1 C++对象在QML中使用

1 C++类继承自QObject 或者QQuickItem(可视元素，相当于继承自Item)

或者QDeclarativeItem(包含declarative模块)

2 在main中使用qmlRegisterType<>()注册成QML类型

3 在QML中实例化并调用

2 C++创建需要暴露的对象给QML

1 暴露方法给QML

1 在函数申明前添加Q\_INVOKABLE宏

2 申明成public slots

2 暴露数据元素给QML

使用QPROPERTY在元对象系统中申明

（元对象系统申明的元素，在调用元素“=”操作的时候自动调用写函数）

3 暴露已经存在的C++对象给QML

MyClass myObj;  
QDeclarativeEngine \*engine=viewer.engine();  
QDeclarativeContext \*context=engine->rootContext();  
context->setContextProperty("myObjectExposeByCXProperty", &myObj);

4 信号与槽

1 QML可以连接C++的信号

2 QML可以绑定C++的数据元素，在被绑定的元素变化，或者QML调用C++某些函数的时候，触发C++的信号

C++调用QML函数

所有的QML函数都通过meta-object system暴露Qt C++端，在Qt C++端可以使用QMetaObject::invokeMethod()方法直接调用

具体方式参照小儿黑资料

更多例程可以在SDK的安装目录中：\QtSDK\Examples\4.7\declarative\tutorials\extending 看到。