# 图标设置

QT中设置应用程序图标：在 .pro 文件内添加 “RC\_ICONS = xxx.ico” 语句；

QT中为组件设置图标，因为有可能选择资源会程序崩溃，选择文件会不能正常显示：ui->openCamera->setIcon(QIcon(":/images/open.png"));

# 函数用法

## QPixmap用法总结

QPixmap与QImage的区别:

QPixmap依赖于硬件，QImage不依赖于硬件。

QPixmap主要是用于绘图，针对屏幕显示而最佳化设计，QImage主要是为图像I/O、图片访问和像素修改而设计的。

当图片小的情况下，直接用QPixmap进行加载，画图时无所谓，当图片大的时候如果直接用QPixmap进行加载，会占很大的内存，一般一张几十K的图片，用QPixmap加载进来会放大很多倍，所以一般图片大的情况下，用QImage进行加载，然后转乘QPixmap用户绘制。QPixmap绘制效果是最好的。

Qpixmap用法:(转自http://blog.csdn.net/caihuisinx/article/details/7343916)

我想qt 中QPixmap这个类大家都很熟悉，它可以很简单的在标签上贴图；例如：

QPixmap p;

p.load("1.png")；

label->setPixmap(p);

就这么简单，但是我今天突然发现个问题，就是：

QPixmap p;

p.load("1.png")；

label->setPixmap(p);

 p.load("2.png");

 label->setPixmap(p);

p.load("1.png")；

label->setPixmap(p);

当用同一个p，即全局变量时，你加载图片1，显示图片1；再加载图片2，显示图片2；可当你再想变回图片一时，你再加载图片一，不好使了，变不回来了，还是显示图片二，你在加载个图片三，好使，再想变回图片二，不好使了。。。。。。

这是为神马呢？？？

原来，p.load（）是将图片加载到缓冲区中：QPixmapCache，当你加载图片二是，图片一并没有被覆盖，此时缓冲区内有图片一和图片二，当你再次加载图片一时，以为缓冲区里应经有了，所以p.load()，直接返回true；但是图片仍是图片二，所以不能显示回图片一！

解决办法很简单，有三种办法：

一、     直接定义局部变量，这样每次都是新的缓冲区；

二、     当再次加载图片时，先将缓冲区清空，    QPixmapCache::clear();然后再加载；

三、     在初始化函数里写：QPixmapCache::setCacheLimit(1);设置缓冲区内只能放一张图片，这样就会替换原来的图片，就可以加载新的图片了！

## Connect函数

[static] [**QMetaObject::Connection**](qmetaobject-connection.html) QObject::**connect**(const [**QObject**](qobject.html#QObject) \**sender*, const **char** \**signal*, const [**QObject**](qobject.html#QObject) \**receiver*, const **char** \**method*, [**Qt::ConnectionType**](qt.html#ConnectionType-enum) *type* = Qt::AutoConnection)

使用该函数时，SIGNAL和SLOT中的参量函数的参数一定不能有变量名，只能存在变量类型，比如下边两个例子：

正确用法：

[**QLabel**](../qtwidgets/qlabel.html) **\*label = new** [**QLabel**](../qtwidgets/qlabel.html)**;**

[**QScrollBar**](../qtwidgets/qscrollbar.html) **\*scrollBar = new** [**QScrollBar**](../qtwidgets/qscrollbar.html)**;**

[**QObject**](qobject.html#QObject)**::connect(scrollBar,SIGNAL(valueChanged(int)),label,SLOT(setNum(int)));**

**错误用法：**

[**QObject**](qobject.html#QObject)**::connect(scrollBar, SIGNAL(valueChanged(int value)), label, SLOT(setNum(int value)));**