

QSettings简介:

用户对应用程序经常有这样的要求：要求它能记住它的settings，比如窗口大小，位置，一些别的设置，等等这些都可以通过QSettings来实现。这些settings一般都是存在系统里的，比如windows一般都写在系统注册表或者写INI文件，许多应用程序是用INI文件来实现的。而QSettings就是提供了一种方便的方法来存储和恢复应用程序的settings。

Qvariant

- QSettings的API是基于Qvariant，Qvariant是一种数据类型的集合，它包含了大部分通常的Qt数据类型，比如QString，QRect，QImage，等等。
- 注意：因为QVariant是不会提供所有数据类型的转化的，比如有toInt(),toPoint(),toSize(),但是却没有对QColor，QImage和QPixmap等数据类型的转化

QSettings使用:

写入设置

- 当我们创建一个QSettings的对象时，我们需要传递给它两个参数，与字典数据类型类似，它有key，以及对应的value。比如下面例子：

```
1 settings.setValue("pos",QVariant(Mainwindow.pos()))
2 settings.setValue("size",QVariant(Mainwindow.size()))
```

- 上面两句就是把当前窗口的位置和大小两个信息记录到了settings中，其中的key就是" pos" 和" size" 两个QString类型，而它所对应的值就是QVariant类型的。当然如果我们要写的key已在settings中存在的话，则会覆盖原来的值，写入新值。

读取设置

```
1 Pos = settings.value("pos").toPoint()
2 Size = settings.value("size").toSize()
```

- 当然如果key所对应的value是int型的，也可toInt(),如果没有我们要找的key，则会返回一个null QVariant如果用toInt的话会得到0。
- 那么实际应用中我们一般会如下：

```
1 pos= settings.value("pos", QVariant(QPoint(200, 200))).toPoint()
```

```
2 size= settings.value("size", QVariant(QSize(400, 400))).toSize()
3 self.resize(size)
4 self.move(pos)
```

- 如果settings里有以前存下的(用setValue设置的)pos和size的值, 则读取, 如果没有, 不会返回null, 而会使用我们给它的起始默认值, 即应用程序第一次运行时的情况。