

微软.NET程序员系列

Microsoft Press

- ◆ 欧美读者评价 ★★★★★
- ◆ 移动设备专家精心编著
- ◆ 与具体开发实例相结合
- ◆ 全面介绍移动设备.NET开发
- ◆ 移动设备Web开发的必备书

移动设备.NET 应用程序设计

Building .NET Applications
for Mobile Devices

[英] Andy Wigley 著
Peter Roxburgh

赵明 李冠军

Visual Studio .NET产品组

等译
审校



Microsoft
.net

清华大学出版社

微软.NET 程序员系列

移动设备.NET 应用 程序设计

[英] Andy Wigley, Peter Roxburgh 著
赵 明 李冠军 等译
Visual Studio .NET 产品组 审校

清 华 大 学 出 版 社
北 京

内 容 简 介

本书是《微软.NET 程序员系列》丛书之一,主要介绍了怎样使用微软 ASP.NET 和 Mobile Internet Toolkit 来创建移动的 Web 应用程序的方法,这种应用程序能够以正确的格式在多种设备上动态地显示相同的页面。

本书主要内容包括将.NET 用于移动 Web、开发移动.NET 应用程序、使用移动 Web 控件、样式与设备定制的显示、状态管理、XML Web 服务、打包和安全性等内容。本书适用于从事移动设备应用程序开发人员以及对开发移动设备应用程序感兴趣的读者。此外,本书给出丰富的应用实例,可以帮助掌握 C# 和 Visual Basic .NET、ASP.NET 编程的读者提高。

Building .NET Application for Mobile Devices

Andy Wigley, Peter Roxburgh

Copyright©2002 by Microsoft Corporation.

Original English language edition published by Microsoft Press, a Division of Microsoft Corporation.

All rights reserved.

No Part of the Contents of this book may be reproduced or transmitted in any form or by any means without permission of the publisher. For sale in the People's Republic of China only.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号:图字 01-2003-0844 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

移动设备.NET 应用程序设计/(英)威格里著,(英)罗克斯巴勒著;
赵明等译.—北京:清华大学出版社,2003

(微软.NET 程序员系列)

书名原文:Building.NET Applications for Mobile Devices

ISBN 7-302-06422-9

I.移... II.①威...②罗...③赵... III.①移动通信—通信设备—程序设计②微型计算机—程序设计
IV.①TN929.5②TP368.33

中国版本图书馆 CIP 数据核字(2003)第 018537 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn>

<http://www.tup.tsinghua.edu.cn>

责任编辑:赵飞虎

印 刷 者:世界知识印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×960 1/16 印张:32 字数:767 千字 彩插页:5

版 次:2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号:ISBN 7-302-06422-9/TP·4840

印 数:0001~4000

定 价:58.00 元

《微软.NET 程序员系列》序

自 2000 年 6 月微软宣布自己的 .NET 战略以来,在不到两年的时间里,.NET 已经从战略变成现实。.NET 带来了全新的、快速而敏捷的企业计算能力,也给软件开发商和软件开发人员提供了支持未来计算的高效 Web 服务开发工具。作为微软.NET 战略的重要组成部分——Visual Studio .NET (中文版)已经于 2002 年 3 月 22 日正式在中国推出。

Visual Studio .NET 是一个功能强大、高效并且可扩展的编程环境。它充分展现了应用程序开发的潜能,并提供了生成应用程序所需的工具和技术。这些应用程序将给当今的企业、机构提供强大的支持,并推动下一代基于 XML Web 服务软件的发展。

有了 Visual Studio .NET,那些对全世界数百万的专业和业余程序员来说曾一度极端复杂、费时费力,甚至让人望而生畏的编程任务现在已不再神秘。更重要的是,Visual Studio .NET 使开发人员能运用既有的技能和知识来迎接新的编程挑战。

在 10 年前,Visual Basic 1.0 成为数以百万计的开发人员的革命性的应用程序开发语言。现在,Visual Studio .NET 为未来的 10 年做好了准备。

微软出版社为了配合 Visual Studio .NET 的推广以及 .NET 技术的普及,邀请 Visual Studio .NET 项目开发组的核心开发人员和计算机图书专业作家精心编写了英文版《微软.NET 程序员系列》丛书;该丛书自面市以来,在美国图书销量排行榜上一直高居前列,颇受好评,成为程序开发人员和网络开发人员了解 .NET 技术的权威工具书。尤其是《Microsoft .NET Framework 程序设计》一书,长期占据美国及欧洲此类书籍的排行榜冠军位置,程序开发人员不可不读此书。

清华大学出版社为了满足中国广大程序开发人员、网络开发人员学习最新技术的渴望,在微软出版社的配合下,从《微软.NET 程序员系列》这套丛书中精选了 50 余本翻译成中文,以满足国内广大读者的需要。这套丛书阵容庞大(且在不断扩充之中),几乎涵盖了 .NET 技术及其应用的各个方面;也正因为如此,翻译和编辑加工的工作量也大得惊人。但为了保持国外优秀技术图书的魅力,同时使读者领会新技术的真谛,本丛书的翻译和编辑都是经过严格筛选的、具有很高的翻译水平或丰富编辑经验的技术人员;另外,我们还聘请微软公司 Visual Studio .NET 产品组的技术专家审读每一本书,确保在技术上准确无误。

相信这套丛书定会帮助程序开发人员、网络开发人员以及那些具有一定编程基础的中、高级读者,快速、全面地掌握 .NET 技术,协助他们为技术生涯的下一个 10 年做好准备,为培养新一代软件人才,并推动中国软件产业的快速发展起到积极的作用!

这套丛书分为 3 个子系列:技术内幕系列、语言参考系列和程序员系列。目前,已出版和在编

1-1AC60/07

的共有 36 本, 已从 2002 年 6 月份起陆续和读者见面。

- 技术内幕系列

目前共有 7 本:

- ◆ 《Visual C++ .NET 技术内幕(第 6 版)》

本书是 Visual C++ 和 MFC 开发的经典著作。它秉承了第 4 版和第 5 版的风格, 已根据该编程语言的最新版本 Visual C++ .NET 进行了全面更新和补充, 是 .NET 时代的 C++ 程序员必读的教材。此外, 本版仍由第 4 版译者潘爱民先生翻译。

- ◆ 《Microsoft .NET Compact Framework 技术内幕》

- ◆ 《Visual Basic .NET 技术内幕》

- ◆ 《Visual C# .NET 技术内幕》

- ◆ 《ADO.NET 技术内幕》

- ◆ 《Microsoft .NET 程序设计技术内幕》

- ◆ 《Visual J# .NET 技术内幕》

- 语言参考系列

目前共有 3 本:

- ◆ 《Visual Basic .NET 语言参考手册》

- ◆ 《Visual C# .NET 语言参考手册》

- ◆ 《Visual C++ .NET 语言参考手册》

- 程序员系列

目前共有 27 本。详细书目, 请参见本书彩插页。

其中, Microsoft .NET Framework 类库(1-4 卷)是 .NET System 命名空间大全, 直接源自 Microsoft .NET Framework。这 4 卷书包含了 .NET Framework Class Library System 命名空间的全部详细信息。帮助开发人员充分利用 .NET Framework 来开发 Web 应用程序、客户端应用程序和 XML Web 服务。

随着技术的发展, 我们将根据读者的需要, 不断增加新的书目。

丛书版式特色

本丛书在风格上力求文字精炼。并采用小 5 号字编排, 内容紧凑, 版面清晰美观, 易于阅读。此外, 书中还安排了一些特色段落, 提供正文之外的一些细节知识:



注意: 提醒阅读和操作过程中应注意的事项, 避免出现错误或问题。



提示: 指点一些操作捷径或实用技巧, 使您少走弯路, 阅读和操作更为高效。



要点：总结关键知识点或操作细节，助您适时掌握要领。



注：提示首次出现的编程元素，以及书中涉及到该元素的其他位置以供参考。



警告：阅读和操作过程中应特别注意的事项，如果使用不当，可能会导致严重后果。

尽管我们倾心相注，精心而为，总有疏忽纰漏，恳请广大读者不吝赐教与指正，我们定会全力改进，以期在后续工作中得以完善。

本丛书在创作过程中得到了微软(中国)公司的大力支持。本丛书能够顺利出版，更是倾注了无数幕后人员的汗水和心力。在此，对他们的辛勤劳动一并表示衷心感谢！

编者

2003 年 3 月

前言

当带有 WAP(无线应用协议)功能的电话第一次涌入欧洲市场时,本书作者们就已经在一起工作了。和许多人一样,他们迫切地想得到一部 Nokia 7110 手机以便着手创建真正的无线 Web 应用程序。这真是快乐无比——一项新技术让数百万公众热切地期待从自己的 WAP 电话中可以访问到很有用的内容。本书作者们当时为一家从事 Internet 站点信用卡支付业务的公司工作,并且在英国创建了第一个可以通过 WAP 电话进行信用卡支付的实用设备。

遗憾的是,移动 Internet 业务并没有像预料的那样繁荣起来。移动电话公司对这种新技术疑虑重重,并且用户所具有的使用移动 Internet 经验还不是移动电话公司所期望的。用户需要像 PC 机那样的小型 Web 浏览器版本,而不是体积小、速度慢而且色彩单一的文本浏览器。“WAP 是垃圾”成为报纸常提到的一句话,很多公司对开设移动 Internet 业务都采取观望的态度。

也有令开发者感到惊异的事情。和许多开发者一样,作者很惊异地发现自己的 WAP 服务只能在 Nokia 浏览器上工作,而不能在 Phone.com 浏览器上正常工作。尽管这两种浏览器使用了相同的标记,但在不同浏览器上应用程序的可用性却完全不一样。作者重新设计了应用程序,使其在两种浏览器上都能够正常工作,这项工作充满了挑战性和乐趣,但是这样就减少了投入开发新解决方案的时间和精力。

幸运的是,移动 Internet 开始走向成熟。公众已经比较了解小型手持设备所具有的功能,WAP 服务正在稳步增长。日本试图避免欧洲所犯的市场错误,因此 i-mode 服务获得了巨大成功,它拥有超过 3 千万的订户(这是个令人惊喜的数字)。但是,从开发者的角度来看,最有前途的是那些看到移动服务有利可图的企业的利润获得增长。PDA(便携式信息助理)和智能电话的功能越来越强大,价格也越来越便宜,它们为商业 Web 应用程序提供了优秀的平台。

不过,现在仍旧存在很多不同类型的平台。它们具有不同的屏幕尺寸,各自支持不同的 HTML、cHTML 和 WAP 标记语言(以及它们的不同版本),以及是否支持彩色和单色显示等等。幸运的是,软件公司开发的软件使开发者只创建一次应用程序就可以让软件对特定设备进行优化输出。作者认为所有这类产品中最令人激动的就是 Microsoft Mobile Internet Toolkit 了。

.NET 框架是 Microsoft 最具雄心的开发项目之一。开发人员如何为 Windows 平台及万维网(World Wide Web, WWW)创建应用程序之类的书籍都得重写。Microsoft 的 .NET 所提供的功能中最令人激动的部分之一是 ASP.NET。使用此项技术,Web 开发人员在编写解决方案时就不会再感到受到很大的束缚了。ASP.NET 应用程序是成熟的 .NET 应用程序,它可以访问 .NET 框架的所有资源,并将标记作为输出结果。Mobile Internet Toolkit 扩展了 ASP.NET 的能力,使开发移动 Web 应用程序变得更容易,并且运行库能使输出结果匹配很多种手持设备。

“适应性、能够自定义、可扩展性”是 Microsoft 的 Mobile Internet Toolkit 开发团队经常重复的一句口号，这句话说明了此工具包的优良性能。为使应用程序适应不同性能的移动客户端，可以方便地对应用程序进行自定义以便发挥特定设备的独有能力。同时，该产品是可扩展的，因此可以方便地创建新控件或为新的手持设备提供支持。

开发早期移动应用程序经常遇到挫折与挑战。使用 Mobile Internet Toolkit 开发应用程序所用的时间和过去相比非常短，并且开发完成后可以立即从很多种浏览器上访问此应用程序，这些浏览器支持 HTML、cHTML 和 WML。

读者对象

本书适用于两种读者。第一类读者是已经具有开发手持设备应用程序经验的无线开发人员。他们可能刚刚进行 Microsoft 程序的开发并且可能没有用过 Visual Studio .NET。第 2 章和第 3 章的内容主要就是针对这些人，这两章介绍了 ASP.NET 和 Visual Studio .NET 并且指导读者进行移动 Web 应用程序的开发。第 4 章说明了 ASP.NET 应用程序进行工作的基础知识，读者需要了解它是如何与移动 Web 窗体一起工作的。

第二类读者是已经具有 .NET 框架和 Visual Studio .NET 开发经验的人员。如果读者以前使用过 ASP.NET，可以略读第 3 章以便了解 Mobile Internet Designer，然后直接进入第 5 章学习使用移动控件工作。

无论读者具有何种知识背景，都需要他们熟悉面向对象程序设计。.NET 框架以及所有以它为基础创建的内容都是完全面向对象的。和 ASP.NET 中的其他内容一样，Mobile Internet Toolkit 控件是类对象，读者需要了解类、方法、属性和继承以便充分利用工具包和 .NET 框架。

读者用不着熟悉 HTML 或 WML 标记语言。熟悉一种编程语言更为重要，例如熟悉 Visual Basic 或 Visual C#。这里要强调的是，所要编写的程序是用于输出标记的面向对象程序。如果未曾编写过设备定制标记，而要编写出功能完备的 Mobile Internet Toolkit 应用程序，这也是有可能的。如果想为特定手持设备自定义应用程序，熟悉一下 HTML 和 WML 会有所帮助。使用模板功能所能做的事情之一是向设备直接发送“原始”标记。当然，想要开发自己控件的高级开发人员必须完全熟悉设备所使用的标记语言。

本书所有示例代码都以 C# 语言编写，C# 是一种与 .NET 框架同时开发的 Microsoft 新的主流编程语言。希望 Visual Basic 开发人员不会由于人们对 C# 的关注而感到倍受冷落。实际上，C# 和 Visual Basic 代码的结构非常相似，并且除了明显的语言语法区别之外，C# 示例对于 Visual Basic .NET 开发人员来说应该具有非常好的可读性。本书中所有示例都有 C# 和 Visual Basic 两种版本，但是第 15 章的后半部分和第 16 章的自定义控件示例却是例外。这两部分中仅有 C# 版本的代码，这并不是因为不能使用

Visual Basic(或.NET 框架所支持的其他语言), 而是因为作者来不及编写其他语言版本的代码了。

本书内容

第 1 章通过移动数据通信程序开发的简史, 得出当前情况是如何形成的。第 1 章还介绍了移动 Web 应用程序开发人员所面临的挑战, 并介绍了 Mobile Internet Toolkit 如何应对这些挑战。第 2 章继续前一章的主题, 但主要关注 ASP.NET, 以及它为何优于 Web 开发人员以前所使用的工具。

第 3 章概要介绍了 Visual Studio .NET, 其重点是 Mobile Internet Toolkit 所具有的性能。这一章介绍了 Mobile Internet Designer 如何从 Toolbox 中拖动移动控件并将其放到移动 Web 窗体页面上, Mobile Internet Designer 使用拖放方式的 GUI 编辑器设计应用程序。第 4 章介绍 ASP.NET 应用程序开发的重要基础知识。如果刚刚接触 ASP.NET 就需要重点阅读这一章, 这章介绍了如何处理客户端和服务器的请求响应交互, 以及移动设备用户的指令如何转换成服务器代码中所捕获的事件。

第 5 章到第 7 章介绍了各种标准移动控件。其目的是提供每种控件的简要说明, 以使读者能够学会如何使用 XML 语法在移动 Web 窗体页面中应用控件, 以及如何在代码中访问控件的属性和方法。每种控件都包含一个或多个示例应用程序以演示其使用方法。

第 8 章介绍 Mobile Internet Toolkit 的特性, 它可以用来增强应用程序的显示效果。这些特性可分为 3 个不同的功能类别: 样式、属性重写和模板。通过样式可以定义用于显示控件外观的颜色和字体, 这将为那些支持这种功能的浏览器增色添光。属性重写用来自定义应用程序, 这对特定型号或类型的客户端设备来说就可以为控件属性指定不同的值。模板具有强大的功能, 可以用它来对列表控件的显示方式进行大量的自定义, 或者可以用它来将设备定制标记插入到发送给特定设备的输出结果中。

第 9 章到第 14 章介绍 Mobile Internet Toolkit 开发人员可能用到的其他功能。其主要内容包括处理 ADO.NET 数据、使用 Visual Studio .NET 和移动设备仿真器测试和调试、典型设计示例以及如何国际化应用程序、处理状态管理以及打包和部署应用程序。

第 15 章到第 17 章介绍 Mobile Internet Toolkit 的可扩展能力。其中的大部分内容是针对高级开发人员的, 只有第 15 章的第一部分涉及到用户控件, 可以使用用户控件方便地为移动 Web 应用程序开发可重用的可视化组件。第 15 章的第 2 部分和第 16 章的全部内容介绍了如何编写自定义移动控件代码。第 17 章介绍如何使 Mobile Internet Toolkit 支持新设备。开发者可以一直等到 Microsoft 发布支持新设备的更新的程序, 或者可以在本章的指导下自行添加支持功能。

使用本书示例

如果要使用本书示例, 就必须安装 Internet 信息服务(Internet Information Services, IIS)。读者可

以在已经安装了 Visual Studio .NET 的本地机器上运行示例，也可以在专用服务器上运行示例。但无论使用什么方式，都必须具有安装与运行安装程序及脚本的权限。

系统需求

运行示例程序需要以下软件：

- Visual Studio .NET 或 .NET 框架 SDK
- Microsoft Mobile Internet Toolkit
- Windows 2000 或 Windows XP Professional

安装 MSDE .NET 框架示例数据库

第 9 章中的一些示例使用了公共数据库，公共数据库安装了 .NET 框架 SDK QuickStart 示例。不必在开发系统上安装 SQL Server 产品，因为 .NET 框架 QuickStart 示例的安装程序会在必要时安装一个名为 MSDE 的独立数据库服务。如果要安装 MSDE Server 和示例数据库，可以在 C:\Program Files\Microsoft.NET\Framework SDK\ 文件夹或 C:\Program Files\Microsoft Visual Studio .NET\Framework SDK 文件夹下双击 StartHere.htm，这样就会出现 Microsoft .NET 框架 SDK 欢迎页面。单击 QuickStarts, Tutorials, And Samples 链接。如果还没有安装 .NET 框架 QuickStart 示例，则显示页面给出安装示例所必须的两个步骤。首先单击 Step1: Install The .NET 框架 Samples Database。数据库安装好之后，再单击 Step2: Set Up The QuickStarts 来安装所有的示例数据库，并安装 .NET 框架 QuickStart 指南。

使用配置脚本

示例中包含一个安装程序，这个安装程序可以将示例文件复制到硬盘中的一个目录下。运行安装程序后，就要运行下面的脚本来设置 IIS 中运行示例应用程序时所需的虚拟目录。

1. 打开 Windows 的【命令提示符】窗口。如果已经安装了 Visual Studio .NET，则可以单击【开始】|【所有程序】| Microsoft Visual Studio .NET | Visual Studio .NET Tools | Visual Studio .NET Command Prompt。这样就可以确保 PATH 环境变量包含 .NET 框架编译器的位置。
2. 如果安装了 .NET 框架 SDK，而没有安装 Visual Studio .NET，则可以单击【开始】|【所有程序】|【附件】|【命令提示符】打开【命令提示符】窗口。然后必须将 .NET 框架编译器的位置添加到 PATH 环境变量中。此位置在 WINDOWS 文件夹 \Microsoft.NET\框架\版本文件夹当中。例如，要更新 Windows XP 系统中的 PATH 变量，输入如下语句：

```
set path=%path%;C:\Windows\Microsoft.NET\Framework\v1.0.3705
```

Windows 目录在 Windows 2000 系统和从 Windows 2000 升级得到的 Windows XP 系统中被命名为 WINNT。

3. 在【命令提示符】窗口中, 进入安装示例的文件夹。
4. 执行配置 IIS 的脚本, 使所有 Web 服务器上的示例都可用。将安装示例的目录名称(当前目录)作为命令行上的一个参数而输入:

```
makeVirtualDirectories D:\BookSamples
```

运行示例目录中另一个脚本来编译示例中所有代码的隐藏模块(code-behind)和代码库:

```
build
```

访问示例

刚才运行的配置脚本对 IIS 进行了设置, 这样所有的示例都处于 URL `http://servername/MITBookSamples` 之下, 本书的每章都在此处有一个文件夹。每章的文件夹都包含一个 C# 示例文件夹和一个 Visual Basic 示例文件夹。本书所包含的代码在这些示例中都可用。

要访问一个示例, 必须首先检查示例目录的名称以及该目录中 .aspx 文件的名称, 因为这个名称是您使用 URL 的最后面部分。例如, 如果要运行第 3 章的 Visual Basic MyFirstMobileApp 示例, 可以在所使用的浏览器中输入下面的 URL:

```
http://servername/mitbooksamples/chapter3/vb/myfirstmobileapp/default.aspx
```

使用 Visual Studio .NET 项目

所有的示例都有 Visual Studio .NET 项目文件。如果想要在 Visual Studio .NET 中打开示例的项目, 使用 Windows 的【资源管理器】双击 .csproj 或 .vbproj 文件来打开它。第一次在 Visual Studio .NET 调试器中运行这些项目时, 会出现一个提示要求必须选择项目的开始页面。右击 Solution Explorer 中的 .aspx 文件并单击 Set As Start Page。

如果要打开特定章节的所有项目, 在每章的 \cs 或 \vb 文件夹中有一个 Visual Studio .NET 解决方案文件(扩展名为 .sln)。双击解决方案文件, 同时打开该章的所有项目。

删除示例

如果要将示例从系统中删除, 请按照下面步骤进行:

1. 打开【Internet 服务管理器】。单击【开始】|【设置】(对 Win2000)|【控制面板】|【管理工具】|【Internet 服务管理器】。打开浏览树状结构并找到【默认 Web 站点】, 单击展开它。
2. 右击列表中的 MITBookSamples, 然后单击【删除】, 单击【确定】以确认删除。
3. 在【控制面板】中单击【添加/删除程序】, 从列表选择 MIT Book Samples, 单击【删除】。

支持信息

为了降低书的成本，减轻读者负担，对于因内容很少而不值得单独配盘的图书，我们将其范例代码或练习文件放在我们的网站上，供读者下载。敬请访问以下网址：<http://www.wenyuan.com.cn>，查找本书的有关链接。

如果您对本书或配书文件有任何建议、意见或想法，请通过以下电子邮件与清华大学出版社计算机应用编辑二室客户服务部取得联系：

service@wenyuan.com.cn

或致函：

北京 100084-157 信箱

读者服务部

邮编：100084

亦可致电：010-62792098-220。

请注意，上述地址并不提供软件产品的支持。

目 录

前言	xi	2.4.2 实现代码	20
第 1 章 将 .NET 引入移动 Web	1	2.4.3 事件响应	21
1.1 无线 Web 的诞生	2	2.5 XML Web 服务简介	21
1.1.1 Internet	3	2.6 会话管理与安全	21
1.1.2 无线通信	3	2.6.1 使用会话	22
1.1.3 移动数据通信	4	2.6.2 使用 cookie	22
1.2 Mobile Internet Toolkit	7	2.6.3 分布式体系结构中的部署	23
1.3 无线混乱的解决方案	11	2.7 调试移动应用程序	23
第 2 章 ASP.NET 与 Mobile Internet Toolkit	12	2.7.1 使用简单工具	23
2.1 ASP.NET 概述	12	2.7.2 程序调试	23
2.1.1 背景	12	2.7.3 使用 Visual Studio .NET 调试	24
2.1.2 更简单的 ASP	12	第 3 章 开发移动 Web 应用程序	25
2.1.3 ASP.NET: Web 开发的新途径	13	3.1 安装开发系统	25
2.2 开发 Web 应用程序	14	3.2 创建第一个移动 Web 应用程序	26
2.2.1 进一步了解 ASP.NET Web 窗体	14	3.2.1 使用 Visual Studio .NET	26
2.2.2 使用 Web 窗体控件	15	3.2.2 创建移动 Web 项目	28
2.2.3 使用 .NET 工具开发 Web 应用程序	16	3.2.3 使用【解决方案资源管理器】和【属性】对话框	30
2.2.4 扩展 Web 应用程序	16	3.2.4 使用 Toolbox 创建用户界面	31
2.3 使用移动 Web 窗体进行开发	17	3.2.5 生成和执行应用程序	32
2.3.1 了解设备性能	17	3.3 使用移动电话仿真器进行测试	33
2.3.2 抽象移动设备的用户界面	17	3.3.1 安装 Openwave 仿真器	33
2.3.3 使用移动 Web 窗体控件	18	3.3.2 使用仿真器进行测试	34
2.3.4 在移动 Web 窗体中使用 .NET 工具	19	3.4 使用 Mobile Internet Designer	36
2.4 .NET 语言简介	19	3.4.1 移动 Web 窗体页面和 Form 控件	37
2.4.1 符合 CLS 的程序设计语言	19	3.4.2 使用多个移动 Web 窗体	38
		3.4.3 在移动 Web 应用程序中向前向后导航	40

3.4.4	复杂控件在特定设备上的 再现	41
3.4.5	在 Visual Studio .NET 中获得 帮助信息	42
3.5	Visual Studio .NET 项目和 SDK 示例 的关系	43
3.6	使用 Visual Studio .NET 项目	46
3.6.1	Visual Studio .NET 移动 Web 应用程序项目所创建的文件	46
3.6.2	选择项目的存放位置	49
3.6.3	指定 Web 服务器的访问 方式	49
3.6.4	复制项目	50
3.6.5	打开项目	51
3.6.6	删除项目	52
3.6.7	团队开发	52
3.6.8	脱机工作	53
第 4 章	动态移动应用程序	54
4.1	编程策略	54
4.1.1	选择语言	54
4.1.2	内联代码和代码隐藏	55
4.2	Hello Universe 应用程序	57
4.3	日程表程序	61
4.3.1	不变性类成员	62
4.3.2	生成并测试应用程序	64
4.3.3	应用程序总结	65
4.4	为日程表程序添加功能	65
4.4.1	生成并测试应用程序	68
4.4.2	应用程序的生存期	68
4.4.3	应用程序总结	69
4.5	时钟程序	69
4.5.1	生成用户界面	69
4.5.2	创建代码隐藏模块	70
4.5.3	生成并测试应用程序	73

第 5 章	使用 Mobile Internet Toolkit 标准控件	74
5.1	控件的类层次	74
5.2	移动控件共有的功能	78
5.3	使用控件示例	82
5.4	容器控件	82
5.4.1	包容规则	83
5.4.2	Form 控件	84
5.4.3	Panel 控件	88
5.5	核心控件	90
5.5.1	Label 控件	91
5.5.2	TextBox 控件	93
5.5.3	TextView 控件	98
5.5.4	Link 控件	101
5.5.5	Command 控件	104
5.5.6	Image 控件	109
第 6 章	列表控件	116
6.1	SelectionList 控件	118
6.1.1	语法	118
6.1.2	属性	119
6.1.3	用法	120
6.1.4	绑定到数据集合	125
6.2	List 控件	128
6.2.1	语法	128
6.2.2	属性	129
6.2.3	用法	130
6.3	ObjectList 控件	135
6.3.1	语法	135
6.3.2	属性	136
6.3.3	用法	139
第 7 章	特殊用途控件和验证控件	152
7.1	特殊用途控件	152
7.1.1	Calendar 控件	153
7.1.2	PhoneCall 控件	156

7.1.3 AdRotator 控件	160	8.4.6 在样式表中定义模板	247
7.2 验证控件	164	第 9 章 访问数据	250
7.2.1 RequiredFieldValidator 控件	165	9.1 使用 List 控件的高级数据绑定 功能	250
7.2.2 CompareValidator 控件	168	9.1.1 定义静态 List 和 SelectList 的列表项	250
7.2.3 RangeValidator 控件	172	9.1.2 绑定到数据集合	251
7.2.4 RegularExpressionValidator 控件	176	9.1.3 重写 List 控件中的单个字段 显示	254
7.2.5 CustomValidator 控件	179	9.2 使用 ASP.NET 声明数据绑定	256
7.2.6 ValidationSummary 控件	182	9.2.1 使用 DataBinder.Eval	259
7.2.7 验证控件示例	185	9.2.2 理解命名容器	259
第 8 章 样式和设备定制的显示	190	9.3 使用 ADO.NET	261
8.1 概述	190	9.3.1 理解 ADO.NET 对象	262
8.1.1 样式	190	9.3.2 DataReader 对象用于对只读 数据的访问	263
8.1.2 属性重写	190	9.3.3 用 DataSet 对象进行数据 绑定	264
8.1.3 模板	191	9.3.4 创建移动 Web 应用程序进行 数据库更新	265
8.2 使用样式和样式表	191	9.4 Microsoft 的可视化数据库工具	271
8.2.1 使用样式属性	192	9.4.1 使用 Server Explorer	271
8.2.2 使用 StyleSheet 控件	196	9.4.2 创建数据组件	272
8.2.3 在 Visual Studio .NET 中 使用样式	204	9.4.3 在 Web 应用程序中使用 数据组件	275
8.3 用属性重写配置设备定制	206	9.5 学习更多 ADO.NET 的知识	278
8.3.1 <DeviceSpecific>和<Choice> 元素	207	第 10 章 状态管理	279
8.3.2 设备过滤器	210	10.1 会话状态	279
8.4 使用模板化控件	222	10.1.1 使用 Session 对象	281
8.4.1 使用 Form 控件模板	224	10.1.2 使用 cookie	283
8.4.2 使用 List 和 ObjectList 控件 模板	230	10.1.3 使用混合的 URL	284
8.4.3 使用 Panel 控件模板	238	10.1.4 为何要使用 cookie	284
8.4.4 通过代码使用模板中的 控件	241	10.2 隐藏变量	285
8.4.5 使用 Visual Studio .NET 工具定义模板	244	10.3 视图状态	288

10.4	应用程序状态.....	291	12.6	访问数据.....	339
10.4.1	在 Global.asax 中使用应用 程序状态	291	12.7	移动设备的注意事项.....	344
10.4.2	使用应用程序状态时的 注意事项	296	第 13 章	调试、测试仿真程序并处理 运行库错误.....	346
第 11 章	设计强制的移动网络应用 程序	298	13.1	调试移动 Web 应用程序	346
11.1	基本设计原则.....	298	13.2	使用跟踪工具.....	349
11.2	创建集成桌面和移动访问的 ASP.NET 应用程序.....	301	13.2.1	添加跟踪语句.....	350
11.2.1	为客户端设备调整功能.....	301	13.2.2	使用应用程序级的跟踪	351
11.2.2	为移动 Web 站点完成重定 向页面	301	13.3	处理错误.....	353
11.3	优化性能	302	13.3.1	实现自定义的错误页面	354
11.3.1	使用页面输出缓存.....	305	13.3.2	编程处理错误.....	357
11.3.2	使用片段缓存.....	311	13.4	用仿真程序和真实设备测试.....	359
11.3.3	使用数据缓存.....	311	13.4.1	使用 Pocket Internet Explorer 进行测试.....	360
11.4	创建多语言和多元文化的移动 Web 应用程序.....	312	13.4.2	使用 Pocket PC 仿真程序 测试.....	361
11.4.1	定义字符串、日期和时间 格式	313	13.4.3	使用移动电话仿真程序 测试.....	362
11.4.2	支持本地化内容.....	315	13.4.4	确认对仿真程序的支持	363
11.4.3	定义字符集编码.....	319	13.4.5	将仿真程序集成到 Visual Studio .NET 中.....	371
第 12 章	XML Web 服务	321	第 14 章	打包、配置和安全	373
12.1	创建 XML Web 服务	322	14.1	打包和部署应用程序.....	373
12.1.1	使用文本编辑器.....	322	14.1.1	复制应用程序.....	374
12.1.2	使用 Visual Studio .NET	323	14.1.2	创建 Web 安装项目	375
12.2	部署和使用 XML Web 服务	324	14.2	配置应用程序.....	378
12.2.1	使用命令行工具.....	324	14.2.1	理解.NET 配置文件	379
12.2.2	使用 Visual Studio .NET	327	14.2.2	Web.config 配置示例	381
12.2.3	使用其他技术.....	329	14.3	ASP.NET 应用程序的安全性.....	382
12.3	定义 XML Web 服务的行为	330	14.3.1	提供验证程序.....	382
12.4	管理会话和应用程序状态.....	332	14.3.2	角色和用户授权.....	386
12.5	使用数据类型.....	335	第 15 章	创建用户控件和自定义 控件.....	388
			15.1	生成用户控件.....	388

15.1.1 创建简单的用户控件.....	388	16.3.2 实现 OnDataBind 事件.....	446
15.1.2 用户控件模块的编码.....	389	16.4 在自定义控件中支持视图状态.....	448
15.1.3 在 Web Forms 页面中使用 用户控件	390	16.4.1 将控件属性保存至 ViewState 对象	448
15.1.4 用户控件示例.....	390	16.4.2 理解 IStateManager 接口	449
15.2 在代码中生成控件.....	401	16.4.3 使用 PrivateViewState.....	452
15.2.1 了解控件的生命周期.....	401	16.5 实现自定义 MobileControlBuilder ..	453
15.2.2 通过继承生成控件.....	404	16.6 在自定义控件中启用客户端回送 ...	455
15.2.3 使用编译过的自定义 控件	407	16.7 处理发送数据.....	459
15.2.4 通过复合生成控件.....	409	16.7.1 在 CMTable 控件中实现 发送数据处理.....	460
第 16 章 从零开始生成控件和使用 设备适配器.....	421	16.7.2 在 DeviceAdapter 类中提供 不同的用户接口	461
16.1 从零开始生成控件.....	421	16.7.3 在设备适配器类中添加 子控件.....	464
16.1.1 控件生命周期.....	422	16.8 开发模板自定义控件.....	466
16.1.2 设备适配器的作用.....	422	16.8.1 实现模板支持.....	467
16.2 生成简单的自定义控件和设备 适配器	428	16.8.2 为数据绑定 List 控件实现 模板支持.....	468
16.2.1 具有 HTML 和 WML 设备 适配器的简单控件.....	428	第 17 章 支持新型客户端.....	475
16.2.2 编写具有 MobileText Writer 类的设备标记	433	17.1 通过配置文件支持新型客户端.....	475
16.2.3 在设备适配器中处理 样式属性	433	17.2 支持新型客户端.....	478
16.2.4 使用自定义控件和设备 适配器	434	17.2.1 识别移动设备.....	478
16.3 生成数据绑定自定义控件.....	437	17.2.2 识别设备功能.....	480
16.3.1 支持内部和自定义分页.....	444	17.3 使用设备适配器.....	487
		17.3.1 设备适配器的作用.....	487
		17.3.2 使用设备适配器集.....	488
		17.3.3 定义设备适配器集.....	490