### 65访问网络

Qt 进行网络访问的类是QnetworkAccessManager。为了使用网络相关的类，需要在 pro 文件中添加QT += network。QNetworkAccessManager类允许应用程序发送网络请求以及接受服务器的响应。事实上，Qt 的整个访问网络 API 都是围绕着这个类进行的。QNetworkAccessManager保存发送的请求的最基本的配置信息，包含了代理和缓存的设置。最好的是，这个 API 本身就是异步设计，这意味着我们不需要自己为其开启线程，以防止界面被锁死。异步的设计避免了这一系列的问题，但是却要求我们使用更多的代码来监听返回。这类似于我们前面提到的QDialog::exec()和QDialog::show()之间的区别。QNetworkAccessManager是使用信号槽来达到这一目的的。

一个应用程序仅需要一个QNetworkAccessManager类的实例。所以，虽然QNetworkAccessManager本身没有被设计为单例，但是我们应该把它当做单例使用。一旦一个QNetworkAccessManager实例创建完毕，我们就可以使用它发送网络请求。这些请求都返回QNetworkReply对象作为响应。这个对象一般会包含有服务器响应的数据。

### 64 Qsettings

简述：用Qt编写界面时，你希望本地信息可保存，可以使用轻量级数据库sqlite，也可使用QSettings读写配置文件。QSettings类提供“持久性”、平台独立的应用程序设置，即程序关闭后QSettings把窗体大小，位置，用户设置的Option 保存起来，下次打开程序时再重新加载。

保存参数：

QSettings settings(QCoreApplication::applicationDirPath()+"/settings.ini", QSettings::*IniFormat*);

settings.setValue("MainWindow/size", size());

settings.setValue("MainWindow/pos", pos());

将在路径QCoreApplication::applicationDirPath()下自动生成settings.ini文件,settings.ini文件内如如下：

[MainWindow]

size=@Size(263 493)

pos=@Point(485 363)

加载参数：

QSettings settings(QCoreApplication::applicationDirPath()+"/settings.ini", QSettings::*IniFormat*);

resize(settings.value("MainWindow/size", QSize(400, 400)).toSize());

move(settings.value("MainWindow/pos", QPoint(200, 200)).toPoint());

如该变了窗口的大小和位置，则在下次重新打开时是在改变了大小和位置的地方。

### 国际化

翻译文本步骤

1. 需要翻译的文本使用tr(“待翻译文本”)；
2. 在.pro文件中增加TRANSLATIONS = International\_ZH\_CN.ts，International\_ZH\_CN.ts为文件名和后缀，文件名可随意取。
3. 在项目文件列表中右击.pro文件，在弹出的菜单中选择“在此弹出命令提示”。输入lupdate international.pro，international.pro为.pro文件。

或者点击 工具》外部》QT语言家》更新翻译，之后将在项目文件夹中生成.ts文件。

1. 使用QT中的Linguist软件打开上一步生成的.ts文件并进行翻译。翻译完成后保存退出。
2. 按照第3步的方法重新进入命令行，输入lrelease international.pro，回车确定后将生成.qm文件。

或则点击 工具》外部》QT语言家》发布翻译。

1. 在main.cpp文件中添加头文件#include <QTranslator> ，再在QApplication a(argc, argv);后面添加如下代码：

QTranslator translator;

translator.load("D:/Users/Admin/Documents/International/International\_ZH\_CN.qm");

//或者translator.load("../International/International\_ZH\_CN.qm");

a.installTranslator(&translator);