向Qt for Symbian移植Android（Java）应用

简介

作为开放手持设备联盟和Google发起的项目，Android已发展成针对移动电话的首个开放性免费平台。该项目自2007年11月5日启动，当时Google、英特尔、T-Mobile、Sprint、宏达国际、高通和摩托罗拉等决定要实现一个雄伟目标：提供各种服务，让消费用户享有极为优秀的用户体验。

Android平台的重要性之一就是：手机厂商和无线运营商能为其产品和服务提供合适的Android版本。该特性将对成本降低及产品创新产生直接影响。于是， Android就被视为智能手机移动平台中的崭新而具潜力的选择。

另一方面， Qt已被视为最新移动编程世界中功能强大的组件。Qt在桌面系统中的使用最为令人瞩目，如KDE、Opera、Skype和VirtualBox，但是最近它被移植到了诺基亚移动平台，如S60和maemo。自1991年由Trolltech发布，Qt以两种不同的许可证（开源许可证和商业许可证）发行，这使Qt更适合非GPL的开源项目，同时也适合商业用户。2008年6月诺基亚收购了Trolltech，目的是加速移动终端和桌面应用的跨平台软件战略的实施。2008年9月29日，诺基亚将Trolltech改名为Qt。

本文提供的信息指引大家在开发中将Android应用移植到Qt。

Android平台概述

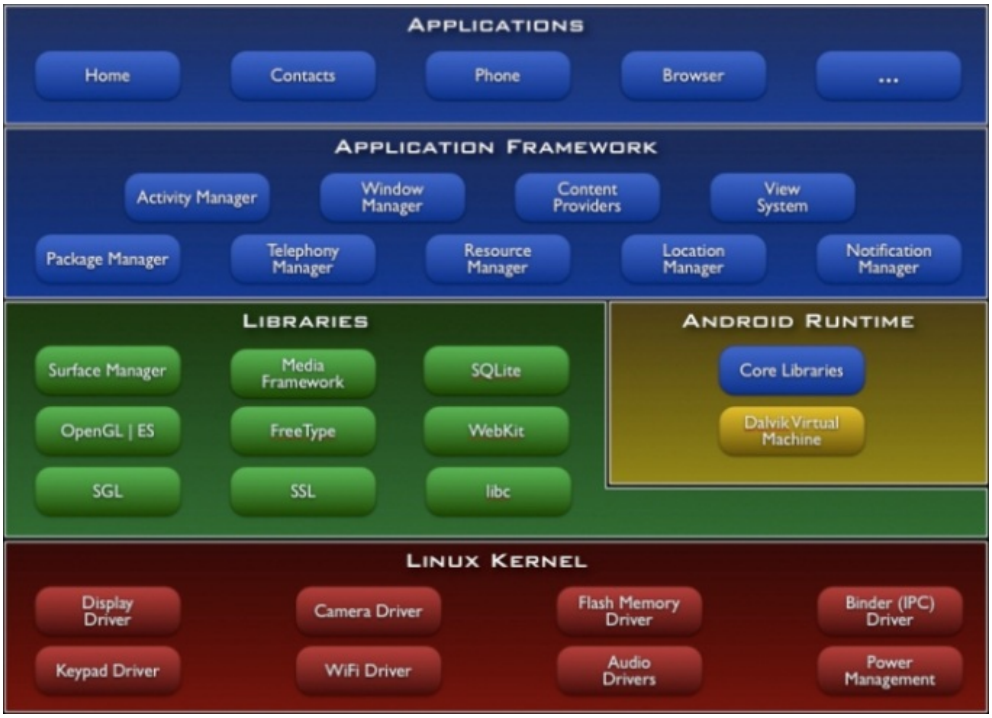
Android应用使用Java编程语言（Dalvik虚拟机）开发，而诸如触摸屏和各种存储功能等终端服务则可通过Google services API访问。用C或其它任何语言所编写的应用也有可能运行，不过需要先将这些应用编译成本地代码后才行，然而这种开发路径并不获Google的正式支持。

自2008年10月起Android已作为开源项目（使用Apache许可证）供大家使用。此后，手机生产商和无线运营商可以自由地向其产品中添加各种封闭性的和专属性的扩展。

尽管Android是基于Linux内核，但按Google的说法，它却不是一个Linux操作系统。此外，它没有本地窗口系统，也不支持全套标准Linux库，包括GNU C库。这个特点就使现有Linux应用或库的重用变得很困难。Android也不使用诸如J2SE和J2ME那样的标准Java APIs。结果是：为这些平台所编写的应用与为Android平台所编写的应用不能兼容。Android仅重用Java语言语法，却并不提供与J2SE或J2ME绑定的完整类库和APIs。

下图说明了目前的Android架构（来自Android开发者指南）。

Copyright © Nokia 2010-2013. All rights reserved.



系统通过一些系统驱动程序（如照相机，显示屏，WiFi和键盘等）访问移动电话的各项资源。之上的一层则由Andriod库和运行系统（RunTime）组成。最后一层Android提供了一套应用框架库，使库扩展和新应用创建均成为可能。

Android可以重用其它应用的其它一些组件。例如，你需要重用适合的滚动条组件，并使其也能用于其它系统，同时可以调用这样的组件来为自己工作。为此Android系统被设计为：在需要用到系统的任何部分时，系统就会启动一个应用进程，并针对那个部分的Java对象进行实例化。这样Android并不提供如main函数这样一个入口，而只提供了一些基本组件：activity（活动），services（服务），broadcast receives（广播接收器）和content providers（内容提供器）。

Activities（活动）代表Android应用的屏幕。从某个活动中你可以显示按钮、标签、菜单等。所有的活动都是*android.app.Activity* 类的子类。**Services（服务）**不可见，但却在后台运行。例如，一个服务能在用户执行其它任务时播放音乐。每个服务都继承自*android.app.Service*基类。**Broadcast receives（广播接收器）**是一些组件，它们可以接收并响应不同的广播告示，例如电量低这样的消息。所有的接收器都继承自*android.content.BroadcastReceive*基类。**content provider（内容提供器）**负责提供可供其它应用使用的应用数据。有了内容提供器，在不同应用之间共享数据就变为可能。所有的内容提供器都继承自*android.content.ContentProvider*基类。

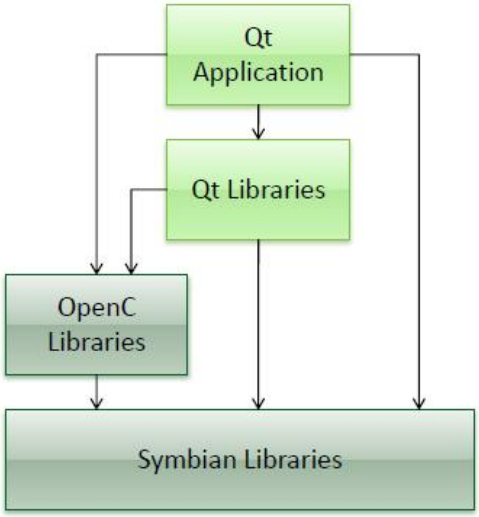
Android开发环境包括：Android SDK、Android源代码，和有助于快速编制Android应用的一些可选的集成开发环境。Android软件开发工具包（SDK）由一些库和工具组成，包括一个能运行应用的模拟器。Android SDK可用于Windows、Mac OS X，和Linux。很多种集成开发环境都提供Android开发支持，如Eclipse（针对Andriod的Eclipse插件）。

Qt S60 概述

Qt应用是用C++加上编译前能生成标准C++代码的附加预处理器所实现的几种非标准扩展所开发的。Qt也提供了与其它几种编程语言的绑定，如Python、Ruby，和Perl。

Qt得到广泛使用得益于其GUI widget，但它也提供了一整套的非GUI相关特性：SQL数据库访问、XML解析、线程管理、网络支持及用于文件处理的一个统一跨平台API。Qt已被成功地移植到了S60诺基亚平台上。Qt将 "让你能创建具创新性用户体验的优秀应用，同时又能快速进入市场"。比较Android，Qt不应该被视为移动电话平台，而应该视为一个有趣而易用的编程框架，定会对移动编程事业做出可观贡献。

Qt现已被移植到了S60平台，却仍有一些APIs还没有被实现，如访问照相机和名片簿模块。不过，通过对某些Symbian应用的回调还是可以访问这些系统服务的。



为开发Qt Symbian应用，你需要安装Qt S60 Nokia SDK，SDK由组成库和开发工具（包括针对S60环境的模拟器）组成。此外，Carbide.C++ IDE（基于Eclipse框架）提供了完整的开发环境，对开发伙伴们提供了很大程度的帮助。你可以在诺基亚论坛维基百科中读到有关Qt的有趣文章（请参阅Qt 分类）。

(c) Copyright Nokia 2010-2013. All rights reserved.

在Carbide 集成开发环境上构建Qt应用是很简单的，这些应用可以在S60模拟器甚至你的终端上（首先需要安装Qt库）启动。任何Qt应用的入口都是main方法（相当于Android应用从某个Activity，即活动中启动）。

范例

这是Android上的一个"Hello World ”简单应用。

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql, povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
package helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {

   @Override
   public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       TextView tv = new TextView(this);
       tv.setText("Hello World");
       setContentView(tv);
   }
}
```

如前文所述，任何Android应用的入口是一个Activity（活动，更精确的说法是android.app.Activity#onCreate(Bundle) 方法）。Android没有标签类，但却有能用作标签的文本视图。然后，我们创建一个文本视图，接着以刚刚创建的文本视图设置Activity的内容。

现在，让我们看一下如何创建一个相同的应用，不过是在Qt框架上。

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql, povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
#include <QApplication>
#include <QLabel>

int main(int argc, char *argv[])
{
  QApplication app(argc, argv);
  QLabel label("Hello, world!");
  label.show();
  return app.exec();
}
```

它只是打开一个应用并显示标有"Hello World"字符串的一个标签。你可以看到，主Qt类（QApplication）通过Main参数进行初始化。如前所述，Qt应用的入口是Qt主类中的

(c) Copyright Nokia 2010-2013. All rights reserved.

main函数。然后，它创建一个简单的标签并显示其内容。最后，应用会从Qt应用的执行中返回代码。另一个要点就是：这样的代码段也能在Maemo平台上得到编译和执行（当然，图形效果不会相同，因为你使用了具有不同widgets的不同平台）。

下例说明了如何在Android和Qt应用上分别插入一个菜单。

在Android上，菜单是以方法*android.app.Activity#onCreateOptionsMenu(Menu)*定义。而菜单项的回调则以方法*android.app.Activity#onOptionsItemSelected(MenuItem)*定义。

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql, povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
package helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Menu;
import android.widget.MenuItem;

public class HelloAndroid extends Activity {

   private final int MENU_QUIT = 1;

   @Override
   public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       TextView tv = new TextView(this);
       tv.setText("Hello World");
       setContentView(tv);
   }

   /* Creates the menu items */
   public boolean onCreateOptionsMenu(Menu menu) {
       menu.add(0, MENU_QUIT, 0, "Quit");
       return true;
   }

   /* Handles item selections */
   public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case MENU_QUIT:
        quit();
        return true;
    }
    return false;
   }

}
```

在Qt上则会简单一些:)。你只需定义一个QtAction，然后将其插入到应用的菜单栏中。最后，以*connect*方法（用于定义Qt中的任意回调）来定义回调。

```
#include <QApplication>
#include <QAction>

int main(int argc, char *argv[])
{
  QApplication app(argc, argv);
  QAction exitAction = new QAction(tr("&Exit"),this);

  // Add action direct to menubar
  menuBar()->addAction(exitAction);
  connect(exitAction, SIGNAL(triggered()),this, SLOT(close()));
}
```

API映射

文件输入/输出

- 对文件或其它设备的读写。.
  - Android类（出自package java.io）：File, FileReader, FileWriter, BufferedReader, BufferedWriter
  - Qt类：QIODevice, QFile, QTemporaryFile, QBuffer, QProcess, QTcpSocket, QUdpSocket, QDataStream, QTextStream

网络功能

- Socket通信。
  - Android类（出自package **java.net**）：Socket, InetSocketAddress, ServerSocket
  - o Qt类：QTcpSocket, QUdpSocket

Copyright © Nokia 2010-2013. All rights reserved.

- HTTP/FTP 通信。
  - Android类（出自package java.net）：HttpURLConnection, URL
  - Qt类：QNetworkAccessManager, QUrl, QUrlInfo
  - Android类（出自package org.apache.http.*）：HttpClient, HttpGet, HttpPost, HttpResponse
  - Qt类：QHttpHeader, QHttpRequestHeader, QHttpResponseHeader
  - Qt 类：QFtp（Android中无对应类）

媒体

- 对文件或其它设备的读写。
  - Android类（出自package **ndroid.media**）：MediaPlayer, MediaRecorder
  - o Qt类：AudioOutput, MediaController, MediaNode, MediaObject, MediaSource，ObjectDescription, Path, VideoPlayer

Android与Qt的对比

Android被设计成移动电话平台。因此，它能访问各种系统资源，如触摸屏、照相机，及通话功能等。另一方面，Qt是一个跨平台应用和用户界面框架，最初针对桌面环境而设计。而且，Qt还未提供能访问移动电话资源的机制。下表说明了Android和Qt之间的一些重要差异。

- Android提供了对**PIM（名片夹，日历）**的访问，而Qt**尚**未提供；
- Android提供了对**电话通话功能**的访问，而Qt**尚**未提供；
- Android提供了对**信息收发功能**的访问，而Qt**尚**未提供；
- Android提供了对**照相机**的访问，而Qt**尚**未提供。

不过已有Archived:Mobile Extensions补充提供了相关的支持。

- 在Android中，你可以从一个XML说明文件中加载应用的用户界面（如果你使用Glade构建自己的应用用户界面，Maemo也支持该特性）。然而Qt并不支持这种特性；
- Android基于Java，而Qt基于C++；
- Qt是跨平台的。也就是说，应用有机会既在Qt上执行也能在Qt for Maemo上执行，甚至能在windows桌面系统上执行。

参考资料

- Qt
  - Qt 网站
  - Qt 开发伙伴库
  - Qt® 开发基础（图书）
  - o 以Qt 4实现C++图形化用户界面（图书）
  - Qt4手册: 构建Qt应用的艺术（图书）
- Android
  - 维基百科
  - 官方网站
  - 开发者指南
  - Android 应用专业开发（图书）

(c) Copyright Nokia 2010-2013. All rights reserved.

(c) Copyright Nokia 2013 All rights reserved