移动与嵌入式开发技术

Professional Android Programming with Mono for Android and .NET/C#

C#开发Android应用实战

——使用Mo[redacted]d和.NET/C#

Xamarin公司首席技术官Miguel de Icaza推荐作序

Wallace B. McClure
Nathan Blevins
[美] John J. Croft IV 著
Jonathan Dick
Chris Hardy
王净 范园芳 田洪 译

清华大学出版社

移动与嵌入式开发技术

C#开发 Android 应用实战

——使用 Mono for Android 和.NET/C#

Wallace B. McClure
Nathan Blevins
[美] John J. Croft IV 著
Jonathan Dick
Chris Hardy
王净 范园芳 译
田洪

清华大学出版社

北 京

Wallace B. McClure, Nathan Blevins, John J. Croft IV, Jonathan Dick , Chris Hardy
Professional Android™ Programming with Mono for Android and .NET/C#
EISBN：978-1-118-02643-4

北京市版权局著作权合同登记号 图字：01-2012-5650

图书在版编目(CIP)数据

C#开发 Android 应用实战——使用 Mono for Android 和.NET/C#/ (美) 麦克卢尔(McClure, W. B.) 等著；王净，范园芳，田洪 译. —北京：清华大学出版社，2013.1
(移动与嵌入式开发技术)
书名原文：Professional Android™ Programming with Mono for Android and .NET/C#
ISBN 978-7-302-30499-9

Ⅰ. ①C… Ⅱ. ①麦… ②王… ③范… ④田… Ⅲ. ①C 语言—程序设计 ②移动终端—应用程序—程序设计 Ⅳ. ①TP312 ②TN929.53

中国版本图书馆 CIP 数据核字(2012)第 257028 号

责任编辑：王　军　韩宏志
装帧设计：牛艳敏
责任校对：成凤进
责任印制：沈　露

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
　　社 总 机：010-62770175　　**邮　　购**：010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：31.5　　**字　　数**：767 千字
版　　次：2013 年 1 月第 1 版　　**印　　次**：2013 年 1 月第1 次印刷
印　　数：1～3000
定　　价：59.80 元

产品编号：042656-01

目　录

第 1 章

Android、移动设备和 Marketplace 简介

本章主要内容：

- Mono 的发展简史及其与.NET Framework 的关系
- Mono for Android 如何为.NET 开发人员开启 Android 平台
- 为什么 Mono for Android 对开发人员具有如此大的吸引力
- Android 及其成长历史
- 探讨跨平台的替代方案

在过去几年中，智能手机的使用以惊人的速度增长。最近，USA Today 就智能手机如何成为人们日常生活中不可缺少的一部分进行了报道。与桌面计算机不同的是，智能手机的增长和普及也带来了竞争，目前还没有任何一个厂家或者平台统治了移动设备市场；移动设备可以使用 Symbian、Research in Motion(黑莓)、Windows Mobile、Android 以及其他平台。此外，设备也可以运行相同的操作系统，并以不同的外观样式呈现给用户。市场上所存在的这种裂痕使开发人员面临着一个非常大的问题：他们如何使用已有的某种开发框架或工具，以及如何使用关于某种设备(该设备拥有较大且不断增长的市场份额)的相关知识？

本章介绍了.NET/C#开发人员如何在拥有最高平台知名度的智能手机(Android)上完成开发工作。同时还介绍了该种智能手机的市场份额为何比其他设备增长得更快。

1.1 产品比较

本节将简要介绍.NET Framework、Mono 以及 Mono for Android。这些产品允许数量最

为庞大的 Android 开发人员针对 Android 移动设备家族(目前市场上增长速度最快的移动平台)进行开发。

1.1.1 .NET Framework

在过去十年中，.NET Framework 的普及率不断提高。从 20 世纪 90 年代末期 Microsoft 就开始.NET Framework 方面的工作，并于 2002 年发布了.NET Framework 的第一个版本。最近，Microsoft 发布.NET Framework 4。该.NET Framework 有多种版本，包括 32 位版本、64 位版本、针对 Xbox 游戏平台的版本以及针对 Microsoft 移动设备的版本(名为 CF(Compact Framework，精简框架))。在开始浏览 Mono 框架之前，需要注意一些有关.NET Framework 的关键事实：

- Microsoft 发布了该框架的开发工具 Visual Studio .NET。该工具是针对.NET 的集成开发环境。
- 该框架基于一种用来执行针对该框架所编写的软件的虚拟机。这种虚拟机环境称为 CLR(Common Language Runtime，公共语言运行时)，主要负责安全、内存管理、程序运行以及异常处理。
- 在.NET Framework 中编写的应用程序的源代码(比如 Visual Basic 或者 C#)最初被编译为一种名为 MSIL 的中间语言。该初始编译由特定于语言的命令行编译器(Visual Studio 或者其他生成工具)来执行。而当执行应用程序时通常会执行二次编译。二次编译将会获取中间语言并将其编译为可在操作系统中运行的可执行代码。二次编译被称为 JIT(just-in-time，即时)编译。
- 该框架是独立于语言的，并可以使用多种语言。在 Visual Studio 中，Microsoft 附带了多种语言，包括 Visual Basic、F#、C++和 C#。
- 该框架包含了为不同语言提供一致功能的一系列库。这些库称为基类库。
- Microsoft 向不同的标准组织提交了.NET Framework 的不同部分，其中包括针对 C#语言、CLI(Common Language Infrastructure，公共语言基础结构)、CTS(Common Type System，公共类型系统)、CLS(Common Language Specification，公共语言规范)和 VES(Virtual Execution System，虚拟执行环境)的部分。
- 该框架拥有数量最庞大的开发人员。因此相比其他的开发框架，有更多的开发人员熟悉.NET Framework。
- .NET Framework 的其中一个缺点是非 Mcirosoft 平台无法使用该框架。

所有这一切的意义在于 Microsoft 为.NET Framework 创建了一个基于标准的环境。虽然大多数使用 Microsoft 平台的开发人员并不必担心.NET Framework 的标准相容性，但.NET Framework 在这方面的意义却不可以低估。通过定义并向遵循委员会提交这些标准，Microsoft 让许多低级别的开发人员可以融入到.NET Framework。在该环境中，Miguel de Icaza 充分发挥了想象力并创建了接下来将要讨论的 Mono。

1.1.2　Mono

Mono 是一个在非 Windows 操作系统中提供 C#编译器和 CLR 的开源项目。目前，Mono 授权于 GPL 版本 2、LGPL 版本 2、MIT 以及双许可证。可在 Mac、Linux、BSD 以及其他操作系统中运行 Mono。通过 C#编译器，还可在 Mono 中运行其他语言，其中包括 F#、Java、Scala 和 Basic 等。

Mono 的创始人是 Miguel de Icaza，于 2001 被正式宣布，并在 2004 年发布了第一个版本，Mono 目前的版本为 2.10，同时 Mono 也在不断地持续更新，可能在你阅读本章的时候已经有了 Mono 的最新版本。目前，Mono 具有许多与.NET 4 中功能相同的功能。Mono 一直以来由 de Icaza 直接领导。最近，Mono 的管理已经移交给 Xamarin 公司，由其指引 Mono 的发展方向。Mono 起始于 C#编译器的开源实现，并由该最初设计发展为目前.NET 的开源实现。现在，Xamarin 的职责是发展 Mono。同时负责开发 Mono for Android、MonoTouch 以及让开发人员使用这些产品所需的软件。Xamarin 在移动领域重点关注 Mono，我认为这些产品将是非常优秀的。

虽然期望 Mono 的功能可以尽可能多地与.NET Framework 的功能相匹配，但这是不可能的，因为 Microsoft 拥有更多资源，并且在这些功能的开发上具有先起步的优势。但与此同时，Mono 项目还是拥有许多与.NET Framework 功能相同的功能。Xamarin 最多只能实现.NET Framework 中大多数 API，但稍滞后一些。

随同 Mono 一起的是名为 MonoDevelop 的开源 IDE，该 IDE 作为 SharpDevelop IDE 的一个端口一起启动。MonoDevelop 最初是一个允许在 Linux 上进行 Mono 开发的项目，但随着 MonoDevelop2.2 的发布，它也具备了在 Mac、Windows 以及其他非 Linux UNIX 平台上用 Mono 进行开发的能力。

虽然.NET Framework 目前非常流行，但是由于存在下面两个问题，使其不适合于在 Android 上运行：

- 从一定程度上讲 Google 和 Microsoft 是竞争对手，所以它们不可能愿意一起协同工作。Microsoft 开发 Windows Mobile 设备有很多年了，而该操作系统直接与 Google 的 Android 操作系统竞争。
- 从根本上讲，.NET Framework 是 Java VM(Virtual Machine，虚拟机)的主要竞争对手，而后者是 Android 设备的核心。该 Java VM 称为 Dalvik。自.NET Framework 最初宣布开始，它与 Java 就一直是竞争对手。

.NET/Mono 和 Android 存在的其中一个缺点是.NET/Mono 开发人员不能够在 Android 平台上利用他们已有的.NET/Mono/C#相关知识。图 1-1 显示了这个概念。.NET/Mono 开发人员不能够针对 Android 进行开发，因为它们是两个完全独立的实体。

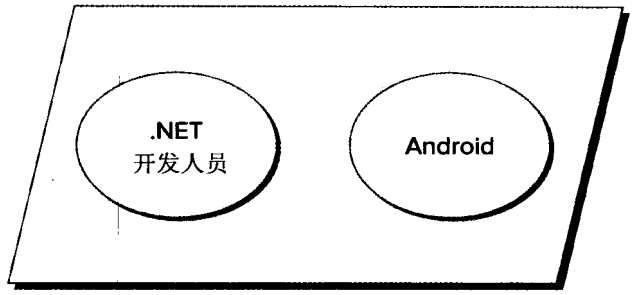

图 1-1

2009 年，Mono 团队发布了 MonoTouch——Mono for Android 的前身。MonoTouch 允许那些熟悉 C#的开发人员针对 Apple 的 iPhone 进行开发。基于构建 MonoTouch 的经验，Mono 团队掌握了如何有效地在设备的本机应用程序编程接口(API)上构建一个 C#/Mono 层。

1.1.3 Mono for Android

2010 年 4 月，Apple 对其软件开发工具包(SDK)授权进行改变，这给移动开发市场带来了恐惧、不确定性和疑惑。这一变化使许多开发人员对 iPhone 和 iOS 开发产生了疑问。而与此同时，Mono 团队正在尝试为 Android 创建一个类似于 MonoTouch 的 Mono 产品。由于 Apple SDK 的变化，Mono 团队宣布了 Mono for Android 产品，并为其投入了大量资源。2011 年春季 Mono 团队正式发布 Mono for Android。虽然 Apple 最终解决了其 SDK 问题，但是 Mono 团队却在这 5 个月的时间里投入大量的资源来开发 Mono for Android。其结果是，如果不是 Apple 在 2010 年几个月里将 MonoTouch 逼入绝境，Mono for Android 也不会得到如此大的发展。

Mono for Android 允许.NET 开发人员创建可在 Android 上运行的本机应用程序。这些应用程序看似在 Dalvik 上运行的本机 Java 应用程序。通过使用 Mono for Android，可将应用程序编译为可在 Android 设备上运行的可执行代码。其意义不应该被低估：如图 1-2 所示，.NET/Mono 开发人员通过使用 Mono for Android 可以针对 Android 进行开发。

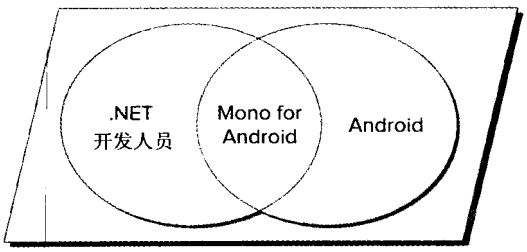

图 1-2

那么 Mono for Android 是如何完成这个过程的呢？它允许对 Windows Froms 应用程序进行转换或者再编译并将其部署到 Android 上吗？Mono for Android 在 Android OS 的本机编程层之上提供了一个.NET 层。而针对 Dalvik 进行开发的开发人员则使用 Java 编写应用

程序。虽然 Mono for Android 没有提供跨 Windows Forms 应用程序进行编译的机制，但它允许开发人员构建在 Android 上运行的应用程序。

总之，Mono for Android 所公开的 API 是.NET 4 Framework 核心功能、Silverlight API 以及本机 Dalvik Java VM 的组合。Mono for Android 在 Android 的本机 API 与.NET 以及 C#开发人员所熟悉的 API 之间提供了一个桥梁(互操作)层。

1. Mono for Android 组件

Mono for Android 由一组针对移动平台进行优化的程序集、命名空间以及类组成。其代码是.NET 4、Silverlight 和 Windows Phone 配置文件的组合，同时还包括允许开发人员充分利用 Android 平台的代码。

命名空间和类

Mono for Android 提供了一组丰富的命名空间和类来支持为 Android 设备构建应用程序。下面列出了一些 Mono for Android 所提供的最流行的程序集和功能：

- Mono.Android.dll：该程序集提供了与 Android API 的 C#绑定。其中包括了支持 Android.*命名空间的命名空间。
- System.dll：该程序集为 Mono for Android 提供了许多.NET Framework 功能。
- Mono.data.Sqlite.dll：该程序集是针对本机 SQLite 数据库的 ADO.NET 提供程序。
- Mono.Data.Tds.dll：该程序集提供了对 TDS 协议的支持，而该协议主要用于连接 SQL Server。
- OpenTK.dll：该程序集提供了对 OpenGL 的支持。
- System.Json.dll：该程序集提供了对使用 JSON 的支持。
- System.ServiceModel.dll：该程序集提供了对 WCF 的支持。
- System.Xml.dll：该程序集提供了对 XML 的支持。
- System.Xml.Linq.dll：该程序集提供了从 LINQ 到 XML 的支持。

在这些程序集中，Mono for Android 还提供了一些非常重要的命名空间。它们是：

- Android：Android.*命名空间提供了对资源、类以及应用程序权限的支持。
- Android.Bluetooth：该命名空间提供了对蓝牙的支持。
- Android.Database：该命名空间提供了对设备上 SQLite 数据库的支持。
- Android.Graphics：该命名空间提供了对图形显示的支持。
- Android.Hardware：该命名空间提供了对 Android 设备上硬件(比如照相机)的支持。
- Android.Location：该命名空间提供了对位置的必要支持。
- Android.Net：该命名空间提供了对网络的支持，其中包括对 VoIP(Voice over IP)和 WiFi 的支持。

这些命名空间只是 Mono for Android 中可用命名空间中很小的一部分，它们的功能不言自明。此外，这些命名空间是特定于 Android 的。使用它们所编写的代码只能在基于 Android 的设备上运行。

1.1.4 开发工具

不管构建哪种类型的项目，开发工具都是创建应用程序的一个重要组成部分。那种只能通过一堆文件、基于字符的编辑器、用于调试的命令行输出和 Makefile 文件来构建应用程序的日子已经一去不复返了。

那些经常使用.NET Framework 的开发人员可能对 Visual Studio 非常熟悉。Visual Studio 是 Microsoft 的开发工具。它提供了对解决方案、项目、可视化设计表面(surface)、数据库以及其他许多功能的支持。

同样，Mono 也有其自己的开发工具；MonoDevelop 是一款免费的、通过 Mono 进行开发的 IDE，最早它是 SharpDevelop IDE 的一个分支。起初 MonoDevelop 只能在 Linux 上运行，但随着 2.2 版本的发布，它开始可以在 Mac 和 Windows 上运行。通过使用 MonoDevelop，不仅可以创建和管理许多项目，还可以调试代码并将其部署到模拟器和设备中以便进行测试。

幸运的是，Mono 团队开发出了 Mono for Android，它可以跨 Visual Studio 和 MonoDevelop 进行工作，同时还可以作为除 Windows 外的其他操作系统的插件。这样就便于跨 Visual Studio、Mac 上的 MonoDevelop 以及 Windows 上的 MonoDevelop 利用 Mono for Android 编写代码。开发人员可以自由地使用他们所偏爱的任何一种开发 IDE。我个人发现 Windows 和 Mac 目前都有其各自的优点，其中包括：

- Windows 上的调试可能是大多数开始使用 Mono for Android 的开发人员开始的地方。
- Mac 上的调试似乎可在 Android 模拟器上更好地工作。

1.2 移动开发

当开发人员在 Android 上利用 Mono for Android 构建应用程序的时候，需要牢记几个要点：

- Android 模拟器适合于进行初始测试；然而，它并不是对于所有测试都是准确的。因为在模拟器中运行的功能并不一定在所有 Android 设备中以相同的方式运行。应该在不同版本的 Android 设备中完成最终测试。

提示：根据截至撰写本书时的 Android SDK 所述，在物理设备上对高级功能进行测试通常更准确。但对于基本的开发，模拟器则更便于使用。而由于快照，模拟器使用起来通常更加快捷。

- .NET 的可执行文件都非常小，因为这些文件可以使用框架的共享副本。Mono for Android 可按照两种不同的方法来部署应用程序。其中最常用的方法是将应用程序和 Mono for Android 绑定到一起。而第二种方法是让应用程序共享 Mono 框架。这

样就可以尽量缩小应用程序的可执行文件，但同时这也意味着必须在设备中安装 Mono for Android 的 Mono 框架的副本。

提示：在撰写本书的时候建议的做法是将应用程序与 Mono for Android 运行时相绑定。通常当对应用程序进行“Release”生成时需要这么做。

- 成为设备上的一名好公民是非常重要的。开发人员需要不断地思考如何实现可成为良好公民的功能。

1.2.1　解决支持问题

虽然 Mono for Android 是一款商业许可产品，但目前仍处于持续的开发中，所以它可能并不支持某些特定的命名空间或程序集。这种情况下，你有两种选择：

(1) 等待 Mono for Android 产品完成对该程序集的实现。

(2) 在项目中添加必要的代码或者对必要程序集的引用。当应用程序需要使用 System.Web.*命名空间中的代码时这种方法是相当常见的。例如，假如一个应用程序需要调用基于 REST 的 Web 服务，并且在发送数据前对其进行编码。此时应该调用 System.Web.HttpUtility.HtmlEncode()。但在默认情况下 System.Web 命名空间并不是 Mono for Android 的一部分。此时必须在应用程序中引用 System.Web 程序集来添加该命名空间。

1.2.2　设计问题

除了为 Android 构建应用程序所涉及的技术问题外，开发人员还需要了解一些设计问题：

- **不要先为一个桌面环境设计一个应用程序，然后再将其缩减到 Android 或任何移动设备中。**Android 并没有桌面计算机所拥有的显示器、硬件或者存储器。所以，Android 和移动设备应用程序最好具有简单且专用的功能，但不要期望这些应用程序能够完成桌面应用程序可以完成的所有工作。
- **Android 模拟器是一个非常好的工具，但不要将测试限制在此工具上。**模拟器仅是个模拟器。键盘和鼠标与 Android 模拟器是结合在一起的，因为模拟器主要运行在桌面计算机上。同时还需要了解的是模拟器最终使用的是开发系统的 CPU。虽然移动设备的 CPU 可以满足设备的需求，但在性能方面还无法与桌面计算机上的 CPU 相媲美。桌面计算机拥有较高的点击速度、更大的内存，一般来说还具有更高速度和更高质量的 Internet 带宽。所以为了真实地测试一个复杂的设计，还是必须在移动设备上通过 Android 来测试应用程序，同时应用程序还需要运行在移动网络中。
- **当在移动设备上进行测试时，虽然 WiFi 也是一种移动网络，但通常办公室或者家中的 WiFi 可以提供比移动运营商的网络更高的质量。**一般来说，相比于 3G(或者

更差)的连接，WiFi 具有低延迟和高带宽的特点。所以必须在移动环境中对应用程序进行测试。同时还可以请一位同事开车带你去周围逛逛以便测试应用程序。

1.3 Android

毫无疑问，从 2010 年上半年起 Android 设备开始高速发展。虽然 Android 手机并不是第一款图形化手机，但是它却是第一款免费向手机设备制造商开放其软件的产品，从而使其更容易使用，并提供了一个可供购买应用程序的易于使用的市场。

1.3.1 Android 发展简史

Google 在 2005 年 7 月收购了一家名为 Android 的小公司，该公司主要从事移动软件的研发。通过这次收购，Google 开始关注移动设备领域。2006 年 12 月关于 Google 进军移动设备的传闻开始蔓延。2007 年秋，Google 组建了 OHA(Open Handset Alliance，开放手机联盟)，其宗旨是为移动设备创建一组标准。该联盟的核心是一个基于 Linux Kernel 版本 2.6(以及更新版本)的移动设备体系结构，以及可用于构建本机 Android 应用程序的 SDK。2008 年秋，Google 发布了第一款 Android 手机。

Android 的最初发布并没有被市场所认可。它遭到了媒体以及该平台首批用户的批评。因为，在那个时候 Android 相比于其他竞争平台的几大优势并没有显现出来。Android 是一个开放平台。因此，不仅移动设备制造商之间相互竞争，而且这些制造商还与 OHA 的其他成员进行竞争。这就意味着硬件级的创新步伐非常显著，此时相比其他平台，Android 平台的优势就显现出来了。Android 设备并不局限于某个制造商或者某个电信运营商。同样，电信运营商之间也必须相互竞争。这两个因素以及其他因素导致了 Android 和移动设备市场许多的创新和进步。

在经历了初期的阵痛后，Android SDK 得到迅猛发展(可在 Android SDK 中找到与 Mono for Android 开发人员有关的可用工具的讨论，本章后面将讨论该内容)。2007、2008 年发布了许多 beta 版本，而 SDK 的 1.0 版本于 2008 年 9 月正式发布。从那时起陆续发布了许多额外的 SDK 版本。

2009 年秋，OHA 推出了 Android 2.0(Eclair)操作系统。对于 Android 来说这可以说是一个分水岭。随着 Android 2.0 的发布，Motorola 也发布了 Droid 手机，同时 Verizon 公司开始推销该产品。从这点上看，Android 已经在市场上快速发展。

2010 年，OHA 发布了 Android 2.1。此外，HTC、Motorola 以及其他制造商也生产出了一系列的高端设备。而这些设备的发布进一步加快了 Android 的发展并提升其品牌知名度。与此同时，许多制造商也相继推出了基于 Android 的平板电脑设备。

2011 年初 OHA 又发布了基于 Android 3.0 的设备(a.k.a.Honeycomb)。该版本的 Android 针对平板电脑环境进行了相应的优化。但此版本的 Android 系统并没有被市场所认可。

2011 年末，Google 发布了 Android 4.0(Ice Cream Sandwich)。该版本的 Android 统一了