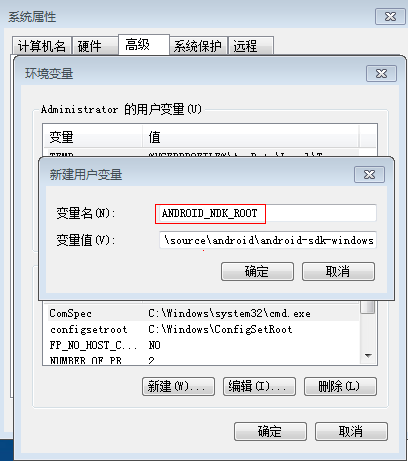
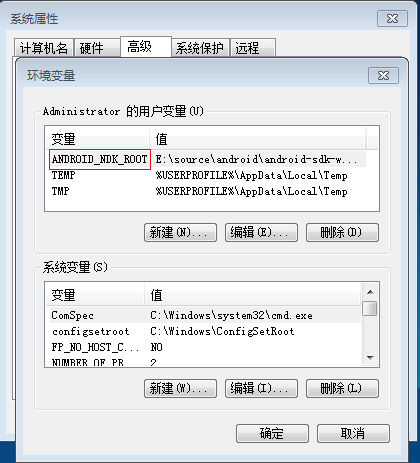
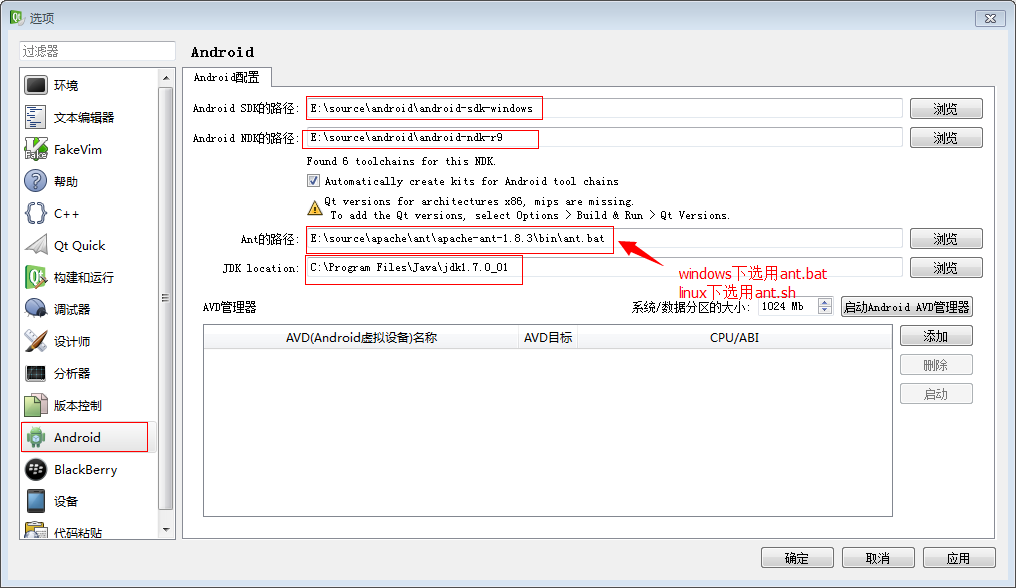
QT开发环境设置：







解析Qt资源文件使用

本文介绍的是Qt资源文件使用，和大部分GUI框架设计工具一样，Qt也引入了资源文件系统。用于方便地将一些二进制文件(主要是图片文件)编译进可执行程序中，免去再发布应用的时候附带其他文件的麻烦。

Qt的资源文件

是以.qrc为后缀名的XML文本格式。在Qt工程中使用资源文件的话，需要在.pro工程文件加入如下代码

RESOURCES     = file.qrc

在工程文件中加入这样语句后，qmake系统会自动调用RCC(资源文件编译器)将file.qrc转化成qrc\_file.cpp，最后使用gcc进行正常的编译。

资源文件的基本格式

来看一个简单的.qrc例子

images/copy.png

images/cut.png

images/new.png

images/open.png

images/paste.png

images/save.png

主要部分和之间的代码，其它部分拷贝过来就可以了,这里的”images /”是相对于你资源文件的路径，和资源文件的格式没有关系。那么你可以在你的源代码中使用QFile访问它们，在本例子中，因为是图片文件，那么你也可以直接用QImage访问它们,不过前面需要加上前缀”:/”，形式为

QImage(”:/images/copy.png”);

资源文件的别名

如果你的资源文件比较多，而且放在一个比较深的目录里，可以通过别名简化

images/cut.png

有了别名后，那么在.cpp文件中可以用别名进行访问

QImage(”:/cut-img.png”);

资源文件的国际化

有些资源文件可能根据使用语言的不同，而使用不同的资源文件。也可以通过别名加语言选项来实现，我没看代码

cut.jpg

cut\_fr.jpg

在使用上面资源的.cpp代码中，根据系统LANG的不同，可以通过同一语句QImage(”:/cut.jpg”)而能够访问到不同的资源文件。

小结：Qt资源文件使用内容介绍完了，从资源文件的基本格式到资源文件的格式化，一一详细介绍了QT文件。希望这些内容对你有所帮助，有情况的话，也可以留言。

**解析Qt资源文件使用**

本文介绍的是**Qt资源文件**使用，和大部分GUI框架设计工具一样，**Qt**也引入了**资源文件**系统。用于方便地将一些二进制**文件**(主要是图片**文件**)编译进可执行程序中，免去再发布应用的时候附带其他文件的麻烦。

**Qt**的**资源文件**

是以.qrc为后缀名的XML文本格式。在Qt工程中使用**资源文件**的话，需要在.pro工程文件加入如下代码

1. RESOURCES     = file.qrc

在工程文件中加入这样语句后，qmake系统会自动调用RCC(**资源文件**编译器)将file.qrc转化成qrc\_file.cpp，最后使用gcc进行正常的编译。

**资源文件的基本格式**

来看一个简单的.qrc例子

1. images/copy.png
2. images/cut.png
3. images/new.png
4. images/open.png
5. images/paste.png
6. images/save.png

主要部分和之间的代码，其它部分拷贝过来就可以了,这里的”images /”是相对于你**资源文件**的路径，和**资源文件**的格式没有关系。那么你可以在你的源代码中使用QFile访问它们，在本例子中，因为是图片文件，那么你也可以直接用QImage访问它们,不过前面需要加上前缀”:/”，形式为

1. QImage(”:/images/copy.png”);

**资源文件的别名**

如果你的**资源文件**比较多，而且放在一个比较深的目录里，可以通过别名简化

1. images/cut.png

有了别名后，那么在.cpp文件中可以用别名进行访问

1. QImage(”:/cut-img.png”);

**资源文件的国际化**

有些**资源文件**可能根据使用语言的不同，而使用不同的**资源文件**。也可以通过别名加语言选项来实现，我没看代码

1. cut.jpg
2. cut\_fr.jpg

在使用上面**资源**的.cpp代码中，根据系统LANG的不同，可以通过同一语句QImage(”:/cut.jpg”)而能够访问到不同的**资源文件**。

小结：**Qt资源文件**使用内容介绍完了，从资源文件的基本格式到资源文件的格式化，一一详细介绍了QT文件。希望这些内容对你有所帮助，有情况的话，也可以留言。

QVideoFrame使用时，必须要先map（），才能把用bits（），用完后unmap（）。例如：

void CFrmPlayer::*present*(const QVideoFrame &frame)

{

if(QVideoFrame::Format\_RGB32 != frame.pixelFormat())

{

qDebug("CFrmPlayer::ShowFrame:frame pixel format is not Format\_RGB32");

return;

}

QVideoFrame f(frame);

if(!f.map(QAbstractVideoBuffer::ReadOnly))

return;

if(NULL == m\_pVideoFrame)

{

m\_pVideoFrame = new QXmppVideoFrame(f.mappedBytes(),

f.size(),

f.bytesPerLine(),

QXmppVideoFrame::Format\_RGB32);

}

if(m\_pVideoFrame)

{

memcpy(m\_pVideoFrame->bits(), f.bits(), m\_pVideoFrame->mappedBytes());

}

f.unmap();

this->update();

return;

}