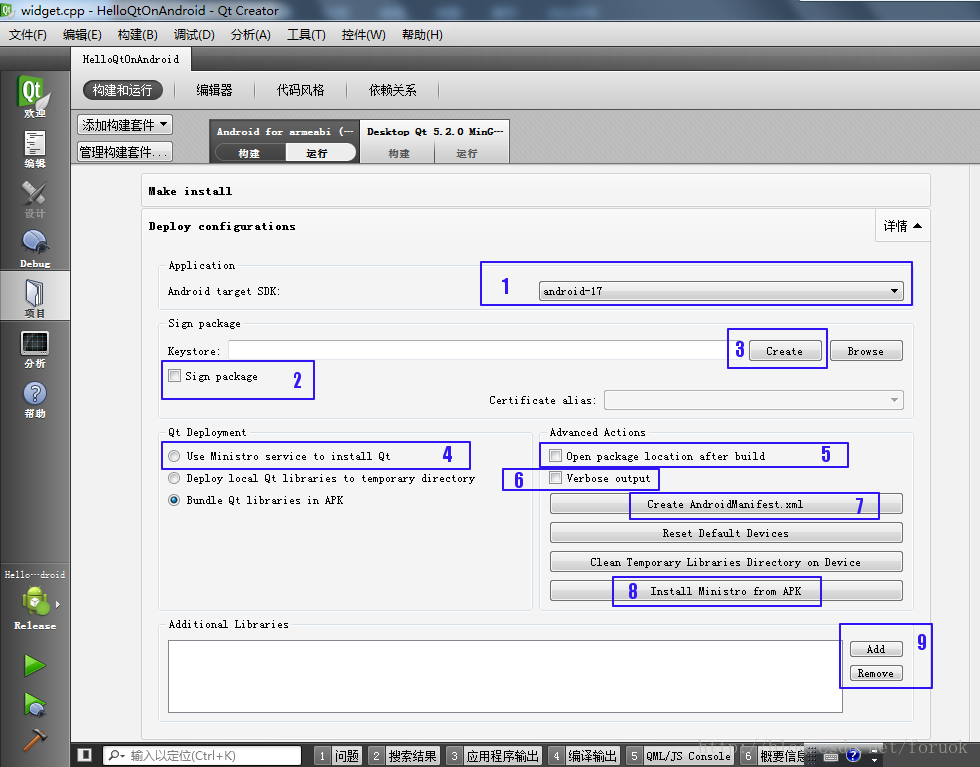
  Android 系统版本很多，较新的 4.4 ，较老的 2.3 ，都有人用。 Qt on Android 开发的 Android 应用，默认在 Android 设备上是非全屏的。而有些应用的需求是全屏显示，比如游戏。那么怎么样才能做到这点呢？

    我们需要分 Android 系统版本来说。

**Android 2.x**

    对于 4.0 以前，比如 2.2 、 2.3 的系统，可以修改 AndroidManifest.xml 文件来实现全屏。

    首先需要为 Qt on Android 应用生成一个 AndroidManifest.xml 文件，具体参考《 [Qt on Android：图文详解Hello World全过程](http://blog.csdn.net/foruok/article/details/23528293" \t "_blank)》一文。创建 AndroidManifest.xml 的示意图如下：



    点击上图中标号为 7 的那个按钮即可创建一个 AndroidManifest.xml 文件。该文件默认生成在 Qt 项目目录下的 android 子目录内。

    然后需要打开 AndroidManifest.xml 文件，找到 Activity 标签，为其加入下面这个属性：

**[html]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. android:theme="@android:style/Theme.NoTitleBar.Fullscreen"

    注意和前、后的其它属性之间留空格。

    在 2.x 系统上运行你的 Qt on Android 应用吧，已经全屏了，连标题栏都没有了。

**Android 4.x**

    Android 系统版本到了 4.0 以后，前面那种设置活动主题的方式无效了。我们需要在 Java 代码中为 Activity 设置一个全屏的窗口标记。

    这在 Java 撰写的 Android 应用上很简单，只需要在 Activity 的 onCreate() 方法中加入下面的代码：

**[java]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. getWindow().addFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN);

    可是到了 Qt on Android 应用，就稍微有些复杂了。

    一步一步说吧，这次以我们的 Qt Quick App 为例来说明，先参考《[Qt Quick 之 Hello World 图文详解](http://blog.csdn.net/foruok/article/details/28850879" \t "_blank)》一文创建一个项目，我演示需要，还是以 HelloQtQuickApp 为基础修改的。

    先看下修改前的效果图（我在 4.0.3 系统的机顶盒上运行）：



    注意到图片下方的状态栏了吧。

**生成 AndroidManifest.xml**

     参考上一节吧，把 Activity 的 Theme 属性也设置一下。

**为 Qt 项目添加 Java 文件**

    话说 Qt on Android 应用， Qt 部分的代码以 JNI 的方式经由 Java 代码启动的。为了能够启动 Qt 代码，进入 Qt 的世界， Qt 框架提供了 QPA 层以及必要的 Activity 、 Application 实现。详情参考《[Qt on Android Episode 1（翻译）](http://blog.csdn.net/foruok/article/details/18697377" \t "_blank)》和《[Qt for Android 部署流程分析](http://blog.csdn.net/foruok/article/details/17796017" \t "_blank)》两篇文章。

    这里以 Qt 5.2.0 为例来作一简单说明。

**QtActivity 简析**

    如果你观察 AndroidManifest.xml 文件，会发现里面的 Activity 标签，指定的 activity 名字是："org.qtproject.qt5.android.bindings.QtActivity" 。没错，就是它！它是由 Qt 框架提供的，你 Qt on Android 应用的入口。 QtActivity 的位置在这里：

**[plain]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. c:\qt\Qt5.2.0\5.2.0\android\_armv5\src\android\java\src\org\qtproject\qt5\android\bindings\QtActivity.java

    感兴趣的可以打开这个 Java 文件研究一下。

    下面是 QtActivity.java 的 onCreate 函数的摘录：

**[java]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. ...
2. @Override
3. **public** **void** onCreate(Bundle savedInstanceState)
4. {
5. **super**.onCreate(savedInstanceState);
6. **try** {
7. setTheme(Class.forName("android.R$style").getDeclaredField(QT\_ANDROID\_DEFAULT\_THEME).getInt(**null**));
8. } **catch** (Exception e) {
9. e.printStackTrace();
10. }
12. **if** (Build.VERSION.SDK\_INT > 10) {
13. **try** {
14. requestWindowFeature(Window.**class**.getField("FEATURE\_ACTION\_BAR").getInt(**null**));
15. } **catch** (Exception e) {
16. e.printStackTrace();
17. }
18. } **else** {
19. requestWindowFeature(Window.FEATURE\_NO\_TITLE);
20. }
21. ...
22. }
23. ...

    如你所见， QtActivity 区分了 SDK 版本， 10 以上的请求 FEATURE\_ACTION\_BAR 特性， 10 以下（Android 2.3及以下）的版本不要 title （太难看了，只显示应用名字，木啥用）。

    你懂的，  Qt Creator 可以编译 Java 源码（通过调用 Ant ）， QtActivity 就是这样编译到你的 APK 中的。所以，我们也可以添加自己的 Java 代码喽。

    我们要添加全屏标记，需要继承 QtActivity 类来实现一个新的 Activity 。当然如果你想粗暴地解决这个问题，可以直接修改 QtActivity.java ，直接加一行代码即可。什么代码？看下面。

**实现QtFullscreenActivity**

    QtFullscreenActivity 继承自 QtActivity 类，代码非常简单。如下：

**[java]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. **package an.qt.helloQtQuickApp;  //程序 要按照报名路径放置**
2. **import** android.content.Context;
3. **import** android.content.Intent;
4. **import** android.app.PendingIntent;
5. **import** android.util.Log;
6. **import** android.os.Bundle;
7. **import** android.view.WindowManager;
9. **public** **class** QtFullscreenActivity **extends** org.qtproject.qt5.android.bindings.QtActivity
10. {
11. **private** **final** **static** String TAG = "QtFullscreen";
12. @Override
13. **public** **void** onCreate(Bundle savedInstanceState) {
14. **super**.onCreate(savedInstanceState);
15. **getWindow().addFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN);**
16. }
17. }

    上面代码中 "getWindow()" 起始的那行，就是设置全屏标记的。

**添加 QtFullscreenActivity 到 pro 文件**

**我把 QtFullscreenActivity.java 文件放在了 android/src/an/qt/helloQtQuickApp/ 下面，所以 pro 文件中相关的部分是酱紫的：**

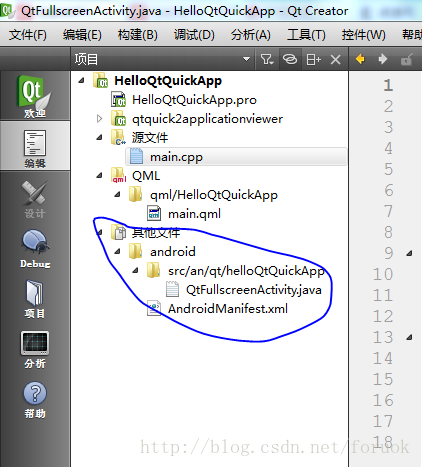
**//一定得在android/src文件夹下面放置才行！！！**

**[plain]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. OTHER\_FILES += \
2. android/AndroidManifest.xml \
3. android/src/an/qt/helloQtQuickApp/QtFullscreenActivity.java

    需要注意的是， **Java 代码中的包名（QtFullscreenActivity第一行），和路径是严格匹配的**， **Java类名则和源代码文件名严格匹配。**

    然后你可以看到下面的项目视图了：



**修改 AndroidManifest.xml**

    这是改动的最后一步了。

    我们实现的 Activity 名字是 QtFullscreenActivity ，所以呢， AndroidManifest.xml 文件中也要修改，保持一致。见下面：

**[html]** [view plaincopy](http://blog.csdn.net/foruok/article/details/38265349)

1. **<?xml** version='1.0' encoding='utf-8'**?>**
2. **<manifest** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:versionCode="1"
4. android:installLocation="auto"
5. package="an.qt.helloQtQuickApp"
6. android:versionName="1.0"**>**
7. **<application**
8. android:name="org.qtproject.qt5.android.bindings.QtApplication"
9. android:label="@string/app\_name"**>**
10. **<activity**
11. **android:name="an.qt.helloQtQuickApp.QtFullscreenActivity"**
12. android:configChanges="orientation|uiMode|screenLayout|screenSize|smallestScreenSize|locale|fontScale|keyboard|keyboardHidden|navigation"
13. **android:theme="@android:style/Theme.NoTitleBar.Fullscreen"**
14. android:label="@string/app\_name"
15. android:screenOrientation="unspecified"**>**
16. **<intent-filter>**
17. ...
18. **</manifest>**

    好啦，到此为止，全部改动就绪。运行我们的 HelloQtQuickApp 吧。

    下面是现在运行的效果：



    好啦，到此结束。