsg

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

13.12.4.42 updateEnv()

```cpp
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:

![Call Graph for updateEnv](image)

Here is the caller graph for this function:

![Caller Graph for updateEnv](image)
13.13 PasswordConfiguration Struct Reference

13.12.5 Member Data Documentation

13.12.5.1 exec

Executor Pass::exec [protected]
Definition at line 32 of file pass.h.
The documentation for this class was generated from the following files:

- src/pass.h
- src/pass.cpp

13.13 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.
- QStringList Characters [CHARSETS_COUNT]
  
The different character sets.

13.13.1 Detailed Description

Holds the Password configuration settings.
Definition at line 10 of file passwordconfiguration.h.

13.13.2 Member Enumeration Documentation

13.13.2.1 characterSet

enum PasswordConfiguration::characterSet

The selected character set.
## Enumerator

<table>
<thead>
<tr>
<th>ALLCHARS</th>
<th>ALPHABETICAL</th>
<th>ALPHANUMERIC</th>
<th>CUSTOM</th>
<th>CHARSET_COUNT</th>
</tr>
</thead>
</table>

Definition at line 14 of file passwordconfiguration.h.

### 13.13.3 Constructor & Destructor Documentation

#### 13.13.3.1 PasswordConfiguration()

PasswordConfiguration::PasswordConfiguration () [inline]

Definition at line 29 of file passwordconfiguration.h.

### 13.13.4 Member Data Documentation

#### 13.13.4.1 Characters

QString PasswordConfiguration::Characters[CHARSETS_COUNT]

The different character sets.

Definition at line 28 of file passwordconfiguration.h.

#### 13.13.4.2 length

int PasswordConfiguration::length

Length of the password.

Definition at line 24 of file passwordconfiguration.h.
13.14.3  selected

enum PasswordConfiguration::characterSet PasswordConfiguration::selected

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

- `src/passwordconfiguration.h`

13.14  PasswordDialog Class Reference

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

#include <passworddialog.h>

Inheritance diagram for PasswordDialog:

![Inheritance Diagram]

Collaboration diagram for PasswordDialog:

![Collaboration Diagram]

**Public Slots**

- void **setPass** (const QString &output)

Generated by Doxygen
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** ()
  
  PasswordDialog::~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.

13.14.1 Detailed Description

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

Includes templated views.

Definition at line 20 of file passworddialog.h.

13.14.2 Constructor & Destructor Documentation

13.14.2.1 **PasswordDialog()** [1/2]

PasswordDialog::PasswordDialog ( const PasswordConfiguration & passConfig,
                                 QWidget * parent = nullptr ) [explicit]

PasswordDialog::PasswordDialog basic constructor.

Parameters

<table>
<thead>
<tr>
<th>passConfig</th>
<th>configuration constant</th>
</tr>
</thead>
<tbody>
<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:

13.14.2.2 PasswordDialog() [2/2]

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

PasswordDialog::PasswordDialog complete constructor.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></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:

### 13.14.2.3 ~PasswordDialog()

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

### 13.14.3 Member Function Documentation

#### 13.14.3.1 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.
13.14.3.2 setLength()

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

PasswordDialog::setLength password length.

Definition at line 208 of file passworddialog.cpp.

Here is the caller graph for this function:

![Caller Graph for setLength](image)

13.14.3.3 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 for setPass](image)
Here is the caller graph for this function:

```
PasswordDialog::Password
   Dialog
```

### 13.14.3.4 setPassword()

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

Sets content in the password field in the interface.

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

**Parameters**

<table>
<thead>
<tr>
<th>password</th>
<th>the password as a QString</th>
</tr>
</thead>
</table>

**See also**

*getPassword*

**Parameters**

<table>
<thead>
<tr>
<th>password</th>
<th></th>
</tr>
</thead>
</table>

Definition at line 124 of file passworddialog.cpp.
Here is the call graph for this function:

```
PasswordDialog::setPassword
FileContent::getNamedValues
FileContent::getPassword
FileContent::getRemainingData
FileContent::parse
NamedValues::takeValue
```

Here is the caller graph for this function:

```
PasswordDialog::Password
Dialog
PasswordDialog::setPass
PasswordDialog::setPassword
```

### 13.14.3.5 setPasswordCharTemplate()

```cpp
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::PasswordDialog

PasswordDialog::setPassword

13.14.3.6 setTemplate()

void 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>
<tr>
<td>rawFields</td>
<td></td>
</tr>
</tbody>
</table>

Definition at line 174 of file passworddialog.cpp.

Here is the caller graph for this function:

PasswordDialog::PasswordDialog

PasswordDialog::setTemplate

13.14.3.7 templateAll()

void PasswordDialog::templateAll ( 
  bool templateAll )

PasswordDialog::templateAll basic setter for use in PasswordDialog::setPassword templating all tokenisable lines.
13.14.3.8 usePwgen()

void PasswordDialog::usePwgen ( 
    bool usePwgen )

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

Definition at line 224 of file passworddialog.cpp.

Here is the caller graph for this function:

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

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

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

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

Inheritance diagram for QProgressIndicator:

![Inheritance Diagram](#)

Collaboration diagram for QProgressIndicator:

![Collaboration Diagram](#)

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 `setDisplayWhenStopped(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..

13.15.1 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.

13.15.2 Constructor & Destructor Documentation

13.15.2.1 **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.

13.15.3 Member Function Documentation

13.15.3.1 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

- setAnimationDelay

Definition at line 40 of file qprogressindicator.h.

13.15.3.2 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.

13.15.3.3 heightForWidth()

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

QProgressIndicator::heightForWidth square ratio.
Parameters

\texttt{w} \quad \text{requested width}

Returns

\texttt{w} \quad \text{returned height}

Definition at line 70 of file qprogressindicator.cpp.

13.15.3.4 \hspace{1em} \textbf{isAnimated()}

\begin{verbatim}
bool QProgressIndicator::isAnimated ( ) const
\end{verbatim}

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

Returns

\textbf{Animation state.}

See also

\texttt{startAnimation stopAnimation}

Definition at line 15 of file qprogressindicator.cpp.

Here is the caller graph for this function:

QProgressIndicator \:::paintEvent \rightarrow QProgressIndicator \:::isAnimated

13.15.3.5 \hspace{1em} \textbf{isDisplayedWhenStopped()}

\begin{verbatim}
bool QProgressIndicator::isDisplayedWhenStopped ( ) const
\end{verbatim}

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

Returns

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

See also

\texttt{setDisplayWhenStopped}

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

```cpp
define QPaintEvent { 
    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:

![Call graph for QProgressIndicator::paintEvent](call_graph.png)

13.15.3.7 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.

13.15.3.8 setColor

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

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

See also

`color`

Definition at line 53 of file qprogressindicator.cpp.

13.15.3.9 setDisplayedWhenStopped

```cpp
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.

13.15.3.10 sizeHint()

```cpp
QSize QProgressIndicator::sizeHint () const [virtual]
```

`QProgressIndicator::sizeHint` default minimum size.

Returns

`QSize(20, 20)`

Definition at line 63 of file qprogressindicator.cpp.

13.15.3.11 startAnimation

```cpp
void QProgressIndicator::startAnimation () [slot]
```

Starts the spin animation.

See also

`stopAnimation isAnimated`

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

void QProgressIndicator::stopAnimation ( ) [slot]

Stops the spin animation.

See also

startAnimation isAnimated

Definition at line 34 of file qprogressindicator.cpp.

13.15.3.13 timerEvent()

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

QProgressIndicator::timerEvent do the actual animation.

Definition at line 75 of file qprogressindicator.cpp.

13.15.4 Property Documentation

13.15.4.1 color

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

QProgressIndicator::color of the component..

Definition at line 1 of file qprogressindicator.h.

13.15.4.2 delay

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

QProgressIndicator::delay in milliseconds.

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

bool QProgressIndicator::displayedWhenStopped [read]

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

13.16 QPushButtonAsQRCode Class Reference

Stylish widget to display the field as QR Code.

#include <qpushbuttonasqrcode.h>

Inheritance diagram for QPushButtonAsQRCode:

```
QPushButtonAsQRCode
|_QPushButton
```

Collaboration diagram for QPushButtonAsQRCode:

```
QPushButtonAsQRCode
|_QPushButton
```
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.

13.16.1 Detailed Description

Stylish widget to display the field as QR Code.

Definition at line 11 of file qpushbuttonasqrcode.h.

13.16.2 Constructor & Destructor Documentation

13.16.2.1 QPushButtonAsQRCode()

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

QPushButtonAsQRCode::QPushButtonAsQRCode basic constructor.

Parameters

| textToCopy | the text to display as qrcode |
| parent     | the parent window            |

Definition at line 12 of file qpushbuttonasqrcode.cpp.

13.16.3 Member Function Documentation

13.16.3.1 clicked

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

```
QPushButtonAsQRCode::clicked
```

13.16.3.2 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.

13.16.3.3 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
13.17 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.*

13.17.1 Detailed Description

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

13.17.2.1 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.

13.17.3 Member Function Documentation

13.17.3.1 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
13.18  QPushButtonWithClipboard Class Reference

Stylish widget to allow copying of password and account details.

#include <qpushbuttonwithclipboard.h>

Inheritance diagram for QPushButtonWithClipboard:

Collaboration diagram for 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.
13.18 QPushButtonWithClipboard Class Reference

13.18.1 Detailed Description

Stylish widget to allow copying of password and account details.

Definition at line 11 of file qpushbuttonwithclipboard.h.

13.18.2 Constructor & Destructor Documentation

13.18.2.1 QPushButtonWithClipboard()

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

QPushButtonWithClipboard::QPushButtonWithClipboard basic constructor.

Parameters

| textToCopy | the text to paste into the clipboard |
| parent | the parent window |

Definition at line 12 of file qpushbuttonwithclipboard.cpp.

13.18.3 Member Function Documentation

13.18.3.1 clicked

void QPushButtonWithClipboard::clicked ( QString ) [signal]

Here is the caller graph for this function:

![Caller Graph](image-url)
13.18.3.2 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.

13.18.3.3 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

13.19 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 &, const QString &p_output=QString())
- void clearClippedText ()
- void setClipboardTimer ()
- bool isFreshStart ()
- void setFreshStart (const bool &fs)

### 13.19.1 Detailed Description

Definition at line 10 of file qtpass.h.

### 13.19.2 Constructor & Destructor Documentation

Generated by Doxygen
13.19.2.1 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)

13.19.2.2 ~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)

13.19.3 Member Function Documentation
13.19.3.1 clearClipboard

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:

Here is the caller graph for this function:

13.19.3.2 clearClippedText()

void QtPass::clearClippedText ( )
Definition at line 364 of file qtpass.cpp.
Here is the caller graph for this function:

13.19.3.3 copyTextToClipboard

void QtPass::copyTextToClipboard ( const QString & text ) [slot]
MainWindow::copyTextToClipboard copies text to your clipboard.

Generated by Doxygen
Parameters

text

Definition at line 397 of file qtpass.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:

13.19.3.4 init()

bool QtPass::init ()

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

Definition at line 60 of file qtpass.cpp.
13.19.3.5 isFreshStart()

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

Definition at line 21 of file qtpass.h.
Here is the caller graph for this function:

![Caller Graph](image)

### 13.19.3.6 setClipboardTimer()

```cpp
void QtPass::setClipboardTimer() {
}
```

Definition at line 366 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)

### 13.19.3.7 setClippedText()

```cpp
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 Image]

Here is the caller graph for this function:

![Caller Graph Image]

### 13.19.3.8 setFreshStart()

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

Definition at line 22 of file qtpass.h.

Here is the caller graph for this function:

![Caller Graph Image]

### 13.19.3.9 showTextAsQRCode

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

displays the text as qrcode
Class Documentation

Parameters

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

Definition at line 416 of file qtpass.cpp.

Here is the call graph for this function:

![Call Graph]

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

- src/qtpass.h
- src/qtpass.cpp

13.20 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 &use Symbols)
• 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 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 ()

13.20.1 Detailed Description

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

Definition at line 21 of file qtpasssettings.h.

13.20.2 Member Function Documentation

13.20.2.1 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:

![Call Graph](image)

Here is the caller graph for this function:

![Caller Graph](image)
13.20.2.2 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 for getAutoclearSeconds()](image)

Here is the caller graph for this function:

![Caller Graph for getAutoclearSeconds()](image)

13.20.2.3 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 for getClipBoardType()](image)

Here is the caller graph for this function:

![Caller Graph for getClipBoardType()](image)
13.20.2.4 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()]

13.20.2.5 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()]
13.20.2.6  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}  
  |                   |
  | QtPass::init      |
  |                   |
  | QtPassSettings::initExecutables|

{ImitatePass::Init}  
  |                   |
  | QtPassSettings::getGitExecutable|

{ImitatePass::GitPull_b}  
  |                   |
  | QtPassSettings::getGitExecutable|

{ConfigDialog::ConfigDialog}  
  |                   |
  | QtPassSettings::getGitExecutable|

```

13.20.2.7  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
Here is the caller graph for this function:

```
QtPassSettings::getGpgHome QtPassSettings::getInstance
```

### 13.20.2.8 getGpgHome()

```cpp
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:
13.20.2.9  getImitatePass()

**ImitatePass** *QtPassSettings::getImitatePass ( ) [static]*

Definition at line 562 of file qtpasssettings.cpp.

Here is the caller graph for this function:

13.20.2.10  getInstance()

**QtPassSettings** *QtPassSettings::getInstance ( ) [static]*

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

13.20.2.11  getPass()

Pass  * QtPassSettings::getPass ( )  [static]
13.20.2.12 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::getPassExecutable QtPassSettings::getInstance

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

![Caller Graph](image)

### 13.20.2.13 `getPassStore()`

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

Definition at line 254 of file qtpasssettings.cpp.

Generated by Doxygen
13.20.2.14 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:

```
QtPassSettings::getPassTemplate  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow
QtPass::init
MainWindow::passShowHandler
PasswordDialog::Password
Dialog
```

13.20.2.15 getPasswordConfiguration()

PasswordConfiguration QtPassSettings::getPasswordConfiguration () [static]

Definition at line 35 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::getPasswordConfiguration  QtPassSettings::getInstance
```

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

```plaintext
QtPassSettings::getPassword

Configuration

ConfigDialog::ConfigDialog

PasswordDialog::Password
```

### 13.20.2.16 `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:

```plaintext
QtPassSettings::getPos

MainWindow::restoreWindow
```

Here is the caller graph for this function:

```plaintext
MainWindow::restoreWindow

QtPassSettings::getPos
```
### 13.20.2.17 getProfile()

```cpp
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:

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

Here is the caller graph for this function:

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

### 13.20.2.18 getProfiles()

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

Definition at line 61 of file qtpasssettings.cpp.

Here is the call graph for this function:

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

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

```
ConfigDialog::ConfigDialog -> QtPassSettings::getProfiles
```

### 13.20.2.19 `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:

```
QtPassSettings::getPwgenExecutable -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow -> QtPass::init
```

Generated by Doxygen
13.20.2.20  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:

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

Here is the caller graph for this function:

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

13.20.2.21  getRealPass()

RealPass * QtPassSettings::getRealPass () [static]

Definition at line 557 of file qtpasssettings.cpp.

Here is the caller graph for this function:

![Caller graph for getRealPass()](image)
13.20.2.22  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
```

13.20.2.23  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
```
Here is the caller graph for this function:

![Caller Graph](image)

### 13.20.2.24 `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:

![Call Graph](image)

Here is the caller graph for this function:

![Caller Graph](image)

### 13.20.2.25 `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
  QtPassSettings::getInstance

13.20.2.26  getWebDavUrl()

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

13.20.2.27  getWebDavUser()

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

Generated by Doxygen
### 13.20.2.28  initExecutables()

```c++
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_1.png)

Here is the caller graph for this function:

![Caller Graph for initExecutables()](caller_graph_1.png)

### 13.20.2.29  isAddGPGId()

```c++
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_2.png)
Here is the caller graph for this function:

![Caller Graph]

### 13.20.2.30 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:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

Generated by Doxygen
13.20.2.31 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
ImitatePass::reencryptPath  -->  QtPassSettings::isAutoPull
```

13.20.2.32 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
```
Here is the caller graph for this function:

![Callable Graph]

### 13.20.2.33 isAvoidCapitals()

```cpp
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:

![Callable Graph]

Here is the caller graph for this function:

![Callable Graph]
13.20.2.34 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()](image)

Here is the caller graph for this function:

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

13.20.2.35 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()](image)

Here is the caller graph for this function:

![Caller graph for isHideContent()](image)
Here is the caller graph for this function:

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

### 13.20.2.36 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:

```
QtPassSettings::isHideOnClose
QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::closeEvent
ConfigDialog::ConfigDialog
QtPassSettings::isHideOnClose
```
13.20.2.37  isHidePassword()

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:

Here is the caller graph for this function:

13.20.2.38  isLessRandom()

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:

Here is the caller graph for this function:
13.20.2.39 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:

```
QtPassSettings::isMaximized  QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::restoreWindow  QtPassSettings::isMaximized
```
13.20.2.40 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:

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

Here is the caller graph for this function:

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

13.20.2.41 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:

![Call Graph for isTemplateAllFields()](image3.png)
Here is the caller graph for this function:

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

### 13.20.2.42 isUseAutoclear()

```cpp
bool QtPassSettings::isUseAutoclear ( 
    const bool & defaultVal = 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:

```
QtPass::copyTextToClipboard
MainWindow::passOtpHandler
```

Generated by Doxygen
### 13.20.2.43 isUseAutoclearPanel()

```cpp
def 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:

![Call Graph for isUseAutoclearPanel()](call_graph_isUseAutoclearPanel.png)

Here is the caller graph for this function:

![Caller Graph for isUseAutoclearPanel()](caller_graph_isUseAutoclearPanel.png)

### 13.20.2.44 isUseGit()

```cpp
def 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:

![Call Graph for isUseGit()](call_graph_isUseGit.png)

Here is the caller graph for this function:

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

```
ConfigDialog::ConfigDialog
ImitatePass::Copy
ImitatePass::GitPush
ImitatePass::Init
ImitatePass::Move
MainWindow::onPush
ImitatePass::Remove
```

### 13.20.2.45 isUseOtp()

```cpp
definition at line 390 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::isUseOtp QtPassSettings::getInstance
```

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

```
ConfigDialog::ConfigDialog -> QtPassSettings::isUseOtp
```

### 13.20.2.46  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:

Generated by Doxygen
13.20.2.47 `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:

![Call Graph for `isUsePwgen()`](attachment:image.png)

Here is the caller graph for this function:

![Caller Graph for `isUsePwgen()`](attachment:image.png)

13.20.2.48 `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:

![Call Graph for `isUseQrencode()`](attachment:image.png)

Here is the caller graph for this function:

![Caller Graph for `isUseQrencode()`](attachment:image.png)
Here is the caller graph for this function:

```
ConfigDialog::ConfigDialog -> QtPassSettings::isUseQrencode
```

### 13.20.2.49 isUseSelection()

```cpp
definition at line 178 of file qtpasssettings.cpp.
```

Here is the call graph for this function:

```
QtPassSettings::isUseSelection -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::passOtpHandler -> QtPass::setClippedText
MainWindow::passShowHandler -> QtPass::copyTextToClipboard
```

### 13.20.2.50 isUseSymbols()

```cpp
definition at line 454 of file qtpasssettings.cpp.
```
Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

### 13.20.2.51 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:

![Call Graph]
Here is the caller graph for this function:

![Caller Graph](image)

### 13.20.2.52 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:

![Call Graph](image)

Here is the caller graph for this function:

![Caller Graph](image)
13.20.2.53  isUseWebDav()

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()](image)

Here is the caller graph for this function:

![Caller Graph for isUseWebDav()](image)

13.20.2.54  setAddGPGId()

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()](image)
13.20.2.55  setAlwaysOnTop()

```cpp
void QtPassSettings::setAlwaysOnTop ( const bool & alwaysOnTop ) [static]
```

Definition at line 507 of file qtpasssettings.cpp.

Here is the call graph for this function:

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

13.20.2.56  setAutoclearPanelSeconds()

```cpp
void QtPassSettings::setAutoclearPanelSeconds ( const int & autoClearPanelSeconds ) [static]
```

Definition at line 221 of file qtpasssettings.cpp.

Here is the call graph for this function:

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

Here is the caller graph for this function:

![Caller graph for setAutoclearPanelSeconds()](image)
13.20.2.57  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()]

13.20.2.58  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
### 13.20.2.59 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:

![Call Graph](image)

### 13.20.2.60 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:

![Call Graph](image)

### 13.20.2.61 setAvoidNumbers()

```cpp
void QtPassSettings::setAvoidNumbers (const bool & avoidNumbers) [static]
```

Definition at line 441 of file qtpasssettings.cpp.
Here is the call graph for this function:

```
QtPassSettings::setAvoidNumbers
    |               |
    |               |
    v               v
QtPassSettings::getInstance
```

### 13.20.2.62 setClipBoardType()

```cpp
template
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
    |               |
    |               |
    v               v
QtPassSettings::getInstance
```

### 13.20.2.63 setGeometry()

```cpp
template
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
    |               |
    |               |
    v               v
QtPassSettings::getInstance
```
Here is the caller graph for this function:

```
MainWindow::closeEvent -> QtPassSettings::setGeometry
```

### 13.20.2.64 `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::MainWindow -> QtPass::init
QtPass::init -> QtPassSettings::initExecutables
```

### 13.20.2.65 `setGpgExecutable()`

```cpp
void QtPassSettings::setGpgExecutable ( const QString & gpgExecutable ) [static]
```

Definition at line 319 of file qtpasssettings.cpp.
13.20 QtPassSettings Class Reference

Here is the call graph for this function:

```
QtPassSettings::setGpgExecutable -> QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
MainWindow::MainWindow -> QPass::init
QPassSettings::initExecutables -> QPassSettings::setGpgExecutable
ConfigDialog::~ConfigDialog
```

13.20.2.66 setHideContent()

```c
void QtPassSettings::setHideContent ( const bool & hideContent ) [static]
```

Definition at line 241 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setHideContent -> QtPassSettings::getInstance
```

13.20.2.67 setHideOnClose()

```c
void QtPassSettings::setHideOnClose ( const bool & hideOnClose ) [static]
```

Definition at line 489 of file qtpasssettings.cpp.
Here is the call graph for this function:

```
QtPassSettings::setHideOn
Close
QtPassSettings::getInstance
```

13.20.2.68 setHidePassword()

```c
void QtPassSettings::setHidePassword (  
    const bool & hidePassword ) [static]
```

Definition at line 232 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setHidePassword
QtPassSettings::getInstance
```

13.20.2.69 setLessRandom()

```c
void QtPassSettings::setLessRandom (  
    const bool & lessRandom ) [static]
```

Definition at line 450 of file qtpasssettings.cpp.

Here is the call graph for this function:

```
QtPassSettings::setLessRandom
QtPassSettings::getInstance
```
13.20.2.70 `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
```

13.20.2.71 `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:

```
MainWindow::MainWindow → QtPassSettings::setPassExecutable
```

Generated by Doxygen
### 13.20.2.72 setPassStore()

**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)

### 13.20.2.73 setPassTemplate()

**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)
### 13.20.2.74 setPasswordChars()

```cpp
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](image)

### 13.20.2.75 setPasswordCharsselection()

```cpp
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](image)

### 13.20.2.76 setPasswordConfiguration()

```cpp
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:

```
QtPassSettings::setPassword
    Configuration
                     
                        QtPassSettings::getInstance
```

### 13.20.2.77 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:

```
QtPassSettings::setPassword
Length
     
                    QtPassSettings::getInstance
```

### 13.20.2.78 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:

```
QtPassSettings::setPos
                     
                        QtPassSettings::getInstance
```
Here is the caller graph for this function:

```plaintext
MainWindow::closeEvent → QtPassSettings::setPos
```

**13.20.2.79 setProfile()**

```c++
void QtPassSettings::setProfile (const QString & profile) [static]
```
Definition at line 379 of file qtpasssettings.cpp.

Here is the call graph for this function:

```plaintext
QtPassSettings::setProfile → QtPassSettings::getInstance
```

**13.20.2.80 setProfiles()**

```c++
void QtPassSettings::setProfiles (const QHash<QString, QString> & profiles) [static]
```
Definition at line 75 of file qtpasssettings.cpp.

Here is the call graph for this function:

```plaintext
QtPassSettings::setProfiles → QtPassSettings::getInstance
```
13.20.2.81 setPwgenExecutable()


```cpp
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()](call_graph_pwgen.png)

Here is the caller graph for this function:

![Caller graph for setPwgenExecutable()](caller_graph_pwgen.png)

13.20.2.82 setQrencodeExecutable()


```cpp
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()](call_graph_qrencode.png)
13.20.2.83 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
```

13.20.2.84 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
```

### 13.20.2.85 setStartMinimized()

```c
void 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
```

### 13.20.2.86 setTemplateAllFields()

```c
void 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
```
### 13.20.2.87 `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()](call_graph1.png)

### 13.20.2.88 `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()](call_graph2.png)

### 13.20.2.89 `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()](call_graph3.png)
13.20.2.90 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:

13.20.2.91 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:

Here is the caller graph for this function:
13.20.2.92 setUsePwgen()

```cpp
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:

13.20.2.93 setUseQrencode()

```cpp
void QtPassSettings::setUseQrencode (const bool & useQrencode) [static]
```

Definition at line 404 of file qtpasssettings.cpp.

Here is the call graph for this function:
13.20.2.94  setUseSelection()

void QtPassSettings::setUseSelection (  
    const bool & useSelection ) [static]

Definition at line 183 of file qtpasssettings.cpp.
Here is the call graph for this function:

QtPassSettings::setUseSelection \rightarrow QtPassSettings::getInstance

13.20.2.95  setUseSymbols()

void QtPassSettings::setUseSymbols (  
    const bool & useSymbols ) [static]

Definition at line 459 of file qtpasssettings.cpp.
Here is the call graph for this function:

QtPassSettings::setUseSymbols \rightarrow QtPassSettings::getInstance

13.20.2.96  setUseTemplate()

void QtPassSettings::setUseTemplate (  
    const bool & useTemplate ) [static]

Definition at line 543 of file qtpasssettings.cpp.
Here is the call graph for this function:

QtPassSettings::setUseTemplate \rightarrow QtPassSettings::getInstance
### 13.20.2.97 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](image)

### 13.20.2.98 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](image)

### 13.20.2.99 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

13.20.2.100  setWebDavPassword()

```cpp
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

13.20.2.101  setWebDavUrl()

```cpp
void QtPassSettings::setWebDavUrl (  
    const QString & webDavUrl )  [static]
```

Definition at line 352 of file qtpasssettings.cpp.
Here is the call graph for this function:

```plaintext
QtPassSettings::setWebDavUrl -> QtPassSettings::getInstance
```

### 13.20.2.102 setWebDavUser()

```c
void QtPassSettings::setWebDavUser ( const QString & webDavUser ) [static]
```

Definition at line 361 of file qtpasssettings.cpp.

Here is the call graph for this function:

```plaintext
QtPassSettings::setWebDavUser -> QtPassSettings::getInstance
```

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

- src/qtpasssettings.h
- src/qtpasssettings.cpp

### 13.21 RealPass Class Reference

Wrapper for executing pass to handle the password-store.

```c
#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.
• 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

13.21.1 Detailed Description

Wrapper for executing pass to handle the password-store.

Definition at line 10 of file realpass.h.

13.21.2 Constructor & Destructor Documentation

13.21.2.1 RealPass()

RealPass::RealPass ( ) [default]

13.21.2.2 ~RealPass()

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

Definition at line 17 of file realpass.h.
13.21.3 Member Function Documentation

13.21.3.1 Copy()

```cpp
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 142 of file realpass.cpp.

Here is the call graph for this function:

![Call Graph](call_graph.png)

13.21.3.2 GitInit()

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

**RealPass::GitInit** pass git init wrapper.

Implements **Pass**.

Definition at line 16 of file realpass.cpp.
13.21.3.3 GitPull()

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

`RealPass::GitPull` pass git pull wrapper.

Implements `Pass`.

Definition at line 29 of file realpass.cpp.

13.21.3.4 GitPull_b()

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

`RealPass::GitInit` pass git pull wrapper which blocks until process finishes.

Implements `Pass`.

Definition at line 22 of file realpass.cpp.

Here is the call graph for this function:

```
RealPass::GitPull_b QtPassSettings::getPassExecutable QtPassSettings::getInstance
```

13.21.3.5 GitPush()

```cpp
void RealPass::GitPush ( ) [virtual]
```

`RealPass::GitPush` pass git push wrapper.

Implements `Pass`.

Definition at line 34 of file realpass.cpp.

13.21.3.6 Init()

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

`RealPass::Init` initialize pass repository.
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 81 of file realpass.cpp.

Here is the call graph for this function:

```
RealPass::Init QtPassSettings::getPassStore QtPassSettings::getInstance
```

### 13.21.3.7 Insert()

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

RealPass::Insert pass insert.

Implements Pass.

Definition at line 60 of file realpass.cpp.

### 13.21.3.8 Move()

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

RealPass::Move move 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 101 of file realpass.cpp.

Here is the call graph for this function:

```
RealPass::Move QtPassSettings::getPassStore QtPassSettings::getInstance
```

### 13.21.3.9 OtpGenerate()

```cpp
default
void RealPass::OtpGenerate (QString file) [virtual]
```

`RealPass::OtpGenerate` pass otp.

**Parameters**

- `file` file containig OTP uri

Implements `Pass`.

Definition at line 53 of file realpass.cpp.

### 13.21.3.10 Remove()

```cpp
default
void RealPass::Remove (QString file, bool isDir = false) [virtual]
```

`RealPass::Remove` pass remove wrapper.

Implements `Pass`.

Definition at line 71 of file realpass.cpp.

### 13.21.3.11 Show()

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

`RealPass::Show` pass show.
Parameters

| file | file to decrypt |

Returns

if block is set, returns exit status of internal decryption process otherwise returns QProcess::NormalExit

Implements Pass.

Definition at line 45 of file realpass.cpp.

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

- src/realpass.h
- src/realpass.cpp

## 13.22 SettingsConstants Class Reference

Table for the naming of configuration items.

```
#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 const QString gpgHome = "gpgHome"
- static const QString useWebDav = "useWebDav"
• 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"

13.22.1 Detailed Description

Table for the naming of configuration items.

Definition at line 10 of file settingsconstants.h.

13.22.2 Member Data Documentation

13.22.2.1 addGPGId

const QString SettingsConstants::addGPGId = "addGPGId"  [static]

Definition at line 29 of file settingsconstants.h.
13.22.2.2 alwaysOnTop

const QString SettingsConstants::alwaysOnTop = "alwaysOnTop" [static]

Definition at line 58 of file settingsconstants.h.

13.22.2.3 autoclearPanelSeconds

const QString SettingsConstants::autoclearPanelSeconds [static]

Initial value:
= "autoclearPanelSeconds"

Definition at line 26 of file settingsconstants.h.

13.22.2.4 autoclearSeconds

const QString SettingsConstants::autoclearSeconds = "autoclearSeconds" [static]

Definition at line 24 of file settingsconstants.h.

13.22.2.5 autoPull

const QString SettingsConstants::autoPull = "autoPull" [static]

Definition at line 59 of file settingsconstants.h.

13.22.2.6 autoPush

const QString SettingsConstants::autoPush = "autoPush" [static]

Definition at line 60 of file settingsconstants.h.

13.22.2.7 avoidCapitals

const QString SettingsConstants::avoidCapitals = "avoidCapitals" [static]

Definition at line 48 of file settingsconstants.h.
13.22.2.8 avoidNumbers

const QString SettingsConstants::avoidNumbers = "avoidNumbers" [static]

Definition at line 49 of file settingsconstants.h.

13.22.2.9 clipBoardType

const QString SettingsConstants::clipBoardType = "clipBoardType" [static]

Definition at line 64 of file settingsconstants.h.

13.22.2.10 geometry

const QString SettingsConstants::geometry [static]

Initial value:
= SettingsConstants::groupMainwindow + "/geometry"

Definition at line 14 of file settingsconstants.h.

13.22.2.11 gitExecutable

const QString SettingsConstants::gitExecutable = "gitExecutable" [static]

Definition at line 32 of file settingsconstants.h.

13.22.2.12 gpgExecutable

const QString SettingsConstants::gpgExecutable = "gpgExecutable" [static]

Definition at line 33 of file settingsconstants.h.

13.22.2.13 gpgHome

const QString SettingsConstants::gpgHome = "gpgHome" [static]

Definition at line 35 of file settingsconstants.h.
13.22.2.14  groupMainwindow

const QString SettingsConstants::groupMainwindow = "mainwindow" [static]

Definition at line 13 of file settingsconstants.h.

13.22.2.15  groupProfiles

const QString SettingsConstants::groupProfiles = "profiles" [static]

Definition at line 41 of file settingsconstants.h.

13.22.2.16  hideContent

const QString SettingsConstants::hideContent = "hideContent" [static]

Definition at line 28 of file settingsconstants.h.

13.22.2.17  hideOnClose

const QString SettingsConstants::hideOnClose = "hideOnClose" [static]

Definition at line 56 of file settingsconstants.h.

13.22.2.18  hidePassword

const QString SettingsConstants::hidePassword = "hidePassword" [static]

Definition at line 27 of file settingsconstants.h.

13.22.2.19  lessRandom

const QString SettingsConstants::lessRandom = "lessRandom" [static]

Definition at line 50 of file settingsconstants.h.
13.22.2.20  maximized

const QString SettingsConstants::maximized [static]

**Initial value:**
```
SettingsConstants::groupMainwindow + "/maximized"
```

Definition at line 20 of file settingsconstants.h.

13.22.2.21  passExecutable

const QString SettingsConstants::passExecutable = "passExecutable" [static]

Definition at line 31 of file settingsconstants.h.

13.22.2.22  passStore

const QString SettingsConstants::passStore = "passStore" [static]

Definition at line 30 of file settingsconstants.h.

13.22.2.23  passTemplate

const QString SettingsConstants::passTemplate = "passTemplate" [static]

Definition at line 61 of file settingsconstants.h.

13.22.2.24  passwordChars

const QString SettingsConstants::passwordChars = "passwordChars" [static]

Definition at line 54 of file settingsconstants.h.

13.22.2.25  passwordCharsselection

const QString SettingsConstants::passwordCharsselection [static]

**Initial value:**
```
"passwordCharsselection"
```

Definition at line 53 of file settingsconstants.h.
13.22.2.26  passwordLength

```cpp
const QString SettingsConstants::passwordLength = "passwordLength" [static]
```

Definition at line 52 of file settingsconstants.h.

13.22.2.27  pos

```cpp
const QString SettingsConstants::pos [static]
```

**Initial value:**
```cpp
    SettingsConstants::groupMainwindow + "/pos"
```

Definition at line 16 of file settingsconstants.h.

13.22.2.28  profile

```cpp
const QString SettingsConstants::profile = "profile" [static]
```

Definition at line 40 of file settingsconstants.h.

13.22.2.29  pwgenExecutable

```cpp
const QString SettingsConstants::pwgenExecutable = "pwgenExecutable" [static]
```

Definition at line 34 of file settingsconstants.h.

13.22.2.30  qrencodeExecutable

```cpp
const QString SettingsConstants::qrencodeExecutable = "qrencodeExecutable" [static]
```

Definition at line 45 of file settingsconstants.h.

13.22.2.31  savestate

```cpp
const QString SettingsConstants::savestate [static]
```

**Initial value:**
```cpp
    SettingsConstants::groupMainwindow + "/savestate"
```

Definition at line 15 of file settingsconstants.h.
13.22.2.32  size

const QString SettingsConstants::size [static]

Initial value:
= SettingsConstants::groupMainwindow + "/size"

Definition at line 17 of file settingsconstants.h.

13.22.2.33  splitterLeft

const QString SettingsConstants::splitterLeft [static]

Initial value:
= SettingsConstants::groupMainwindow + "/splitterLeft"

Definition at line 18 of file settingsconstants.h.

13.22.2.34  splitterRight

const QString SettingsConstants::splitterRight [static]

Initial value:
= SettingsConstants::groupMainwindow + "/splitterRight"

Definition at line 19 of file settingsconstants.h.

13.22.2.35  startMinimized

const QString SettingsConstants::startMinimized = "startMinimized" [static]

Definition at line 57 of file settingsconstants.h.

13.22.2.36  templateAllFields

const QString SettingsConstants::templateAllFields = "templateAllFields" [static]

Definition at line 63 of file settingsconstants.h.
13.22.2.37  useAutoclear

const QString SettingsConstants::useAutoclear = "useAutoclear" [static]

Definition at line 22 of file settingsconstants.h.

13.22.2.38  useAutoclearPanel

const QString SettingsConstants::useAutoclearPanel = "useAutoclearPanel" [static]

Definition at line 25 of file settingsconstants.h.

13.22.2.39  useClipboard

const QString SettingsConstants::useClipboard = "useClipboard" [static]

Definition at line 46 of file settingsconstants.h.

13.22.2.40  useGit

const QString SettingsConstants::useGit = "useGit" [static]

Definition at line 42 of file settingsconstants.h.

13.22.2.41  useOtp

const QString SettingsConstants::useOtp = "useOtp" [static]

Definition at line 43 of file settingsconstants.h.

13.22.2.42  usePass

const QString SettingsConstants::usePass = "usePass" [static]

Definition at line 21 of file settingsconstants.h.
13.22.2.43 usePwgen

```cpp
const QString SettingsConstants::usePwgen = "usePwgen" [static]
```
Definition at line 47 of file settingsconstants.h.

13.22.2.44 useQrencode

```cpp
const QString SettingsConstants::useQrencode = "useQrencode" [static]
```
Definition at line 44 of file settingsconstants.h.

13.22.2.45 useSelection

```cpp
const QString SettingsConstants::useSelection = "useSelection" [static]
```
Definition at line 23 of file settingsconstants.h.

13.22.2.46 useSymbols

```cpp
const QString SettingsConstants::useSymbols = "useSymbols" [static]
```
Definition at line 51 of file settingsconstants.h.

13.22.2.47 useTemplate

```cpp
const QString SettingsConstants::useTemplate = "useTemplate" [static]
```
Definition at line 62 of file settingsconstants.h.

13.22.2.48 useTrayIcon

```cpp
const QString SettingsConstants::useTrayIcon = "useTrayIcon" [static]
```
Definition at line 55 of file settingsconstants.h.
13.22.2.49 useWebDav

```
const QString SettingsConstants::useWebDav = "useWebDav"  [static]
```

Definition at line 36 of file settingsconstants.h.

13.22.2.50 version

```
const QString SettingsConstants::version = "version"  [static]
```

Definition at line 12 of file settingsconstants.h.

13.22.2.51 webDavPassword

```
const QString SettingsConstants::webDavPassword = "webDavPassword"  [static]
```

Definition at line 39 of file settingsconstants.h.

13.22.2.52 webDavUrl

```
const QString SettingsConstants::webDavUrl = "webDavUrl"  [static]
```

Definition at line 37 of file settingsconstants.h.

13.22.2.53 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
13.23 simpleTransaction Class Reference

#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

13.23.1 Detailed Description

Definition at line 7 of file simpletransaction.h.

13.23.2 Constructor & Destructor Documentation

13.23.2.1 simpleTransaction()

```
simpleTransaction::simpleTransaction ( ) [inline]
```

Definition at line 13 of file simpletransaction.h.
13.23.3 Member Function Documentation

13.23.3.1 transactionAdd()

```cpp
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).

**simpleTransaction::transactionAdd**

**Parameters**

<table>
<thead>
<tr>
<th>id</th>
<th>process that shall be treated as part of transaction</th>
</tr>
</thead>
</table>

Definition at line 25 of file simpletransaction.cpp.

13.23.3.2 transactionEnd()

```cpp
void simpleTransaction::transactionEnd ( Enums::PROCESS pid )
```

transactionEnd marks end of transaction

**simpleTransaction::transactionEnd**

**Parameters**

<table>
<thead>
<tr>
<th>pid</th>
<th>value that will be used as a result of transaction</th>
</tr>
</thead>
</table>

Definition at line 40 of file simpletransaction.cpp.

13.23.3.3 transactionIsOver()

```cpp
PROCESS simpleTransaction::transactionIsOver ( Enums::PROCESS id )
```

transactionIsOver checks whether currently finished process is last in current transaction

**simpleTransaction::transactionIsOver**
Returns
result of transaction as set by transactionAdd or transactionEnd if the transaction is over or PROCESS::IN←VALID if it's not yet over

Parameters

id

Returns

Definition at line 58 of file simpletransaction.cpp.

13.23.3.4 transactionStart()

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

13.24 SingleApplication Class Reference

The SingleApplication class is used for commandline integration.

#include <singleapplication.h>

Inheritance diagram for SingleApplication:

[Diagram of inheritance: SingleApplication inherits from QApplication]
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 whether to be server or client.

- bool **sendMessage** (const QString &message)

  `SingleApplication::sendMessage` send a message (from commandline) to an already running QtPass instance.

### 13.24.1 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.

### 13.24.2 Constructor & Destructor Documentation

#### 13.24.2.1 SingleApplication()

`SingleApplication::SingleApplication` this replaces the QApplication allowing for local socket based communications.
### Parameters

- `argc`
- `argv`
- `uniqueKey`

Definition at line 15 of file `singleapplication.cpp`.

Here is the call graph for this function:

![Call Graph](image)

#### 13.24.3 Member Function Documentation

##### 13.24.3.1 isRunning()

```
bool SingleApplication::isRunning ()
```

`SingleApplication::isRunning` is there already a QtPass instance running, to check wether to be server or client.

Returns

Definition at line 64 of file `singleapplication.cpp`.

Here is the caller graph for this function:

![Caller Graph](image)

##### 13.24.3.2 messageAvailable

```
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:

---

### 13.24.3.3 receiveMessage

```cpp
define SingleApplication::receiveMessage () [slot]

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:

---

### 13.24.3.4 sendMessage()

```cpp
define bool SingleApplication::sendMessage (const QString & message)

SingleApplication::sendMessage send a message (from command line) to an already running QtPass instance.
```
Parameters

message

Returns

Definition at line 72 of file singleapplication.cpp.

Here is the caller graph for this function:

```
main -> SingleApplication::sendMessage
```

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

- src/singleapplication.h
- src/singleapplication.cpp

### 13.25 StoreModel Class Reference

The QSortFilterProxyModel for handling filesystem searches.

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

Inheritance diagram for StoreModel:

```
QSortFilterProxyModel
    StoreModel
```

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.`

### 13.25.1 Detailed Description

The QSortFilterProxyModel for handling filesystem searches.

Definition at line 12 of file storemodel.h.
13.25 StoreModel Class Reference

13.25.2 Constructor & Destructor Documentation

13.25.2.1 StoreModel()

StoreModel::StoreModel()

StoreModel::StoreModel SubClass of QSortFilterProxyModel via http://www.qtcentre.org/threads/46471-
QTreeView-Filter.

Definition at line 32 of file storemodel.cpp.

13.25.3 Member Function Documentation

13.25.3.1 canDropMimeData()

bool StoreModel::canDropMimeData ( const QMimeData * data, Qt::DropAction action, int row, int column,
const QModelIndex & parent ) const [override]

StoreModel::canDropMimeData.

Parameters

| data   |
| action |
| row    |
| column |
| parent |
Returns

Definition at line 187 of file storemodel.cpp.

Here is the call graph for this function:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

### 13.25.3.2 data()

```cpp
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>
<th>role</th>
</tr>
</thead>
</table>

**Returns**

Definition at line 95 of file storemodel.cpp.
13.25.3.3  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>QModelIndex &amp;</td>
</tr>
</tbody>
</table>

**Returns**

Definition at line 236 of file storemodel.cpp.

Here is the call graph for this function:
13.25.3.4 filterAcceptsRow()

```cpp
bool StoreModel::filterAcceptsRow (  
    int sourceRow,  
    const QModelIndex & sourceParent ) const [override]
```

`StoreModel::filterAcceptsRow` should row be shown, wrapper for `StoreModel::ShowThis` method.

Parameters

- `sourceRow`
- `sourceParent`

Returns

Definition at line 41 of file storemodel.cpp.

Here is the call graph for this function:

![Call Graph](image)

13.25.3.5 flags()

```cpp
Qt::ItemFlags StoreModel::flags (  
    const QModelIndex & index ) const [override]
```

`StoreModel::flags`.

Parameters

- `index`

Returns

Definition at line 132 of file storemodel.cpp.
13.25.3.6 lessThan()

```cpp
bool StoreModel::lessThan(
    const QModelIndex & source_left,
    const QModelIndex & source_right) const [override]
```

**Parameters**

- `source_left`
- `source_right`

**Returns**

Definition at line 300 of file storemodel.cpp.

13.25.3.7 mimeData()

```cpp
QMimeData * StoreModel::mimeData ( 
    const QModelIndexList & indexes ) const [override]
```

**Parameters**

- `indexes`

**Returns**

Definition at line 156 of file storemodel.cpp.

13.25.3.8 mimeTypes()

```cpp
QStringList StoreModel::mimeTypes ( ) const [override]
```

**Returns**

Definition at line 145 of file storemodel.cpp.
13.25.3.9  setModelAndStore()

```cpp
void StoreModel::setModelAndStore ( 
    QFileSystemModel * sourceModel, 
    QString passStore )
```

`StoreModel::setModelAndStore` update the source model and store.

**Parameters**

- `sourceModel`
- `passStore`

Definition at line 82 of file storemodel.cpp.

Here is the caller graph for this function:

![Caller Graph](diagram)

13.25.3.10  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 53 of file storemodel.cpp.
Here is the caller graph for this function:

```
StoreModel::filterAcceptsRow --> StoreModel::ShowThis
```

### 13.25.3.11 supportedDragActions()

```cpp
Qt::DropActions StoreModel::supportedDragActions () const [override]
```

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

- src/storemodel.h
- src/storemodel.cpp

The documentation for this class was generated from the following files:
13.26  TrayIcon Class Reference

Handles the systemtray icon and menu.

#include <trayicon.h>

Inheritance diagram for TrayIcon:

```
  QWidget
  
  TrayIcon

Collaboration diagram for 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 systray message for notification.
- void setVisible (bool visible)
  TrayIcon::setVisible show or hide the icon.
- bool getIsAllocated ()
  TrayIcon::getIsAllocated return if TrayIcon is allocated.
13.26.1 Detailed Description

Handles the systemtray icon and menu.

Definition at line 14 of file trayicon.h.

13.26.2 Constructor & Destructor Documentation

13.26.2.1 TrayIcon()

TrayIcon::TrayIcon ( QMainWindow * parent ) [explicit]

TrayIcon::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:

13.26.3 Member Function Documentation

13.26.3.1 getIsAllocated()

bool TrayIcon::getIsAllocated ( )

TrayIcon::getIsAllocated return if TrayIcon is allocated.

Definition at line 57 of file trayicon.cpp.

Generated by Doxygen
13.26.3.2 iconActivated

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:

Here is the caller graph for this function:

13.26.3.3 setVisible()

void TrayIcon::setVisible (  
   bool visible )

TrayIcon::setVisible show or hide the icon.

Parameters
visible
13.26.3.4  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 --------> TrayIcon::iconActivated --------> TrayIcon::showHideParent
```

13.26.3.5  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 --------> TrayIcon::iconActivated --------> TrayIcon::showMessage
```

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

Generated by Doxygen
13.27 tst_ui Class Reference

The `tst_ui` class is our first unit test.

Inheritance diagram for tst_ui:

![Inheritance Diagram](image)

Collaboration diagram for tst_ui:

![Collaboration Diagram](image)

13.27.1 Detailed Description

The `tst_ui` class is our first unit test.

Definition at line 9 of file `tst_ui.cpp`.

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

- tests/auto/ui/tst_ui.cpp
The `tst_util` class is our first unit test.

Inheritance diagram for `tst_util`:

```
QObject
    tst_util
```

Collaboration diagram for `tst_util`:

```
QObject
    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
13.28.1 Detailed Description

The `tst_util` class is our first unit test.

Definition at line 10 of file `tst_util.cpp`.

13.28.2 Constructor & Destructor Documentation

13.28.2.1 `tst_util()`

`tst_util::tst_util ( ) [default]`

`tst_util::tst_util` basic constructor

13.28.2.2 `~tst_util()`

`tst_util::~tst_util ( ) [override], [default]`

`tst_util::~tst_util` basic destructor

13.28.3 Member Function Documentation

13.28.3.1 `cleanup`

`void tst_util::cleanup ( ) [slot]`

`tst_util::cleanup` unit test cleanup method

Definition at line 50 of file `tst_util.cpp`.

13.28.3.2 `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`
13.29 UserInfo Struct Reference

Stores key info lines including validity, creation date and more.

#include <userinfo.h>

Public Member Functions

- UserInfo()
- bool fullyValid()
  
  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()
  
  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()
  
  UserInfo::isValid when fullyValid or marginallyValid.

Public Attributes

- QString name
  
  UserInfo::name full name.

- QString key_id
  
  UserInfo::key_id hexadecimal representation.

- char validity
  
  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
  
  UserInfo::have_secret secret key is available (can decrypt with this key)

- bool enabled
  
  UserInfo::enabled.

- QDateTime expiry
  
  UserInfo::expiry date/time key expires.

- QDateTime created
  
  UserInfo::created date/time key was created.

13.29.1 Detailed Description

Stores key info lines including validity, creation date and more.

Definition at line 11 of file userinfo.h.

13.29.2 Constructor & Destructor Documentation
13.29.2.1 UserInfo()

UserInfo::UserInfo ( ) [inline]
Definition at line 12 of file userinfo.h.

13.29.3 Member Function Documentation

13.29.3.1 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:

```
UserInfo::isValid → UserInfo::fullyValid
```

13.29.3.2 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:

```
UserInfo::isValid → UserInfo::fullyValid
UserInfo::fullyValid → UserInfo::marginallyValid
```

Generated by Doxygen
13.29.3.3 marginallyValid()

bool UserInfo::marginallyValid ( )  [inline]

UserInfo::marginallyValid when validity is m. 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
```

13.29.4 Member Data Documentation

13.29.4.1 created

QDateTime UserInfo::created

UserInfo::created date/time key was created.

Definition at line 58 of file userinfo.h.

13.29.4.2 enabled

bool UserInfo::enabled

UserInfo::enabled.

Definition at line 50 of file userinfo.h.

13.29.4.3 expiry

QDateTime UserInfo::expiry

UserInfo::expiry date/time key expires.

Definition at line 54 of file userinfo.h.
13.29.4.4  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.

13.29.4.5  key_id

QString UserInfo::key_id

UserInfo::key_id hexadecimal representation.

Definition at line 36 of file userinfo.h.

13.29.4.6  name

QString UserInfo::name

UserInfo::name full name.

Definition at line 32 of file userinfo.h.

13.29.4.7  validity

char UserInfo::validity

UserInfo::validity GnuPG representation of validity http://git.gnupg.org/cgi-bin/gitweb.cgi?p=gnupg.git;a=blob_plain;f=doc/DETAILS.

Definition at line 41 of file userinfo.h.

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

- src/userinfo.h
13.30 UsersDialog Class Reference

Handles listing and editing of GPG users.

#include <usersdialog.h>

Inheritance diagram for UsersDialog:

```
#include <usersdialog.h>

Inheritance 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.

13.30.1 Detailed Description

Handles listing and editing of GPG users.

Selection of whom to encrypt to.

Definition at line 23 of file usersdialog.h.

13.30.2 Constructor & Destructor Documentation

13.30.2.1 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:
13.30.2.2 ~UsersDialog()

UsersDialog::~UsersDialog ( )

UsersDialog::~UsersDialog basic destructor.

Definition at line 82 of file usersdialog.cpp.

13.30.3 Member Function Documentation

13.30.3.1 accept

void UsersDialog::accept ( ) [slot]

UsersDialog::accept.

Definition at line 89 of file usersdialog.cpp.

Here is the call graph for this function:

```
UsersDialog::accept
QtPassSettings::getPass
Pass::Init
QtPassSettings::getImitatePass
QtPassSettings::getRealPass
Pass::init
QtPassSettings::isUsePass
QtPassSettings::getGpgHome
QtPassSettings::getInstance
```

Here is the caller graph for this function:

```
UsersDialog::accept
UsersDialog::UsersDialog
```

13.30.3.2 closeEvent()

void UsersDialog::closeEvent ( )

UsersDialog::closeEvent might have to store size and location if that is wanted.
13.30.3.3   keyPressEvent()

```cpp
void UsersDialog::keyPressEvent (
    QKeyEvent * event ) [protected]
```

UsersDialog::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`

13.31   Util Class Reference

Some static utilities to be used elsewhere.

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

**Static Public Member Functions**

- static QString findBinaryInPath (QString binary)
  
  `Util::findBinaryInPath` search for executables.

- static QString findPasswordStore ()
  
  `Util::findPasswordStore` look for common .password-store folder location.

- static QString normalizeFolderPath (QString path)
  
  `Util::normalizeFolderPath` let's always end folders with a QDir::separator()

- static bool checkConfig ()
  
  `Util::checkConfig` do we have requisite settings?

- static QString getDir (const QModelIndex &index, bool forPass, const QFileSystemModel &model, const StoreModel &storeModel)
  
  `Util::getDir` get selected folder path.

- static void copyDir (const QString &src, const QString &dest)
  
  `Util::copyDir`.

Generated by Doxygen
13.31 Util Class Reference

13.31.1 Detailed Description

Some static utilities to be used elsewhere.

Definition at line 15 of file util.h.

13.31.2 Member Function Documentation

13.31.2.1 checkConfig()

bool Util::checkConfig () [static]

Util::checkConfig do we have prequisite 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:
13.31.2.2 copyDir()

void Util::copyDir (  
    const QString & src,  
    const QString & dest ) [static]

Util::copyDir.
Parameters

| src | dest |

Definition at line 182 of file util.cpp.

### 13.31.2.3 findBinaryInPath()

```cpp
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:

```
MainWinow:MainWindow -> QtPass::init -> QtPass::settings::initExecutables -> Util::findBinaryInPath
```
13.31.2.4 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:

![Call Graph]

Here is the caller graph for this function:

![Caller Graph]

13.31.2.5 getDir()

QString Util::getDir( 
    const QModelIndex & index,
    bool forPass,
    const QFileSystemModel & model,
    const StoreModel & storeModel ) [static]

Util::getDir get selectd folder path.

Parameters

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>index</strong></td>
<td>short or full path</td>
</tr>
<tr>
<td><strong>forPass</strong></td>
<td>the filesystem model to operate on</td>
</tr>
<tr>
<td><strong>model</strong></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

13.31.2.6 normalizeFolderPath()

QString Util::normalizeFolderPath (QString path) [static]

Util::normalizeFolderPath lets always end folders with a QDir::separator()

Parameters

path

Returns

Definition at line 79 of file util.cpp.

Here is the call graph for this function:

MainWindow::MainWindow → QtPass::init → Util::findPasswordStore → Util::normalizeFolderPath

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

- src/util.h
- src/util.cpp
Chapter 14

File Documentation

14.1 CHANGELOG.md File Reference

14.2 CODE_OF_CONDUCT.md File Reference

14.3 CONTRIBUTING.md File Reference

14.4 FAQ.md File Reference

14.5 main/main.cpp File Reference

```cpp
#include "mainwindow.h"
#include <QApplication>
#include <QDir>
#include <QTranslator>
#include <QtWidgets>
```

Include dependency graph for main.cpp:

Functions

- int main (int argc, char *argv[])
14.5.1 Function Documentation

14.5.1.1 main()

```c
int main ( 
    int argc,
    char * argv[] )
```

main

Parameters

<table>
<thead>
<tr>
<th>argc</th>
<th>argv</th>
</tr>
</thead>
</table>

Returns

Definition at line 43 of file main.cpp.

Here is the call graph for this function:

[Call graph diagram]

14.6 README.md File Reference

14.7 src/configdialog.cpp File Reference

```c
#include "configdialog.h"
#include "keygendialog.h"
```
```cpp
#include "mainwindow.h"
#include "qtpasssettings.h"
#include "ui_configdialog.h"
#include <QClipboard>
#include <QDir>
#include <QFileDialog>
#include <QMessageBox>
#include <QPushButton>
#include <QSystemTrayIcon>
#include <QTableWidgetItem>
#include "enums.h"
#include "passwordconfiguration.h"
#include <QDialog>
```

**14.8 src/configdialog.h File Reference**

```cpp
#include "enums.h"
#include "passwordconfiguration.h"
#include <QDialog>
```

Include dependency graph for configdialog.h:
This graph shows which files directly or indirectly include this file:

Classes

- class ConfigDialog
  
  The ConfigDialog handles the configuration interface.

Namespaces

- Ui

14.9 src/debughelper.h File Reference

#include <QDebug>

Include dependency graph for debughelper.h:

Macros

- #define dbg() qDebug() << __FILE__ ":" << __LINE__

14.9.1 Macro Definition Documentation
14.9.1.1 dbg

#define dbg( ) qDebug( ) << __FILE__ " ": " << __LINE__

Definition at line 7 of file debughelper.h.

14.10 src/deselectabletreeview.h File Reference

#include <QCoreApplication>
#include <QMouseEvent>
#include <QTime>
#include <QTreeView>

Include dependency graph for deselectabletreeview.h:

```
class DeselectableTreeView

The DeselectableTreeView class loosely based on
http://stackoverflow.com/questions/2761284/
thanks to Yassir Ennazk.
```

14.11 src/enums.h File Reference

This graph shows which files directly or indirectly include this file:

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 }

14.12 src/executor.cpp File Reference

#include "executor.h"
#include <QCoreApplication>
#include <QDir>

Include dependency graph for executor.cpp:

14.13 src/executor.h File Reference

#include <QObject>
#include <QProcess>
#include <QQueue>
Include dependency graph for executor.h:

This graph shows which files directly or indirectly include this file:

Classes

- class Executor
  
  *Executes external commands for handling password, git and other data.*

14.14 src/filecontent.cpp File Reference

#include "filecontent.h"

Include dependency graph for filecontent.cpp:
14.15  src/filecontent.h File Reference

#include <QList>
#include <QString>
#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

14.16  src/imitatepass.cpp File Reference

#include "imitatepass.h"
#include "qtpasssettings.h"
#include <QDirIterator>
#include <QRegularExpression>
#include <utility>

Include dependency graph for imitatepass.cpp:

This graph shows which files directly or indirectly include this file:

Classes

- class ImitatePass

Imitates pass features when pass is not enabled or available.
14.18 src/keygendialog.cpp File Reference

#include "keygendialog.h"
#include "configdialog.h"
#include "qprogressindicator.h"
#include "ui_keygendialog.h"
#include <QMessageBox>
#include <QRegularExpression>

Include dependency graph for keygendialog.cpp:

14.19 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

14.20  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:

14.21  src/mainwindow.h File Reference

#include "storemodel.h"
#include <QFileSystemModel>
#include <QItemSelectionModel>
#include <QMainSelectionModel>
#include <QProcess>

Generated by Doxygen
#include <QTimer>

Include dependency graph for mainwindow.h:

![Dependency Graph](image)

This graph shows which files directly or indirectly include this file:

![File Inclusion 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

14.21.1 Macro Definition Documentation

14.21.1.1 SingleApplication

#define SingleApplication QApplication

Definition at line 15 of filemainwindow.h.
# include "pass.h"
# include "qtpasssettings.h"
# include "util.h"

Include dependency graph for pass.cpp:

This graph shows which files directly or indirectly include this file:

---

# 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.

14.24 src/passwordconfiguration.h File Reference

#include <QString>

Include dependency graph for passwordconfiguration.h:

```
src/passwordconfiguration.h
QString
```

This graph shows which files directly or indirectly include this file:

Classes

- struct **PasswordConfiguration**
  
  Holds the Password configuration settings.

14.25 src/passworddialog.cpp File Reference

#include "passworddialog.h"
#include "filecontent.h"
#include "pass.h"
#include "passwordconfiguration.h"
#include "qtpasssettings.h"
#include "ui_passworddialog.h"
#include <QLabel>
14.26 src/passworddialog.h File Reference

#include "passwordconfiguration.h"
#include <QDialog>

14.26 src/passworddialog.h File Reference

#include <QLineEdit>

Include dependency graph for passworddialog.cpp:

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

14.27  src/qprogressindicator.cpp File Reference

```cpp
#include "qprogressindicator.h"
#include <QPainter>
```

Include dependency graph for qprogressindicator.cpp:

14.28  src/qprogressindicator.h File Reference

```cpp
#include <QColor>
#include <QWidget>
```

Include dependency graph for qprogressindicator.h:
This graph shows which files directly or indirectly include this file:

Classes

- class QProgressIndicator

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

14.29 src/qpushbuttonasqrcode.cpp File Reference

#include "qpushbuttonasqrcode.h"
#include <QTimer>

Include dependency graph for qpushbuttonasqrcode.cpp:
14.30  src/qpushbuttonasqrcode.h File Reference

```c
#include <QPushButton>
```

Include dependency graph for qpushbuttonasqrcode.h:

This graph shows which files directly or indirectly include this file:

```
src/qpushbuttonasqrcode.h
src/mainwindow.cpp
src/qpushbuttonasqrcode.cpp
```

Classes

- class QPushButtonAsQRCode
  
  Stylish widget to display the field as QR Code.

14.31  src/qpushbuttonshowpassword.cpp File Reference

```c
#include "qpushbuttonshowpassword.h"
#include <QTimer>
```
Include dependency graph for qpushbuttonshowpassword.cpp:

```
#include <QLineEdit>
#include <QPushButton>
```

Include dependency graph for qpushbuttonshowpassword.h:

```
#include <QLineEdit>
#include <QPushButton>
```
This graph shows which files directly or indirectly include this file:

```
src/qpushbuttonshowpassword.h
```

```
src/mainwindow.cpp
src/qpushbuttonshowpassword.cpp
```

### Classes

- class `QPushbuttonShowPassword`

### 14.33 src/qpushbuttonwithclipboard.cpp File Reference

```c
#include "qpushbuttonwithclipboard.h"
#include <QTimer>
```

Include dependency graph for `qpushbuttonwithclipboard.cpp`:

```
src/qpushbuttonwithclipboard.cpp
```

```
qpushbuttonwithclipboard.h
QTimer
```

```
QPushButton
```

### 14.34 src/qpushbuttonwithclipboard.h File Reference

```c
#include <QPushButton>
```
Include dependency graph for qpushbuttonwithclipboard.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.**

14.35  src/qtpass.cpp File Reference

```cpp
#include "qtpass.h"
#include "mainwindow.h"
#include "qtpasssettings.h"
#include <QApplication>
#include <QClipboard>
#include <QDialog>
#include <QLabel>
#include <QPixmap>
#include <QPushButton>
#include <QVBoxLayout>
#include <QInputDialog>
#include <QLineEdit>
```

Generated by Doxygen
#include <utility>

Include dependency graph for qtpass.cpp:

14.36 src/qtpass.h File Reference

#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
Classes

- class QtPassSettings

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

14.39 src/realpass.cpp File Reference

```cpp
#include "realpass.h"
#include "qtpasssettings.h"
#include <QDir>
#include <QFileInfo>
#include <QRegularExpression>
#include <utility>
```

Include dependency graph for realpass.cpp:

14.40 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.

14.41 src/settingsconstants.cpp File Reference

#include "settingsconstants.h"

Include dependency graph for settingsconstants.cpp:

14.42 src/settingsconstants.h File Reference

#include <QString>

Include dependency graph for settingsconstants.h:
This graph shows which files directly or indirectly include this file:

```
src/settingsconstants.h
src/mainwindow.cpp
src/qtpasssettings.h
src/settingsconstants.cpp
src/configdialog.cpp
src/imitatepass.cpp
src/pass.cpp
src/passworddialog.cpp
src/qtpass.cpp
src/qtpasssettings.cpp
src/realpass.cpp
src/storemodel.cpp
src/usersdialog.cpp
src/util.cpp
```

Classes

- class **SettingsConstants**
  
  Table for the naming of configuration items.

14.43 src/simpletransaction.cpp File Reference

```
#include "simpletransaction.h"
#include <utility>
```

Include dependency graph for simpletransaction.cpp:

```
src/simpletransaction.cpp

simpletransaction.h

utility

enums.h

queue
```

14.44 src/simpletransaction.h File Reference

```
#include "enums.h"
#include <queue>
```
Include dependency graph for simpletransaction.h:

```
src/simpletransaction.h

enums.h  queue
```

This graph shows which files directly or indirectly include this file:

```
singleapplication.h QLocalSocket utility
QApplication QLocalServer QSharedMemory
```

### Classes

- `class simpleTransaction`

### src/singleapplication.cpp File Reference

```
#include "singleapplication.h"
#include <QLocalSocket>
#include <utility>
```

Include dependency graph for singleapplication.cpp:

```
singleapplication.cpp

singleapplication.h  QLocalSocket  utility

QApplication  QLocalServer  QSharedMemory
```

Generated by Doxygen
14.46 src/singleapplication.h File Reference

```cpp
#include <QApplication>
#include <QLocalServer>
#include <QSharedMemory>
```

Include dependency graph for singleapplication.h:

```
src/singleapplication.h

QApplication  QLocalServer  QSharedMemory
```

This graph shows which files directly or indirectly include this file:

```
src/singleapplication.h

src/singleapplication.cpp
```

Classes

- class SingleApplication
  
  The SingleApplication class is used for commandline integration.

14.47 src/storemodel.cpp File Reference

```cpp
#include "storemodel.h"
#include "qtpasssettings.h"
#include <QDebug>
#include <QRegularExpression>
#include <QMessageBox>
#include <QMimeData>
```
#include <utility>

Include dependency graph for storemodel.cpp:

![Dependency Graph](image)

**Functions**

- QDataStream & operator<< (QDataStream & out, const dragAndDropInfoPasswordStore & dragAndDropInfoPasswordStore)
- QDataStream & operator>>(QDataStream & in, dragAndDropInfoPasswordStore & dragAndDropInfoPasswordStore)

**14.47.1 Function Documentation**

**14.47.1.1 operator<<()**

QDataStream& operator<<(QDataStream & out, const dragAndDropInfoPasswordStore & dragAndDropInfoPasswordStore)

Definition at line 10 of file storemodel.cpp.

**14.47.1.2 operator>>()**

QDataStream& operator>>(QDataStream & in, dragAndDropInfoPasswordStore & dragAndDropInfoPasswordStore)

Definition at line 20 of file storemodel.cpp.
14.48 src/storemodel.h File Reference

```c
#include "util.h"
#include <QSortFilterProxyModel>
```

Include dependency graph for storemodel.h:

This graph shows which files directly or indirectly include this file:

- class StoreModel
  The QSortFilterProxyModel for handling filesystem searches.
- struct dragAndDropInfoPasswordStore

14.49 src/trayicon.cpp File Reference

```c
#include "trayicon.h"
#include <QAction>
#include <QApplication>
#include <QMainWindow>
```
#include <QMenu>

Include dependency graph for trayicon.cpp:

```
src/trayicon.cpp
    trayicon.h  QAction  QMainWindow  QMenu  QSystemTrayIcon  QWidget
        QApplication
```

14.50 src/trayicon.h File Reference

#include <QSystemTrayIcon>
#include <QWidget>

Include dependency graph for trayicon.h:

```
src/trayicon.h
    QSystemTrayIcon  QWidget
```

This graph shows which files directly or indirectly include this file:

```
src/trayicon.h
    src/trayicon.cpp
    src/mainwindow.cpp
```
Classes

• class TrayIcon
  
  Handles the systemtray icon and menu.

14.51 src/userinfo.h File Reference

#include <QDateTime>
#include <QString>

Include dependency graph for userinfo.h:

This graph shows which files directly or indirectly include this file:

Classes

• struct UserInfo
  
  Stores key info lines including validity, creation date and more.

14.52 src/usersdialog.cpp File Reference

#include "usersdialog.h"
#include "qtpasssettings.h"
#include "ui_usersdialog.h"
#include <QCloseEvent>
#include <QKeyEvent>
#include <QMessageBox>
#include <QRegularExpression>
#include <QWidget>
#include <utility>

Include dependency graph for usersdialog.cpp:

---

14.53 src/usersdialog.h File Reference

---

#include "userinfo.h"
#include <QDialog>
#include <QList>

Include dependency graph for usersdialog.h:
This graph shows which files directly or indirectly include this file:

Classes

- class UsersDialog

  Handles listing and editing of GPG users.

Namespaces

- Ui

14.54 src/util.cpp File Reference

```cpp
#include "util.h"
#include <QDir>
#include <QFileInfo>
#include <sys/time.h>
#include "qtpasssettings.h"
```

Include dependency graph for util.cpp:
#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.

14.56 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.

### 14.57 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)

14.57.1 Function Documentation

14.57.1.1 operator==()

bool operator== (  
    const NamedValue & a,  
    const NamedValue & b )

Definition at line 28 of file tst_util.cpp.
Index

~ConfigDialog
  ConfigDialog, 52
~DeselectableTreeView
  DeselectableTreeView, 62
~ImitatePass
  ImitatePass, 77
~KeygenDialog
  KeygenDialog, 86
~MainWindow
  MainWindow, 90
~Pass
  Pass, 109
~PasswordDialog
  PasswordDialog, 132
~QtPass
  QtPass, 154
~RealPass
  RealPass, 221
~UsersDialog
  UsersDialog, 264
~tst_util
  tst_util, 258
accept
  UsersDialog, 265
addGPGId
  SettingsConstants, 227
ALLCHARS
  PasswordConfiguration, 128
ALPHABETICAL
  PasswordConfiguration, 128
ALPHANUMERIC
  PasswordConfiguration, 128
alwaysOnTop
  SettingsConstants, 227
animationDelay
  QProgressIndicator, 140
autoclearPanelSeconds
  SettingsConstants, 228
autoclearSeconds
  SettingsConstants, 228
autoPull
  SettingsConstants, 228
autoPush
  SettingsConstants, 228
avoidCapitals
  SettingsConstants, 228
avoidNumbers
  SettingsConstants, 228
boundedRandom
  Pass, 109
cancelNext
  Executor, 66
canDropMimeData
  StoreModel, 245
changeEvent
  MainWindow, 90
CHANGELOG.md, 273
Characters
  PasswordConfiguration, 128
color
  QProgressIndicator, 140, 144
clearClipboard
  QtPass, 155
clicked
  QPushButtonAsQRCode, 146
  QPushButtonShowPassword, 149
  QPushButtonWithClipboard, 151
CLIPBOARD_ALWAYS
  Enums, 47
CLIPBOARD.Never
  Enums, 47
CLIPBOARD_ON_DEMAND
  Enums, 47
clipBoardType
  Enums, 47
closeEvent
  SettingsConstants, 229
config
  MainWindow, 91
ConfigDialog, 49
~ConfigDialog, 52
closeEvent, 53
ConfigDialog, 51
genKey, 53
getPasswordConfiguration, 54
getProfiles, 54
setPasswordConfiguration, 54
setPwgenPath, 54
useAutoclear, 55
useAutoclearPanel, 55
useGit, 56
useOtp, 56
usePwgen, 57
useQrencode, 58
useSelection, 58
useTemplate, 59
useTrayIcon, 59
wizard, 60
CONTRIBUTING.md, 273
Copy
  ImitatePass, 77
  Pass, 109
  RealPass, 222
copyDir
  Util, 267
copyTextToClipboard
  QtPass, 155
created
  UserInfo, 261
critical
  MainWindow, 92
  Pass, 110
CUSTOM
  PasswordConfiguration, 128
data
  StoreModel, 246
dbg
  debughelper.h, 276
ddebughelper.h
dbg, 276
delay
  QProgressIndicator, 144
deselect
  MainWindow, 93
DeselectableTreeView, 60
  ~DeselectableTreeView, 62
  DeselectableTreeView, 61
  emptyClicked, 62
displayedWhenStopped
  QProgressIndicator, 144
dragAndDropInfo, 62
dragAndDropInfoPasswordStore, 63
  isDir, 63
  isFile, 63
  path, 63
dropMimeData
  StoreModel, 247
emptyClicked
  DeselectableTreeView, 62
enabled
  UserInfo, 261
endReencryptPath
  ImitatePass, 78
  MainWindow, 93
 Enums, 47
  CLIPBOARD_ALWAYS, 47
  CLIPBOARD_NEVER, 47
  CLIPBOARD_ON_DEMAND, 47
clipBoardType, 47
  GIT_ADD, 48
  GIT_COMMIT, 48
  GIT_COPY, 48
  GIT_INIT, 48
  GIT_MOVE, 48
  GIT_PULL, 48
  GIT_PUSH, 48
  GIT_RM, 48
  GPG_GENKEYS, 48
  INVALID, 48
  PASS_COPY, 48
  PASS_INIT, 48
  PASS_INSERT, 48
  PASS_MOVE, 48
  PASS_OTP_GENERATE, 48
  PASS_REMOVE, 48
  PASS_SHOW, 48
  PROCESS, 48
  PROCESS_COUNT, 48
error
  Executor, 66
  Pass, 110
eventFilter
  MainWindow, 93
eexec
  Pass, 127
execute
  Executor, 66–68
executeBlocking
  Executor, 69, 70
executeWrapper
  ImitatePass, 78
  MainWindow, 93
executeWrapperStarted
  MainWindow, 94
Executor, 64
  cancelNext, 66
  error, 66
  execute, 66–68
  executeBlocking, 69, 70
  finished, 70
  setEnvironment, 71
  starting, 71
expiry
  UserInfo, 261
INDEX

FAQ.md, 273
FileContent, 72
  getNamedValues, 72
getPassword, 73
getRemainingData, 73
getRemainingDataForDisplay, 74
parse, 74
filterAcceptsRow
  StoreModel, 247
findBinaryInPath
  Util, 269
findPasswordStore
  Util, 269
finished
  Executor, 70
  ImitatePass, 79
  Pass, 111
finishedAny
  Pass, 112
finishedCopy
  Pass, 112
finishedGenerate
  Pass, 113
finishedGenerateGPGKeys
  Pass, 113
finishedGitInit
  Pass, 113
finishedGitPull
  Pass, 113
finishedGitPush
  Pass, 114
finishedInit
  Pass, 114
finishedInserst
  Pass, 115
finishedMove
  Pass, 115
finishedOtpGenerate
  Pass, 115
finishedRemove
  Pass, 116
finishedShow
  Pass, 116
flags
  StoreModel, 248
flashText
  MainWindow, 94
fullyValid
  UserInfo, 260
Generate_b
  Pass, 117
generateGPGKeyPair
  MainWindow, 94
GenerateGPGKeys
  Pass, 117
generateKeyPair
  MainWindow, 95
generateRandomPassword
  Pass, 118
genKey
  ConfigDialog, 53
generate
  SettingsConstants, 229
getAutoclearPanelSeconds
  QtPassSettings, 163
getAutoclearSeconds
  QtPassSettings, 163
getClipboardType
  QtPassSettings, 164
getClipboardTypeRaw
  QtPassSettings, 164
getCurrentTreeViewIndex
  MainWindow, 95
getDir
  Util, 270
getGeometry
  QtPassSettings, 165
getGitExecutable
  QtPassSettings, 165
getGpgExecutable
  QtPassSettings, 166
getGpgHome
  QtPassSettings, 167
getImitatePass
  QtPassSettings, 167
getInstance
  QtPassSettings, 168
getIsAllocated
  TrayIcon, 253
ggetKeygenDialog
  MainWindow, 95
getNamedValues
  FileContent, 72
getPass
  QtPassSettings, 169
getPassExecutable
  QtPassSettings, 170
getPassStore
  QtPassSettings, 171
getPassTemplate
  QtPassSettings, 171
getPassword
  FileContent, 73
  PasswordDialog, 132
getPasswordConfiguration
  ConfigDialog, 54
  QtPassSettings, 172
getPos
  QtPassSettings, 173
getProfile
  QtPassSettings, 173
getProfiles
  ConfigDialog, 54
  QtPassSettings, 174
getPwgenExecutable
  QtPassSettings, 175
getQrencodeExecutable
QtPassSettings, 175

getRealPass
QtPassSettings, 176

getRecipientList
Pass, 118

getRecipientString
Pass, 119

getRemainingData
FileContent, 73

getRemainingDataForDisplay
FileContent, 74

getSaveState
QtPassSettings, 176

getSize
QtPassSettings, 177

getTextToCopy
QPushButtonAsQRCode, 147
QPushButtonWithClipboard, 151

getVersion
QtPassSettings, 178

getWebDavPassword
QtPassSettings, 178

getWebDavUrl
QtPassSettings, 179

getWebDavUser
QtPassSettings, 179

GIT_ADD
 Enums, 48

GIT_COMMIT
 Enums, 48

GIT_COPY
 Enums, 48

GIT_INIT
 Enums, 48

GIT_MOVE
 Enums, 48

GIT_PULL
 Enums, 48

GIT_PUSH
 Enums, 48

GIT_RM
 Enums, 48

gitExecutable
 SettingsConstants, 229

GitInit
 ImitatePass, 79
 Pass, 120
 RealPass, 222

GitPull
 ImitatePass, 80
 Pass, 120
 RealPass, 222

GitPull_b
 ImitatePass, 80
 Pass, 120
 RealPass, 223

GitPush

RealPass, 224

INVALID

Enums, 48

isAddGPGId

QtPassSettings, 180

isAlwaysOnTop

QtPassSettings, 181

isAnimated

QProgressIndicator, 141

isAutoPull

QtPassSettings, 181

isAutoPush

QtPassSettings, 182

isAvoidCapitals

QtPassSettings, 183

isAvoidNumbers

QtPassSettings, 183

isDir

dragAndDropInfoPasswordStore, 63

isDisplayedWhenStopped

QProgressIndicator, 141

isFile

dragAndDropInfoPasswordStore, 63

isFreshStart

QtPass, 157

isHideContent

QtPassSettings, 184

isHideOnClose

QtPassSettings, 185

isHidePassword

QtPassSettings, 185

isLessRandom

QtPassSettings, 186

isMaximized

QtPassSettings, 187

isRunning

SingleApplication, 241

isStartMinimized

QtPassSettings, 187

isTemplateAllFields

QtPassSettings, 188

isUseAutoclear

QtPassSettings, 189

isUseAutoclearPanel

QtPassSettings, 189

isUseGit

QtPassSettings, 190

isUseOtp

QtPassSettings, 191

isUsePass

QtPassSettings, 192

isUsePwgen

QtPassSettings, 192

isUseQrencode

QtPassSettings, 193

isUseSelection

QtPassSettings, 194

isUseSymbols

QtPassSettings, 194

isUseTemplate

QtPassSettings, 195

isUseTrayIcon

QtPassSettings, 195

isUseWebDav

QtPassSettings, 196

isValid

UserInfo, 260

key_id

UserInfo, 262

KeygenDialog, 85

~KeygenDialog, 86

closeEvent, 86

KeygenDialog, 86

keyPressEvent

MainWindow, 96

UsersDialog, 266

length

PasswordConfiguration, 128

lessRandom

SettingsConstants, 230

lessThan

StoreModel, 248

listKeys

Pass, 122, 123

main

main.cpp, 274

main.cpp

main, 274

main/main.cpp, 273

MainWindow, 87

~MainWindow, 90

changeEvent, 90

cleanKeygenDialog, 91

closeEvent, 91

config, 91

critical, 92

deselect, 93

dirReencryptPath, 93

eventFilter, 93

executeWrapperStarted, 94

flashText, 94

generateGPGKeyPair, 94

generateKeyPair, 95

getCurrentTreeViewIndex, 95

getKeygenDialog, 95

keyPressEvent, 96

MainWindow, 89

messageAvailable, 96

on_treeView_clicked, 97

onPush, 97

passGitInitNeeded, 98

passOtpHandler, 98

passShowHandler, 98

passShowHandlerFinished, 99
restoreWindow, 99
setUiElementsEnabled, 100
showStatusMessage, 101
startReencryptPath, 102
userDialog, 102
mainwindow.h
    SingleApplication, 284
marginallyValid
    UserInfo, 260
maximized
    SettingsConstants, 230
messageAvailable
    MainWindow, 96
    SingleApplication, 241
mimeType
    StoreModel, 249
mimeTypes
    StoreModel, 249
Move
    ImitatePass, 82
    Pass, 124
    RealPass, 224
name
    NamedValue, 103
    UserInfo, 262
NamedValue, 103
    name, 103
    value, 103
NamedValues, 104
    NamedValues, 105
takeValue, 105
normalizeFolderPath
    Util, 271
on_treeView_clicked
    MainWindow, 97
onPush
    MainWindow, 97
operator<<
    storemodel.cpp, 301
operator>>
    storemodel.cpp, 301
operator==
    tst_util.cpp, 309
OtpGenerate
    ImitatePass, 82
    Pass, 124
    RealPass, 225
paintEvent
    QProgressIndicator, 141
parse
    FileContent, 74
Pass, 106
    ~Pass, 109
    boundedRandom, 109
    Copy, 109
    critical, 110
error, 110
exec, 127
executeWrapper, 110, 111
finished, 111
finishedAny, 112
finishedCopy, 112
finishedGenerate, 113
finishedGenerateGPGKeys, 113
finishedGitInit, 113
finishedGitPull, 113
finishedGitPush, 114
finishedInit, 114
finishedInsert, 115
finishedMove, 115
finishedOtpGenerate, 115
finishedRemove, 116
finishedShow, 116
Generate_b, 117
GenerateGPGKeys, 117
generateRandomPassword, 118
getRecipientList, 118
getRecipientString, 119
GitInit, 120
GitPull, 120
GitPull_b, 120
GitPush, 120
Init, 121
init, 121
Insert, 122
listKeys, 122, 123
Move, 124
OtpGenerate, 124
Pass, 108
PROCESS, 108
processErrorExit, 124
Remove, 125
Show, 125
startingExecuteWrapper, 125
statusMsg, 126
updateEnv, 126
PASS_COPY
    Enums, 48
PASS_INIT
    Enums, 48
PASS_INSERT
    Enums, 48
PASS_MOVE
    Enums, 48
PASS_OTP_GENERATE
    Enums, 48
PASS_REMOVE
    Enums, 48
PASS_SHOW
    Enums, 48
passExecutable
    SettingsConstants, 231
passGitInitNeeded
    MainWindow, 98
Index

getPwgenExecutable, 175
getQrencodeExecutable, 175
getRealPass, 176
getSavestate, 176
getSize, 177
getVersion, 178
getWebDavPassword, 178
getWebDavUrl, 179
getWebDavUser, 179
initExecutables, 179
isAddGPGId, 180
isAlwaysOnTop, 181
isAutoPull, 181
isAutoPush, 182
isAvoidCapitals, 183
isAvoidNumbers, 183
isHideContent, 184
isHideOnClose, 185
isHidePassword, 185
isLessRandom, 186
isMaximized, 187
isStartMinimized, 187
isTemplateAllFields, 188
isUseAutoclear, 189
isUseAutoclearPanel, 189
isUseGit, 190
isUseOtp, 191
isUsePass, 192
isUsePwgen, 192
isUseQrencode, 193
isUseSelection, 194
isUseSymbols, 194
isUseTemplate, 195
isUseTrayIcon, 196
isUseWebDav, 196
setAddGPGId, 197
setAlwaysOnTop, 197
setAutoclearPanelSeconds, 198
setAutoclearSeconds, 198
setAutoPull, 199
setAutoPush, 199
setAvoidCapitals, 200
setAvoidNumbers, 200
setClipboardType, 201
setGeometry, 201
setGpgExecutable, 202
setGitExecutable, 202
setHideContent, 203
setHideOnClose, 203
setHidePassword, 204
setLessRandom, 204
setMaximized, 204
setPwgenExecutable, 205
setPassStore, 205
setPassTemplate, 206
setPasswordChars, 206
setPasswordCharsselection, 207
setPasswordConfiguration, 207
setPasswordLength, 208
setPos, 208
setProfile, 209
setProfiles, 209
setPwgenExecutable, 209
setQrencodeExecutable, 210
setSavestate, 210
setSize, 211
setStartMinimized, 212
setTemplateAllFields, 212
setUseAutoclear, 212
setUseAutoclearPanel, 213
setUseGit, 213
setUseOtp, 213
setUsePass, 214
setUsePwgen, 214
setUseQrencode, 215
setUseSelection, 215
setUseSymbols, 216
setUseTemplate, 216
setUseTrayIcon, 216
setUseWebDav, 217
setVersion, 217
setWebDavPassword, 218
setWebDavUrl, 218
setWebDavUser, 219

README.md, 274
RealPass, 219
~RealPass, 221
Copy, 222
GitInit, 222
GitPull, 222
GitPull_b, 223
GitPush, 223
Init, 223
Insert, 224
Move, 224
OtpGenerate, 225
RealPass, 221
Remove, 225
Show, 225
receiveMessage
SingleApplication, 242
reencryptPath
ImitatePass, 83
Remove
ImitatePass, 83
Pass, 125
RealPass, 225
restoreWindow
MainWindow, 99

savestate
SettingsConstants, 232
selected
PasswordConfiguration, 128
sendMessage
SingleApplication, 242
setAddGPGId
    QtPassSettings, 197
setAlwaysOnTop
    QtPassSettings, 197
setAnimationDelay
    QProgressIndicator, 142
setAutoclearPanelSeconds
    QtPassSettings, 198
setAutoclearSeconds
    QtPassSettings, 198
setAutoPull
    QtPassSettings, 199
setAutoPush
    QtPassSettings, 199
setAvoidCapitals
    QtPassSettings, 200
setAvoidNumbers
    QtPassSettings, 200
setClipboardTimer
    QtPass, 158
setClipboardType
    QtPassSettings, 201
setClippedText
    QtPass, 158
setColor
    QProgressIndicator, 142
setDisplayWhenStopped
    QProgressIndicator, 143
setEnvironment
    Executor, 71
setFreshStart
    QtPass, 159
setGeometry
    QtPassSettings, 201
setGitExecutable
    QtPassSettings, 202
setGpgExecutable
    QtPassSettings, 202
setHideContent
    QtPassSettings, 203
setHideOnClose
    QtPassSettings, 203
setHidePassword
    QtPassSettings, 204
setLength
    PasswordDialog, 132
setLessRandom
    QtPassSettings, 204
setMaximized
    QtPassSettings, 204
setModelAndStore
    StoreModel, 249
setPass
    PasswordDialog, 133
setPassExecutable
    QtPassSettings, 205
setPassStore
    QtPassSettings, 205
setPassTemplate
    QtPassSettings, 206
setPassword
    PasswordDialog, 134
setPasswordChars
    QtPassSettings, 206
setPasswordCharsSelection
    QtPassSettings, 207
setPasswordCharTemplate
    PasswordDialog, 135
setPasswordConfiguration
    ConfigDialog, 54
    QtPassSettings, 207
setPasswordLength
    QtPassSettings, 208
setPos
    QtPassSettings, 208
setProfile
    QtPassSettings, 209
setProfiles
    QtPassSettings, 209
setPwgenExecutable
    QtPassSettings, 209
setPwgenPath
    ConfigDialog, 54
setQrencodeExecutable
    QtPassSettings, 210
setSavestate
    QtPassSettings, 210
setSize
    QtPassSettings, 211
setStartMinimized
    QtPassSettings, 212
setTemplate
    PasswordDialog, 136
setTemplateAllFields
    QtPassSettings, 212
setTextToCopy
    QPushButtonAsQRCode, 147
    QPushButtonWithClipboard, 152
SettingsConstants, 226
addGPId, 227
alwaysOnTop, 227
autoclearPanelSeconds, 228
autoclearSeconds, 228
autoPull, 228
autoPush, 228
avoidCapitals, 228
avoidNumbers, 228
clipBoardType, 229
gitExecutable, 229
gpgExecutable, 229
gpgHome, 229
groupMainWindow, 229
groupProfiles, 230
hideContent, 230
hideOnClose, 230
hidePassword, 230
lessRandom, 230
maximized, 230
passExecutable, 231
passStore, 231
passTemplate, 231
passwordChars, 231
passwordCharsselection, 231
passwordLength, 231
pos, 232
profile, 232
pwgenExecutable, 232
qrencodeExecutable, 232
savestate, 232
size, 232
splitterLeft, 233
splitterRight, 233
startMinimized, 233
templateAllFields, 233
useAutoclear, 233
useAutoclearPanel, 234
useClipboard, 234
useGit, 234
useOtp, 234
usePass, 234
usePwgen, 234
useQrencode, 235
useSelection, 235
useSymbols, 235
useTemplate, 235
useTrayIcon, 235
useWebDav, 235
version, 236
webDavPassword, 236
webDavUrl, 236
webDavUser, 236
setUiElementsEnabled
MainWindow, 100
setUseAutoclear
QtPassSettings, 212
setUseAutoclearPanel
QtPassSettings, 213
setUseGit
QtPassSettings, 213
setUseOtp
QtPassSettings, 213
setUsePass
QtPassSettings, 214
setUsePwgen
QtPassSettings, 214
setUseQrencode
QtPassSettings, 215
setUseSelection
QtPassSettings, 215
setUseSymbols
QtPassSettings, 216
setUseTemplate
QtPassSettings, 216
setUseTrayIcon
QtPassSettings, 216
setUseWebDav
QtPassSettings, 217
setVersion
QtPassSettings, 217
setVisible
TrayIcon, 254
setWebDavPassword
QtPassSettings, 218
setWebDavUrl
QtPassSettings, 218
setWebDavUser
QtPassSettings, 219
Show
ImitatePass, 84
Pass, 125
RealPass, 225
showHideParent
TrayIcon, 255
showMessage
TrayIcon, 255
showStatusMessage
MainWindow, 101
showTextAsQRCode
QtPass, 159
ShowThis
StoreModel, 250
simpleTransaction, 237
templateAllFields, 233
transactionAdd, 238
transactionDate, 238
transactionEnd, 238
transactionIsOver, 238
transactionStart, 239
SingleApplication, 239
isRunning, 241
mainwindow.h, 284
messageAvailable, 241
receiveMessage, 242
sendMessage, 242
SingleApplication, 240
size
SettingsConstants, 232
sizeHint
QProgressIndicator, 143
splitterLeft
SettingsConstants, 233
splitterRight
SettingsConstants, 233
src/configdialog.cpp, 274
src/configdialog.h, 275
src/debughelper.h, 276
src/deselectabletreeview.h, 277
src/enums.h, 277
src/executer.cpp, 278
src/executer.h, 278
src/filecontent.cpp, 279
src/filecontent.h, 280

Generated by Doxygen
useAutoclearPanel
  ConfigDialog, 55
  SettingsConstants, 234
useClipboard
  SettingsConstants, 234
useGit
  ConfigDialog, 56
  SettingsConstants, 234
useOtp
  ConfigDialog, 56
  SettingsConstants, 234
usePass
  SettingsConstants, 234
usePwgen
  ConfigDialog, 57
  PasswordDialog, 137
  SettingsConstants, 234
useQrencode
  ConfigDialog, 58
  SettingsConstants, 235
userDialog
  MainWindow, 102
UserInfo, 259
  created, 261
  enabled, 261
  expiry, 261
  have_secret, 261
  isValid, 260
  key_id, 262
  marginallyValid, 260
  name, 262
UserInfo, 259
  validity, 262
UsersDialog, 263
~UsersDialog, 264
  accept, 265
  closeEvent, 265
  keyPressEvent, 266
UsersDialog, 264
useSelection
  ConfigDialog, 58
  SettingsConstants, 235
useSymbols
  SettingsConstants, 235
useTemplate
  ConfigDialog, 59
  SettingsConstants, 235
useTrayIcon
  ConfigDialog, 59
  SettingsConstants, 235
useWebDav
  SettingsConstants, 235
Util, 266
  checkConfig, 267
  copyDir, 267
  findBinaryInPath, 269
  findPasswordStore, 269
getDir, 270
  normalizeFolderPath, 271

validity
  UserInfo, 262
value
  NamedValue, 103
version
  SettingsConstants, 236
webDavPassword
  SettingsConstants, 236
webDavUrl
  SettingsConstants, 236
webDavUser
  SettingsConstants, 236
wizard
  ConfigDialog, 60