

基础篇



第 1 章 Android 开发起步

本章将要介绍的是 Android 开发起步的相关知识，首先将会对 Android 平台进行简单的介绍，其中包括 Android 的背景及其应用程序的框架。然后会向读者讲解如何搭建 Android 的开发环境、如何在 Ecl 中创建一个项目，以及如何使用 Android 平台提供的工具调试和监控应用程序。

1.1 Android 平台简介

Android 平台自发布以来就一直吸引着广大开发者的目光，本节就对 Android 平台做简单的介绍，读者通过本节的学习可以对 Android 平台的来龙去脉及内部结构有比较透彻的了解。

1.1.1 Android 背景介绍

Android 是一个开放的手机操作系统平台，为移动设备提供了一个包含了操作系统、中间件及应用程序的软件叠层架构。Android SDK 为开发人员使用 Java 语言编写 Android 平台下的应用程序提供了必要的工具和 API。

1. Android 的诞生

Android 也是一家公司的名称，这家公司的创始人名叫 Andy Rubin。Andy Rubin 原来是苹果的一名工程师，后来创办过几家公司。在创建 Android 公司时，Andy Rubin 准备打造一个移动终端平台并将其对开发人员开放，后来 Google 收购了该公司，于是就有了今天如日中天的 Android。

2. Android 的特性

Android 自推出以来如此受追捧不仅仅是因为网络巨头 Google 的大力扶持，Android 自身所具有的特性才是吸引全球精英开发者的更重要原因。

- 应用程序框架：可以方便地重用和替换手机组件。
- Dalvik 虚拟机：专为移动设备优化过的虚拟机。
- 内部集成浏览器：基于开源的 WebKit 引擎。
- 优化的图形系统：其中自定义了 2D 图形库，3D 图形库基于 OpenGL ES 1.0，可选硬件加速。

- SQLite: 集成了轻量级数据库管理系统。
- 多媒体支持: 支持常见的音频和视频, 以及各种图片格式, 如 MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF 等。
- GSM 技术、蓝牙、EDGE、3G 和 WiFi: 需要硬件支持。
- 摄像头、GPS、罗盘、加速度计: 需要硬件支持。
- 完备的开发环境: 包括设备模拟器、调试工具、内存和性能分析工具, 以及用于 Eclipse 开发环境的插件。

3. Android 的优点

目前市面上的手机操作系统除了 Android, 还有 Symbian、iPhone 等, 与这些手机操作系统相比, Android 具有如下的优点。

(1) 真正开放

Android 平台提供了从底层操作系统到上层的程序界面的所有软件, 使用这个平台无须缴纳任何授权许可费用, 同时不同的厂商还可以根据自身需求修改和扩展 Android 平台。

(2) 应用程序相互平等

在 Android 平台下, 除了应用程序运行的载体虚拟机之外, 其他的软件是完全平等的。例如可以自己开发收发 E-mail 的应用程序来替换系统提供的相应软件。

(3) 应用程序之间沟通无界限

在 Android 平台下开发应用程序, 可以方便地实现应用程序之间的数据共享, 只需要经过简单的声明或操作, 应用程序就可以访问或调用其他应用程序的功能, 或者将自己的部分数据和功能提供给其他应用程序使用。

1.1.2 Android 平台架构

在 1.1.1 节介绍了 Android 平台的诞生及其特性和优点, 本节将会对其内部的系统框架进行分析。了解其框架有助于更好地在 Android 平台上开发应用。Android 平台的应用程序框架如图 1-1 所示。

Android 平台架构中的各模块介绍如下。

1. 应用程序

Android 平台搭载了一系列的核心应用程序, 如 E-mail 客户端、日历、浏览器、电话簿等。这些应用程序都是用 Java 语言开发的。

2. 应用程序框架

通过提供一个开放的开发平台, Android 使开发极其丰富和新颖的应用程序成为可能。开发人员可以自由地使用设备的硬件、访问位置信息、运行后台服务、设置闹钟、添加状态栏的提醒等。

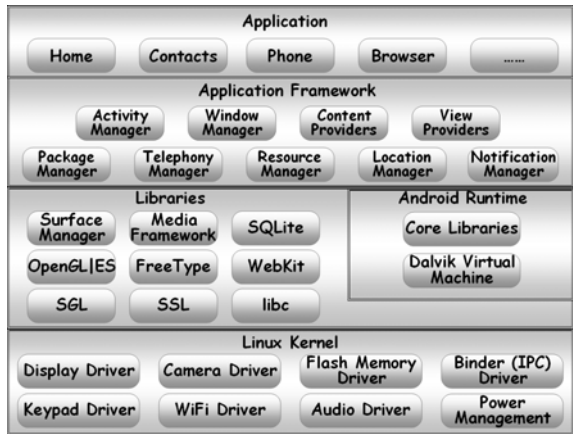


图 1-1 Android 平台的应用程序框架

同时，开发人员可以像核心应用程序那样随意地使用框架的 API 来实现自己的功能，应用程序框架对组件之间的重用做了简化。在 Android 平台下，任何应用程序都可以发布自己的功能模块，其他的应用程序依据一定的安全限制就可以无障碍地访问这些模块。同时这种机制还允许用户随意更换组件。

3. 库

Android 包含一个可供不同组件调用的 C/C++库，这些功能模块通过应用程序框架暴露给开发者。下面列举一些主要的核心库。

- 系统 C 库 (libc)，标准 C 系统库的一个 BSD 系统的衍生实现，并为基于 Linux 的嵌入式设备做了优化。
- 媒体库 (Media Libraries)，基于 PacketVideo 的 OpenCORE，该库支持多种音频和视频格式，以及静态的图片文件。
- SurfaceManager，主要负责管理针对显示系统的访问及无缝地混合来自不同应用程序的 2D 和 3D 图层。
- SGL，2D 图像引擎。
- OpenGL ES，一个基于 OpenGL ES 1.0 API 的实现，该库或者使用 3D 硬件加速，或者使用内置的、经过高度优化过的 3D 软件光栅处理。
- FreeType，提供对点阵字和向量字的绘制支持。
- SQLite，一个面向所有应用程序的轻量级关系型数据库引擎。

4. Android 运行时

在 Android 平台下，每个应用程序都在自己的进程中运行，而且系统为其分配了专有的 Dalvik 虚拟机，Dalvik 虚拟机在设计时就添加了可以在一个设备上运行多个虚拟机的特性。Dalvik 虚拟机执行 .dex 格式的文件，这类文件经过了优化，占有的内存非常小。

与 Java 虚拟机是基于栈的不同，Dalvik 虚拟机是基于寄存器的，Dalvik 虚拟机并不直接执行编译后的 Java 类文件，而是通过内置的 dx 工具将其转换为 “.dex” 格式后再执行。Dalvik 虚拟机依赖于 Linux 核心部件来实现线程管理等功能。

5. Linux Kernel

Android 平台使用 Linux 2.6 版本来构建诸如安全性、内存管理、进程管理、网络协议栈和驱动模型等系统服务，Linux 核心在硬件层和软件层之间建立了一个抽象层，使得 Android 平台的硬件细节对开发人员透明化。

1.2 Android 开发环境的搭建

本书在前面的章节介绍了 Android 平台的相关知识，了解了 Android 的背景知识及平台架构，本节将向读者介绍如何在 Eclipse 中搭建自己的 Android 开发环境，同时还将通过一个“Hello Android”项目向读者演示 Android 平台下应用程序的开发过程。

1.2.1 相关软件的下载与安装

搭建 Android 开发环境的第一步是下载与开发环境有关的软件资源，这些资源主要包括 JDK、Eclipse、Android SDK 及 Android 的支持插件。

1. JDK 的下载与安装

Android 平台下应用程序的开发是采用 Java 语言的，所以首先需要下载并安装 JDK。打开浏览器，在地址栏输入如下地址并访问：<http://java.sun.com/javase/downloads/index.jsp>，打开后的页面如图 1-2 所示。



图 1-2 JDK 下载界面 1

单击如图 1-2 所示的“Download JDK”按钮即可进入下载界面，选择操作系统平台之后，就可以下载 JDK 了，目前最新的 JDK 版本为 JDK 6 Update 18。

双击并运行下载好的 JDK 安装文件，按照提示将 JDK 安装到指定的文件夹，本书中将其安装到了 C:\Program Files\Java\jdk1.6.0_18 目录下。

2. Eclipse 的下载与安装

打开浏览器，在地址栏输入 <http://www.eclipse.org/downloads/>，在下载界面找到 Eclipse Classic 3.5.2，单击下载链接，将 Eclipse 下载到电脑上。Eclipse 不需要安装，只需将下载到电脑上的压缩包解压到硬盘上某个目录即可，本节中将其解压到 F:\eclipse 目录下。

3. Android SDK 的下载与安装

① 打开浏览器，在地址栏输入 <http://developer.android.com/sdk/index.html>，打开 Android SDK 的下载页面，将 android-sdk_r05-windows.zip 下载到电脑上，并将其解压到硬盘上某个位置，本书中为 F:\android-sdk-windows。

② 运行 F:\android-sdk-windows 目录下的 SDK Setup.exe，程序将自动检测是否有更新的 SDK 版本可供下载，检查结果如图 1-3 所示。



图 1-3 运行 SDK Setup 示意图

③ 单击如图 1-3 所示的“Accept All”单选按钮，并单击“Install”按钮安装最新版本的 SDK。单击“Install”按钮后如图 1-4 所示。最新的 SDK 文件将会下载到 F:\android-sdk-windows 目录中。

④ 在桌面上右键单击“我的电脑”，依次选择“高级”-“环境变量”-“系统变量”，在系统变量 Path 中添加“F:\android-sdk-windows\tools”并单击“确定”按钮，如图 1-5 所示。



图 1-4 Android SDK 下载示意图



图 1-5 将 tools 路径添加到系统变量 Path 中

4. ADT 的下载与安装

下面介绍在 Eclipse 环境下进行 Android 开发所必需的插件 ADT（Android Development Tools）的安装，主要通过如下几个步骤来完成。

① 运行 F:\eclipse 目录下的 eclipse.exe，单击“Help-Install New Software...”，弹出如图 1-6 所示的界面。

② 单击“Add...”按钮，弹出如图 1-7 所示的添加新站点的界面，在“Name”文本框中输入站点名称（本书为 ADT，读者可自行取名），在“Location”文本框中输入 <http://dl-ssl.google.com/android/eclipse/>，输入完成后单击“OK”按钮。

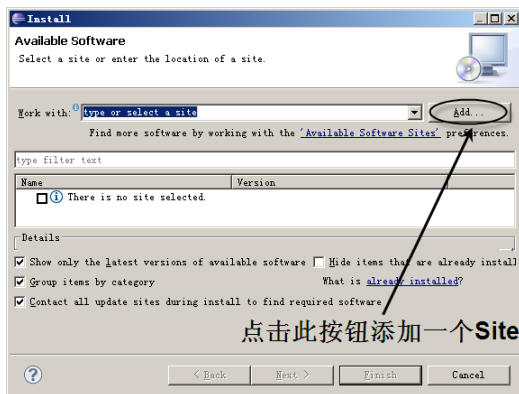


图 1-6 Install New Software 界面

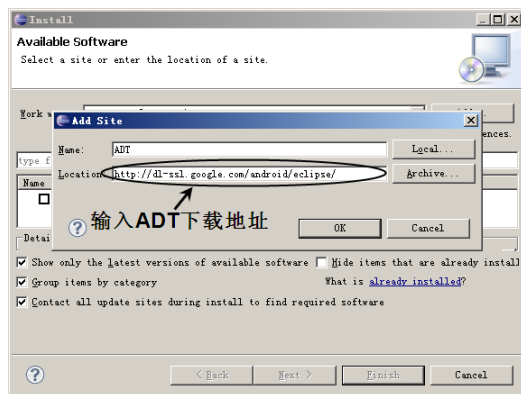


图 1-7 添加 Site 界面

③ Eclipse 会自动连接该站点，并将连接结果显示在如图 1-8 所示的列表中。单击“Next”按钮，将弹出显示所要安装的插件的细节内容的界面，如图 1-9 所示。

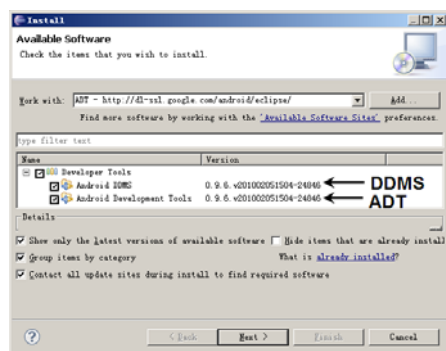


图 1-8 获取 ADT 信息示意图



图 1-9 显示所要安装的插件细节的界面

④ 继续单击“Next”按钮，将弹出如图 1-10 所示的确认安装插件界面，选中“I accept the terms of the license agreements”单选按钮并单击“Finish”按钮，Eclipse 将会进行 ADT 和 DDMS 插件的安装，安装进度如图 1-11 所示。

⑤ 在安装的过程中会弹出如图 1-12 所示的插件中包含未注册内容的警告，单击“OK”按钮继续安装。安装结束后会弹出如图 1-13 所示的是否重启 Eclipse 的提示框，单击“Yes”重启 Eclipse。

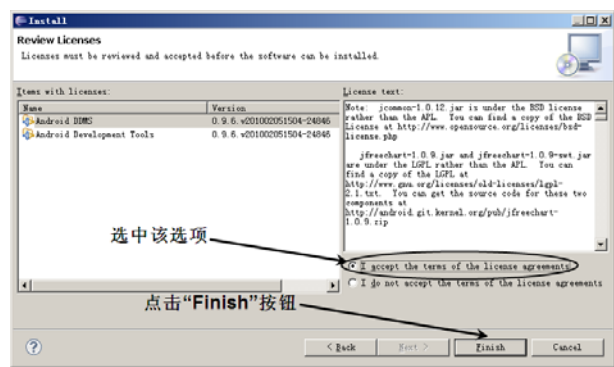


图 1-10 确认安装插件界面

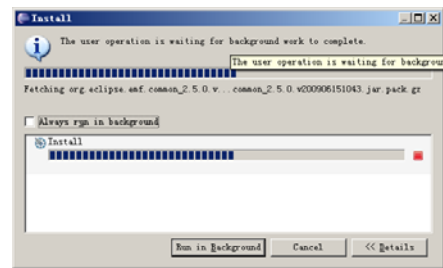


图 1-11 正在安装插件界面

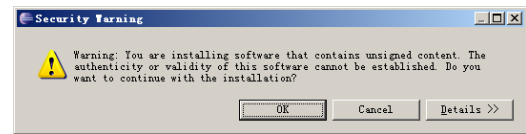


图 1-12 弹出未注册警告提示框

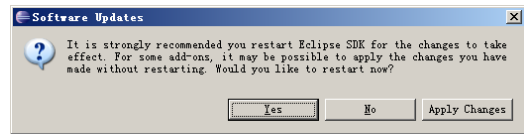


图 1-13 弹出重启 Eclipse 提示框

⑥ Eclipse 重启之后，选择“Window” - “Preferences”菜单选项，如图 1-14 所示。

在 Preferences 菜单左侧选择“Android”，Eclipse 会提示还没有指定 Android SDK 的路径，在如图 1-15 所示的界面中输入 Android SDK 的路径，确认后完成 ADT 插件的安装。

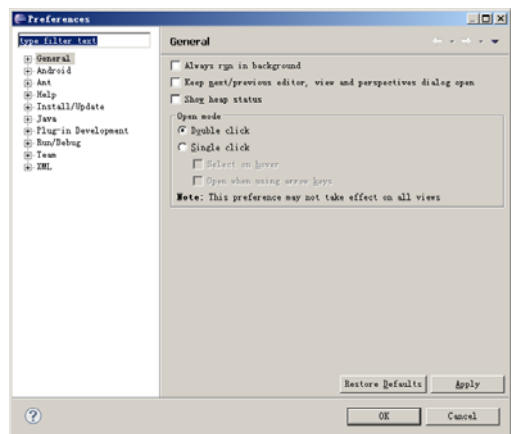


图 1-14 Preferences 菜单界面

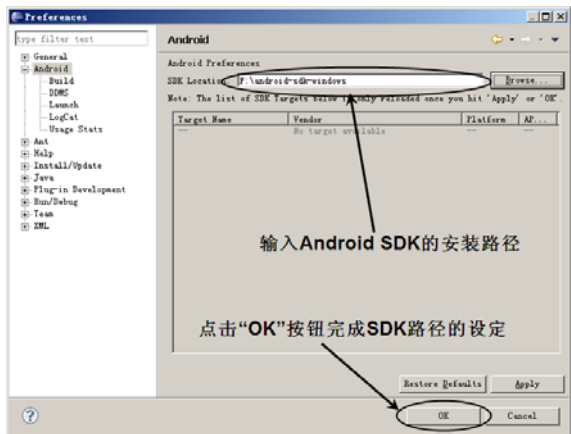


图 1-15 配置 Android SDK 的路径

1.2.2 虚拟设备的创建和使用

使用 Andoid SDK 开发应用程序需要进行测试，Android 为开发人员提供了可以在电脑上直接测试应用程序的虚拟设备 AVD（Android Virtual Device），或称做模拟器。使用 AVD 可以方便地对应用程序进行测试。本节介绍如何在 Android 平台下创建 AVD。

1. Eclipse 环境下 AVD 的创建

首先介绍如何在 Eclipse 环境下创建 AVD，步骤如下。

① 启动 Eclipse，依次选择“Window” - “Android SDK and AVD Manager”，如图 1-16 所示。

② 单击如图 1-16 中所示的“New...”按钮，弹出如图 1-17 所示的对话框。

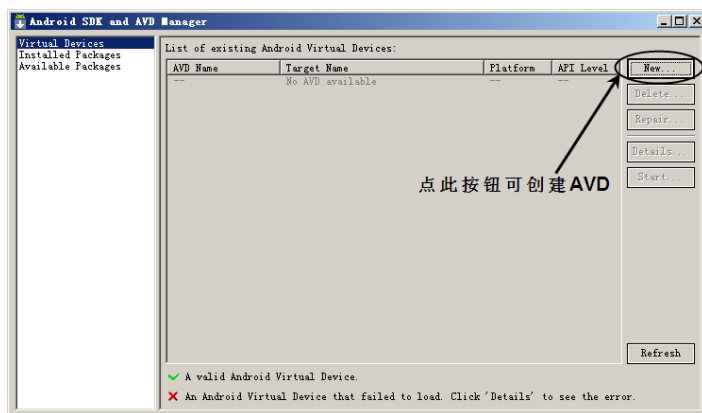


图 1-16 Android SDK and AVD Manager 界面



图 1-17 创建 AVD 对话框

③ 在图 1-17 所示的对话框中设置所要创建的 AVD 的名称、API 版本、SD 卡的大小及 AVD 的皮肤，单击“Create AVD”按钮完成 AVD 的创建。创建成功的 AVD 将会显示在如图 1-18 所示的列表中。

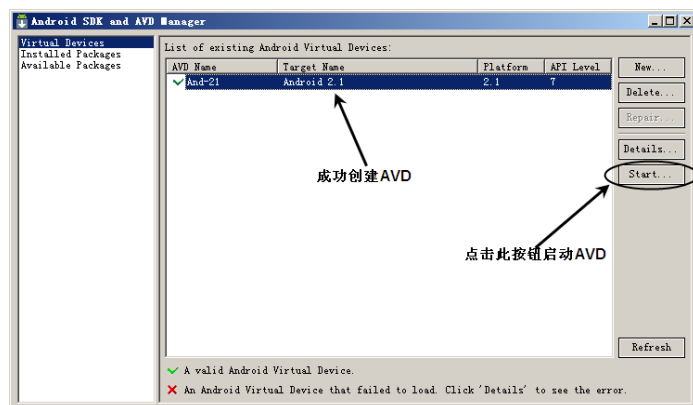


图 1-18 成功创建 AVD

④ 在如图 1-18 所示界面中单击“Start...”按钮，将会启动所选的 AVD，本例中将启动名为“And-21”的 AVD，启动后的虚拟设备如图 1-19 所示。

2. 命令行下 AVD 的创建

下面介绍如何在命令行下创建 AVD 虚拟设备，步骤如下。

① 启动命令行窗口，在其中输入“android list target”，屏幕中将显示当前可用的设备列表，如图 1-20 所示。

② 在命令行输入“android create avd --name And-21-2 --target 11”，结果如图 1-21 所示，直接按回车键将完成 AVD 的创建。创建成功时显示的界面如图 1-22 所示。



图 1-19 启动后的虚拟设备

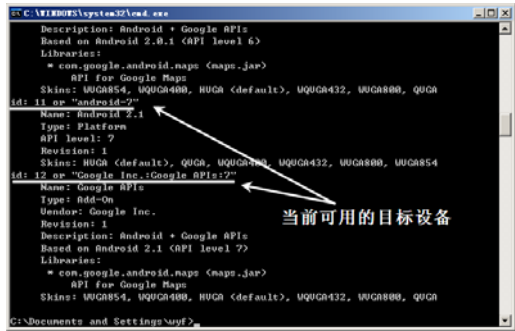


图 1-20 当前可用的设备列表

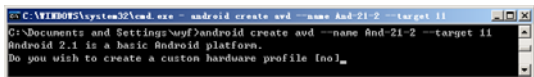


图 1-21 创建 AVD 界面



图 1-22 成功创建 AVD 界面

此时可以通过在命令行输入“`emulator -avd And-21-2`”来启动刚刚创建好的以“`And-21-2`”为名称的 AVD。同时，在命令行下创建的 AVD 也可以在 Eclipse 的“Android SDK and AVD Manager”中看到，同时可以在 Eclipse 中启动它。

1.2.3 创建“Hello Android”

前面的章节已完成了 Android 开发环境的初步搭建及虚拟设备 AVD 的创建，本节将向读者介绍如何在 Eclipse 环境下开发第一个 Android 应用程序——Hello Android。开发步骤如下。

① 启动 Eclipse，按照如图 1-23 所示依次选择“File” - “New” - “Android Project”，将弹出如图 1-24 所示的创建新项目界面。

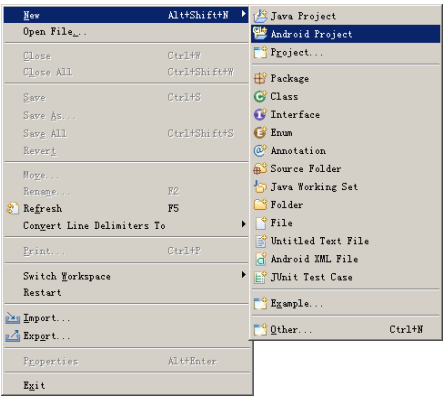


图 1-23 创建 Android 项目菜单

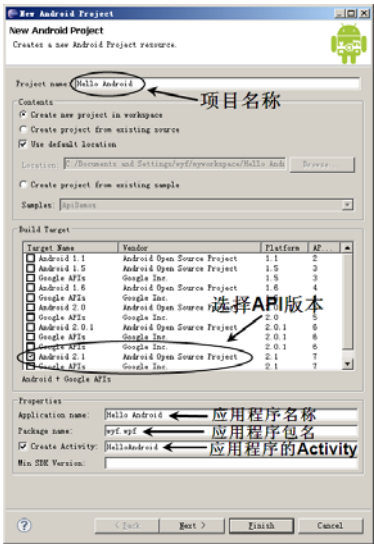


图 1-24 创建 Android 项目界面



图 1-25 Hello Android 项目的目录结构

在如图 1-24 所示的创建 Android 项目界面填入必要的信息，如项目名称、应用程序名称、Activity 名称等，同时还需要为项目选择目标 API 版本，填写好相关信息后单击“Finish”按钮，将创建一个项目。该项目的目录结构如图 1-25 所示。

在 Hello Android 目录中，src 目录用于存放源代码，gen 目录用于存放系统自动生成的类，res 和 assets 目录中存放了程序中的非代码资源。

提示：如果读者在“File”-“New”中未找到“Android Project”菜单选项，请选择“Other...”并在其中寻找“Android Project”。

② 启动虚拟设备，在 Hello Android 项目上单击右键，按照如图 1-26 所示从菜单中选择“Run As”-“Android Application”，这时如果没有已启动的虚拟设备，系统将会自动启动一个虚拟设备，并将应用程序在此设备中运行。

③ 运行项目后，观察虚拟设备的屏幕，显示如图 1-27 所示的界面。

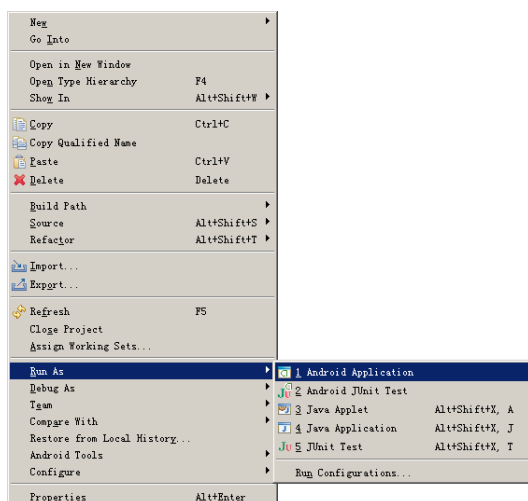


图 1-26 运行项目菜单选项

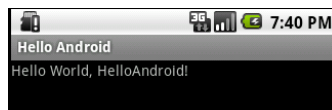


图 1-27 Hello Android 运行示意图

1.2.4 Android 应用程序的调试

在 Eclipse 开发环境中，不仅可以通过 ADT 插件来进行 Android 应用程序的开发，还可以非常方便地对应用程序进行测试。本节介绍如何在 Eclipse 开发环境中对 Android 应用程序进行测试。最主要的测试工具是 DDMS。

DDMS 是 Dalvik Debug Monitor Service 的简称，主要功能是监控应用程序的运行并打印日志、模拟电话打入与接听、模拟短信的收发、虚拟地理位置等。有了 DDMS 的帮助，可以大大降低应用程序的测试成本。

1. DDMS 的启动

启动 Eclipse 后，单击 Eclipse 右上角的“Open Perspective”，从下拉菜单中选择“DDMS”，

如图 1-28 所示。如果下拉菜单中没有 DDMS，则单击下拉菜单中的“Other...”，并从弹出的对话框中选择“DDMS”，如图 1-29 所示。

提示：除了在 Eclipse 中启动 DDMS，也可以通过运行 SDK 安装目录下的 tools 文件夹下的 ddms.bat 来启动 DDMS。



图 1-28 在“Open Perspective”中直接添加 DDMS 图 1-29 在“Open Perspective”的弹出菜单中添加 DDMS

2. Devices 面板

首先来看 DDMS 系统中的 Devices 面板，如图 1-30 所示。在 Devices 面板中，可以看到与 DDMS 连接的设备终端的信息及设备终端上运行的应用程序。

在 Devices 面板中，可以设置应用程序更新 Heap 状态、更新 Thread 状态，或者直接停止某个应用程序的执行。同时 Devices 面板中还可以截取手机屏幕。

3. Emulator Control 面板

DDMS 中的 Emulator Control 面板如图 1-31 所示，在 Emulator 面板中可以向模拟器 AVD 中打入电话或发送短信，还可以虚拟模拟器的位置信息。

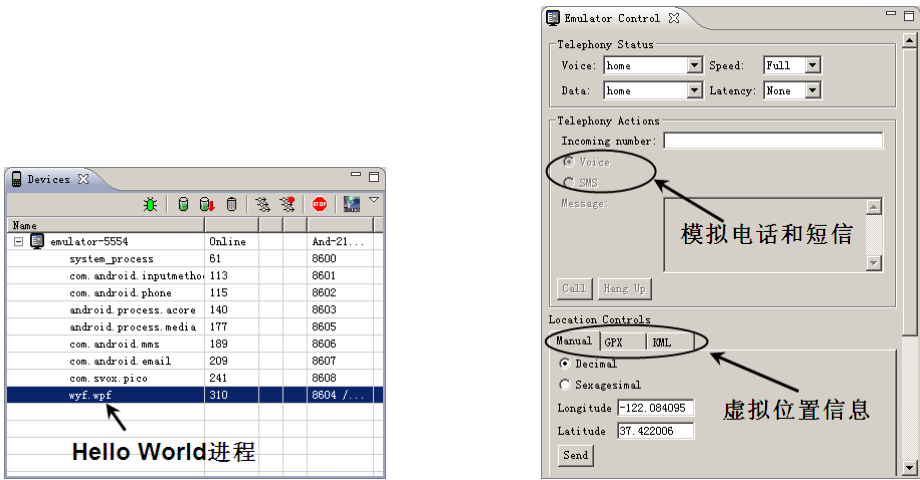


图 1-30 Devices 面板示意图

图 1-31 Emulator Control 面板

4. 信息输出面板

如图 1-32 所示是 DDMS 中的信息输出面板，信息输出面板在调试中起到的作用最大，其显示的信息窗口主要包括如下几个。

- Thread 更新信息，要使该窗口显示内容，需要在 Devices 面板中设置 Thread 更新。Thread 更新信息窗口的主要作用是显示应用程序当前状态下所有正在执行的线程的状态。

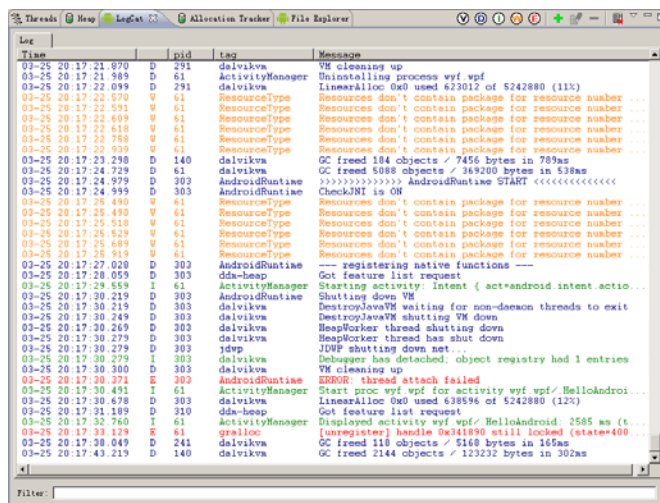


图 1-32 DDMS 的信息输出窗口

- Heap 更新信息，要使该窗口显示内容，需要在 Devices 面板中设置 Heap 更新。Heap 更新信息窗口主要的作用是显示应用程序中堆的分配与回收等。
- File Explorer，该信息窗口主要显示 Android 模拟器中的文件，如果模拟器启动时加载了 SD 卡，也可以在该窗口查看 SD 卡的信息，同时还可以向 SD 卡中拷入或拷出文件。
- LogCat，该信息窗口在 Android 应用程序的调试中是使用最多的。LogCat 中会显示应用程序的运行信息，包括调试信息、警告信息、错误信息、普通信息及冗余信息。不同类型的信息具有不同的显示颜色，方便开发人员观察。

5. DDMS 中的日志输出

在 DDMS 中的 LogCat 可以输出应用程序的运行信息，如果开发人员需要在程序运行时打印一些调试用的消息，除了使用传统的“System.out.println()”之外，还可以使用 android.util 包下的 Log 类，该类可以将信息以日志的形式输出到 LogCat 中。

Log 类中主要用到的方法及说明如表 1-1 所示。

表 1-1 Log 类中的方法及说明

方法名	方法说明	参数说明
Log.v(String tag, String msg)	输出冗余消息	tag: 日志标签，可用于过滤日志信息 msg: 输出的日志消息
Log.d(String tag, String msg)	输出调试消息	
Log.i(String tag, String msg)	输出普通消息	
Log.w(String tag, String msg)	输出警告消息	
Log.e(String tag, String msg)	输出错误消息	

下面通过一个例子来说明 Log 类的用法，按照 1.2.3 节的步骤创建一个项目 HelloAndroid2，打开其项目中 HelloAndroid2.Java，将已有的代码替换为如下代码。

```
1 package wyf.wpf; //声明包语句
2 import android.app.Activity; //引入相关类
3 import android.os.Bundle; //引入相关类
4 import android.util.Log; //引入相关类
5 public class HelloAndroid2 extends Activity {
6     String TAG = "MyLog"; //定义日志标签
7     @Override
8     public void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.main);
11        Log.v(TAG, "This is a VERBOSE message"); //输出冗余消息
12        Log.d(TAG, "This is a DEBUG message"); //输出调试消息
13        Log.i(TAG, "This is an INFO message"); //输出普通消息
14        Log.w(TAG, "This is a WARNING message"); //输出警告消息
15        Log.e(TAG, "This is a ERROR message"); //输出错误消息
16    }
17 }
```

代码位置：见随书光盘中源代码/第 1 章/Hello Android2/src/wyf/wpf 目录下的 HelloAndroid2.Java。

运行 HelloAndroid2 项目，将视图切换到 DDMS，观察 LogCat 的输出，如图 1-33 所示。

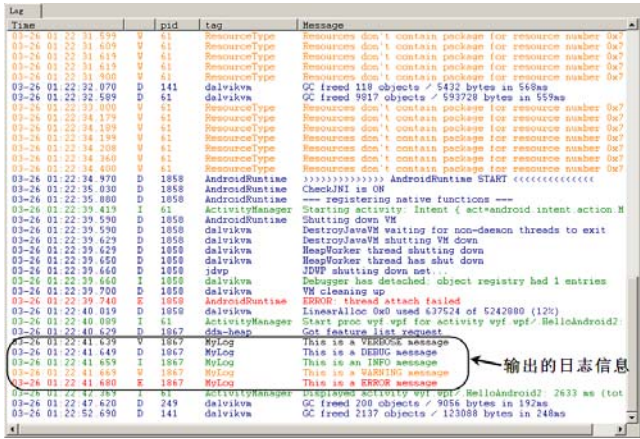


图 1-33 LogCat 的日志输出信息

当 LogCat 输出的信息量很大需要对其内容进行过滤时，可以在 DDMS 中根据标签进行过滤，这样使得应用程序调试变得更加方便。

1.3 小结

本章作为 Andoid 开发的起步章节，主要向读者介绍了 Android 平台的来历和背景，以及 Android 平台的内部架构，了解这些知识有助于读者以后更深入地学习 Android 开发。在本章的后半部分主要介绍了如何搭建 Android 的开发环境。相信读者在学习本章之后，已经熟练掌握了在 Eclipse 环境下开发 Android 应用程序的步骤。