Qt 程序获取程序所在路径、用户目录路径、临时文件夹等特殊路径的方法

经常我们的程序中需要访问一些特殊的路径，比如程序所在的路径、用户目录路径、临时文件夹等。在 Qt 中实现这几个功能所用的方法虽然都不难，但是各不相同，每次用到时还要现去查，很不方便。因此就写了这篇博客，把这几种需求的实现方式总结了一下。算是个备忘录吧。

程序所在路径

获取程序所在路径，QCoreApplication 类里就实现了相关的功能：

QString QCoreApplication::applicationDirPath()

* 1
* 1

比如我们有一个程序在：

C:/Qt/examples/tools/regexp/regexp.exe

* 1
* 1

那么 qApp->applicationDirPath() 的结果是：

C:/Qt/examples/tools/regexp

* 1
* 1

如果除了程序所在路径，我们还想要程序的完整名称。那么可以这么写：

qApp->applicationFilePath()

* 1
* 1

还是上面的例子，结果是：

C:/Qt/examples/tools/regexp/regexp.exe

* 1
* 1

当前工作目录

QDir 提供了一个静态函数 currentPath() 可以获取当前工作目录，函数原型如下：

QString QDir::currentPath()

* 1
* 1

如果我们是双击一个程序运行的，那么程序的工作目录就是程序所在目录。

如果是在命令行下运行一个程序，那么运行程序时在命令行的哪个目录，那个目录就是当前目录。

用户目录路径

Qt 4 中的方法。下面的方法只对 Qt 4 有效，Qt 5 已经删除了 storageLocation() 方法。

QDesktopServices::storageLocation(QDesktopServices::HomeLocation);

* 1
* 1

Qt 5 中引入的方法。

QStandardPaths::writableLocation(QStandardPaths::HomeLocation);

* 1
* 1

或者

QStandardPaths::standardLocations(QStandardPaths::HomeLocation);

* 1
* 1

这两个方法的区别是 standardLocations() 返回值是 QStringList。当然对于 HomeLocation 来说这个 QStringList 中只有一个 QString。

还有另外一种方法，利用 QDir 类的一个静态函数：

QDir::homePath();

* 1
* 1

我的文档路径

Qt 4 中的方法。下面的方法只对 Qt 4 有效，Qt 5 已经删除了 storageLocation() 方法。

QDesktopServices::storageLocation(QDesktopServices::DocumentsLocation);

* 1
* 1

Qt 5 中引入的方法。

QStandardPaths::writableLocation(QStandardPaths::DocumentsLocation);

QStandardPaths::standardLocations(QStandardPaths::DocumentsLocation);

* 1
* 2
* 1
* 2

桌面路径

Qt 4 中的方法。下面的方法只对 Qt 4 有效，Qt 5 已经删除了 storageLocation() 方法。

QDesktopServices::storageLocation(QDesktopServices::DesktopLocation);

* 1
* 1

Qt 5 中引入的方法。

QStandardPaths::writableLocation(QStandardPaths::DesktopLocation);

QStandardPaths::standardLocations(QStandardPaths::DesktopLocation);

* 1
* 2
* 1
* 2

程序数据存放路径

通常我们会将程序所需的一些数据存入注册表。但是有时需要存储的数据太多，放在注册表中就不适合了。这时我们就要找个专门的地方来放数据。以前我喜欢将数据直接放到程序所在目录，但是后来发现我的程序运行时经常没有权限对这个目录下的文件进行写操作。后来发现其实 Qt 早就替我们考虑过这些问题了。

Qt 4 中的方法。下面的方法只对 Qt 4 有效，Qt 5 已经删除了 storageLocation() 方法。

QDesktopServices::storageLocation(QDesktopServices::DataLocation);

* 1
* 1

Qt 5 中引入的方法。

QStandardPaths::writableLocation(QStandardPaths::AppDataLocation);

QStandardPaths::standardLocations(QStandardPaths::AppDataLocation);

* 1
* 2
* 1
* 2

Qt 5.5 中引入了另一种方法：

QStandardPaths::writableLocation(QStandardPaths::AppConfigLocation);

QStandardPaths::standardLocations(QStandardPaths::AppConfigLocation);

* 1
* 2
* 1
* 2

这个方法一般来说和上面的方法得到的结果是相同的。按照 Qt 帮助文档的解释，这个方法可以确保返回的路径非空。所以我认为应该优先选用这个方法。

临时文件路径

Qt 4 中的方法。下面的方法只对 Qt 4 有效，Qt 5 已经删除了 storageLocation() 方法。

QDesktopServices::storageLocation(QDesktopServices::TempLocation);

* 1
* 1

Qt 5 中引入的方法。

QStandardPaths::writableLocation(QStandardPaths::TempLocation);

QStandardPaths::standardLocations(QStandardPaths::TempLocation);

* 1
* 2
* 1
* 2

更传统的方法是利用 QDir 的一个静态函数 tempPath()。

QDir::tempPath();

* 1
* 1

在这个目录下生成临时文件和临时目录需要用到另外两个类： QTemporaryFile 和 QTemporaryDir。就不展开介绍了，大家可以参考 qt 的帮助文档。

至此，常用的各种特殊路径就介绍的差不多了。剩下还有些不常用的，可以参考 QStandardPaths 类的介绍。