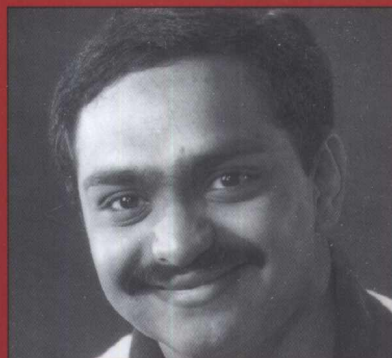
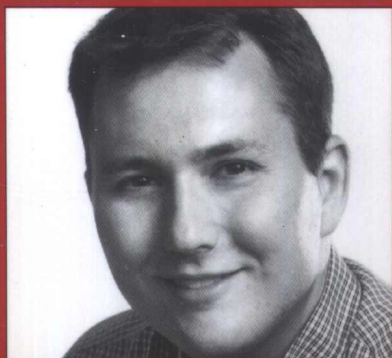
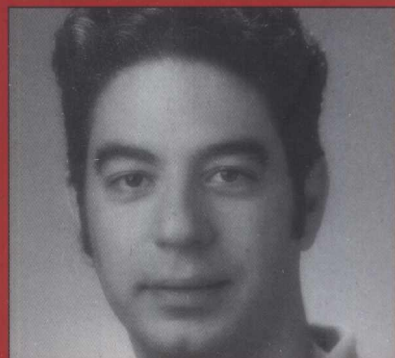
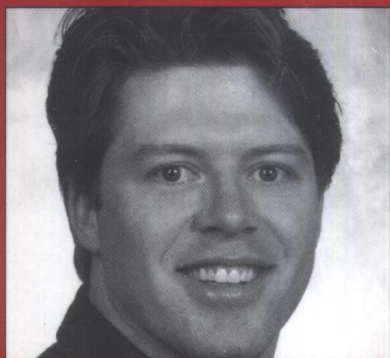
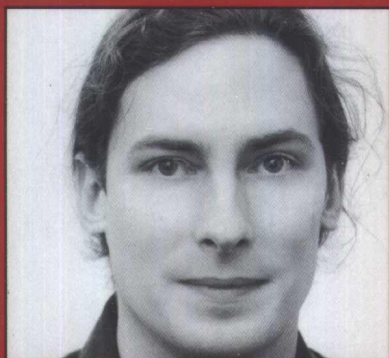


PROGRAMMER TO PROGRAMMER™



ASP.NET Mobile Controls

ASP.NET

移动控件编程

Matt Butler

Matthew Gibbs 等著 康博 译



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



226

7P3/3.4
B11

ASP.NET 移动控件编程

Matt Butler
Matthew Gibbs
康 博

等著

译



A0996542

清华大学出版社

(京) 新登字 158 号

北京市版权局著作权合同登记号: 01-2002-2469

内 容 简 介

本书深入浅出地介绍了利用 ASP.NET 和 Microsoft Mobile Internet Toolkit 进行移动 Web 程序开发等方面的内容。具体包括: .NET 移动概述、ASP.NET 移动控件概述、各种控件的应用及高级功能、客户端设备仿真器、Visual Studio .NET 集成和案例分析等等。

本书内容全面, 语言生动, 各个章节很好地结合了大量示例, 非常适合广大从事 Web 开发或在此领域有浓厚兴趣的程序员阅读学习, 同时也可用作各大专院校的相关教材。

Matt Butler, Matthew Gibbs et al: ASP.NET Mobile Controls Tutorial Guide —— Adaptive Web Content For Mobile Devices With The MMIT.

EISBN: 1-861005-22-9

Copyright©2001 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由清华大学出版社和英国乐思出版公司合作出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

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。

书 名: ASP.NET 移动控件编程

作 者: Matt Butler, Matthew Gibbs 等著 康博 译

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.tsinghua.edu.cn>

责任编辑: 郭东青

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印 张: 22.25 字 数: 569 千字

版 次: 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

书 号: ISBN 7-302-05565-3/TP·3286

印 数: 0001~4000

定 价: 38.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前言

ASP.NET 移动控件可以为开发人员提供创建交互式 Web 站点所需的工具, 这些工具可用于不同的移动客户设备上。它们包含在名为 Microsoft Mobile Internet Toolkit 的可选的 ASP.NET 插件中, 它包括控件本身和用来同 Visual Studio .NET 集成的附加功能。

本书分为 3 个主要部分:

- 介绍 WAP、WML 以及围绕用有限资源进行客户端编程的概念。
- 举例说明如何使用各个 ASP.NET 移动控件, 生成目标输出, 并根据查看网页的不同个人设备来调整输出。
- 为您示范在 Visual Studio .NET 开发环境中如何利用这些控件, 最后以一个完整的移动应用程序案例分析结束。

本书为 ASP.NET 程序员介绍了为移动客户端设备创建有意义的内容所需了解的所有概念和控件。

本书读者对象

本书是专门针对那些正在通过使用 .NET Framework 为移动设备创建或计划创建移动设备应用程序的开发人员编写的。因此您必须具备使用 ASP.NET 的经验, 还需要具备 C# 编程经验。本书的读者应该对创建 ASP.NET 应用程序不感到困难, 而且能够很容易地理解简单的 C# 代码。

本书还包含了对 ASP.NET 基本内容的简要回顾, 但这对于初学者来说可能过于高深。

注意:

如果您需要了解有关 ASP.NET 的更多信息, 可以参阅《ASP.NET 入门经典——C# 编程篇》(清华大学出版社出版)。

本书主要内容

本书是根据 Microsoft Mobile Internet Toolkit 的试用版编写的, 该版本所运行的平台为 .NET Framework 试用版, 同时, 本书中所提供的程序代码均可在 .NET 正式版上运行。本书包括以下内容:

ASP.NET 移动控件简介

从第 1 章到第 3 章, 将讨论有关移动通信的主要内容和移动客户端设备的优缺点, 并开始讨论为了最大限度地发挥这些设备的潜能, 开发人员需要具备怎样不同的思维方式。接着, 讨论 MMIT 的安装和设置程序以及客户端浏览器的选择。另外, 还要介绍如何使用 ASP.NET 和移动控件进行文件管理的基本内容。然后, 本章将深入讨论移动通信的详细内容, 并结合过去的通信技术, 以及这些技术的来源和将来可能的发展方向。



ASP.NET 移动控件

从第 4 章到第 9 章,将详细讨论 ASP.NET 移动控件。首先介绍整个 ASP.NET 移动 Web 窗体的生存期,并将之与传统 ASP.NET Web 窗体比较。然后,继续介绍 Microsoft Mobile Internet Toolkit 所提供的基本控件,以及如何能够以多种方式运用它们来产生有意义的结果。本章还将通过使用一个简单的 Try It Out 示例来论证这些观点,并将通过多种不同的浏览器来查看输出,以便清楚地展示各个客户端在表达方式上的不同。

高级功能和优化

从 10 章到 16 章,将介绍更高级的 ASP.NET 移动控件,例如与验证用户输入和处理复杂输出相关联的控件。然后讨论如何来调整您的控件所产生的输出格式,以及如何保护您的移动 Web 应用程序拒绝未经授权的访问。另外,还将介绍如何为新设备增加对 MMIT 的支持,以及如何创建自己的移动控件。

Visual Studio .NET 集成

本书的最后各章将介绍如何利用 Visual Studio .NET 创建标准 Web 应用程序和移动应用程序。这部分不仅介绍 IDE 所提供的功能,还将展示如何简便地将 MMIT 与之集成。最后,通过一个案例分析来结束本书的学习,该案例是在 Visual Studio .NET 中使用 MMIT 创建的,它汇集了本书中的所有概念,并用它们来解决一个真实世界的问题。

运行 ASP.NET 需要做些什么

为了回答这个问题,首先要明了您自己在本书中的角色,您的角色是 Web 开发人员或 Web 管理员。因此,您要用这个角色编制 Web 页面,要在 Web 服务器上发布 Web 页,并对它们进行测试以确定其外观和能否正常运行。这只是您必须做的一部分工作。如果您还找不到相关内容,也不必担心,因为下面我们就来看看您可以从什么地方获得这些内容。

为了能够使用本书,您需要配置以下几项:

- .NET Framework(可以从 Microsoft 的官方网站获得)。
- 一个 ASP.NET 适用的 Web 服务器(ASP.NET 要求 IIS 5.0 或更高版本,也就是说要运行于 Windows 2000 Professional 或更高版本,或者 Windows XP Professional 上)。
- Microsoft Mobile Internet Toolkit(同样可从 Microsoft 的官方网站获得。具体内容请参见第 2 章)。
- 一些用于测试的客户端设备仿真器(都可以免费下载。具体内容请参见第 2 章)。

以下为可选项:

- Visual Studio .NET, 用于本书最后部分的综合章节及案例分析。

注意:

本书中的代码均是在 .NET Framework 正式版上运行设计的,因此不能保证在其他版本上能够正确运行。

客户支持

我们很看重来自读者的反馈意见，期待您对本书的看法建议，包括您喜欢哪些内容，不喜欢哪些内容，以及您对我们下一次的尝试有什么期待。您可以给我们发电子邮件，邮箱地址是 feedback@wrox.com，以便让我们知道您的意见和建议。请确保您的邮件中包含了本书名。

如何下载本书中的代码示例

请访问 Wrox 的站点，网址为 <http://www.wrox.com/>，您只要通过 Search 功能定位标题，或者使用一个标题列表定位标题。然后单击 Code 栏中的 Download，或者单击本书中详细信息页上的 Download Code。

从我们的站点下载的文件都是经过 WinZip 压缩的。当您将这些文件保存到自己硬盘的文件夹中时，要用 WinZip 或 PKUnzip 等解压缩程序来解压缩文件。而且解压缩时，文件中的代码通常要保存到专门的文件夹中。所以在开始解压缩过程时，要确保将您的软件(WinZip、PKUnzip 等)设置为使用这些文件夹名。

勘误表

虽然我们已经努力尽量保证本书中的代码或文字不出错，但是没有人能够完美无暇，书中难免出现错误。如果您在书中发现了诸如拼写错误或错误代码，请告诉我们，我们将非常感激。您发来的勘误表可以节省其他读者的时间，当然也有助于我们提高质量。您可以发电子邮件给我们，地址是 support@wrox.com，我们会查看您提供的信息，如果正确的话，则将它递交给本书的勘误网页，用于本书的后续版本中。

您可以访问 <http://www.wrox.com/>，并通过 Advanced Search 或标题列表简单地定位标题，在我们的网站上找到勘误表。单击本书详细信息页封面图下方的 Book Errata 链接。

电子邮件支持

如果您希望能够直接与熟悉本书的专家讨论书中的相关问题，那么请发电子邮件至 support@wrox.com，并在电子邮件的主题中附带书名和 ISBN 的最后 4 位数。一封标准的电子邮件应该符合下面的要求：

- 电子邮件主题栏中要填入书名、ISBN 的最后 4 位以及问题所在的页码。
- 正文中要有您的姓名、联系方式以及您的问题。

另外，我们不会给您发送垃圾邮件。我们需要您的详细信息以便节省您和我们的时间。当您通过电子邮件发送消息时，会得到一系列的支持：

- 客户支持 —— 您的消息首先将被送给我们的客户支持人员。他们是读到您的邮件的第一批人。他们备有常见问题的文件，并会立即为您解答任何关于本书或我们的网站的一般性问题。
- 编辑 —— 更深层次的问题会被送至负责本书的技术编辑。他们具有使用程序设计语言或特殊产品的丰富经验，能够解答某一主题的详细技术问题。
- 作者 —— 最后，如果我们的编辑不能回答您的问题，他或者她就会将问题交给作者。一般来说，我们尽力保证作者的写作不受干扰；但是，我们也很乐意将这类特殊的问题交由



他来处理。Wrox 出版社的所有作者都协助为书籍提供售后支持。他们会将自己的答复通过电子邮件发给客户和编辑，以便让所有的读者受益。

Wrox 支持部门只提供对已发布主题相关的问题的支持，通过 <http://p2p.wrox.com/forum> 上的社团列表，您可以找到对 wrox 书内容以外其他问题的支持。

p2p.wrox.com

如果希望与作者或同行进行讨论，请加入 P2P 邮件列表。通过我们特有的系统，程序员之间可以通过邮件列表、论坛、新闻组以及一对一的电子邮件支持系统进行联系。如果您向 P2P 发出请求，许多在我们的邮件列表上注册的 Wrox 作者及其他技术专家一定会为您作出解答。当您阅读本书时，或是当您开发自己的应用程序时，在 p2p.wrox.com 网站上，您都会找到很多对您有所帮助的列表。特别是，本书在网站上的 .NET 类别中所对应的列表为 `aspx` 和 `aspx_professional`。

通过以下步骤可订阅邮件列表：

1. 到 <http://p2p.wrox.com/>。
2. 从左边的菜单栏上选择正确的类别。
3. 单击要加入的邮件列表。
4. 按照订阅提示填写自己的电子邮件地址及密码。
5. 收到确认邮件后回信。
6. 使用订阅管理器加入更多的邮件列表并设置自己的电子邮件选项。

为什么该系统能提供最好的支持

因为您可以有选择地加入邮件列表，您还可以接收邮件列表的每周摘要。如果您没有时间或工具接收邮件列表的话，您可以搜索并下载我们的联机压缩文件。此外，您的电子邮件地址是受到 Lyris 系统保护的，任何垃圾邮件都将被自动删除。如果您在加入、离开邮件列表，或其他有关邮件列表等方面还有问题的话，请通过电子邮件与我们联系，地址是 listsupport@p2p.wrox.com。

目 录

第 1 章 .NET 移动天地	1
1.1 .NET	2
1.2 用 ASP.NET 优化 Web 应用程序	3
1.3 移动世界	5
1.4 小结	9
第 2 章 开发准备阶段	10
2.1 安装	10
2.1.1 Microsoft Mobile Internet Toolkit	10
2.1.2 Nokia Mobile Internet Toolkit	13
2.1.3 Openwave 仿真器	16
2.1.4 Microsoft Mobile Explorer	18
2.2 虚拟目录和 Web 应用程序	20
2.2.1 什么是虚拟目录	20
2.2.2 设置虚拟目录	20
2.2.3 IIS 中的权限	23
2.3 测试安装	24
2.4 出错帮助	26
2.5 小结	26
第 3 章 移动通信	27
3.1 可移动的重要性	27
3.1.1 移动更有利	27
3.1.2 移动业务	27
3.2 体验 WAP	28
3.3 WAP 协议	30
3.3.1 传输协议	31
3.3.2 表示格式	31
3.3.3 WAP 技术的成功之处	35
3.3.4 WAP 给我们的启示	36
3.4 小结	36



第 4 章 ASP.NET 移动窗体的结构	37
4.1 .NET Framework 基础知识	37
4.1.1 公共语言运行时	37
4.1.2 .NET 类库	38
4.2 移动窗体的生存期	39
4.2.1 移动窗体处理	42
4.2.2 移动窗体与 Web 窗体	44
4.3 Anatomy 101	45
4.3.1 Web 页	46
4.3.2 移动页	49
4.4 小结	52
第 5 章 构造移动 Web 窗体页	53
5.1 移动 Web 窗体的结构	53
5.1.1 <mobile:Form>	53
5.1.2 <mobile:Panel>	57
5.2 小结	60
第 6 章 与用户的交互作用	62
6.1 用户技巧	62
6.2 <mobile:Call>控件	63
6.3 <mobile:Label>控件	66
6.4 <mobile:TextBox>控件	69
6.5 <mobile:TextView>控件	74
6.6 其他问题	76
6.7 小结	76
第 7 章 导航一个移动站点	77
7.1 超链接	77
7.2 <mobile:Link>控件	78
7.2.1 使用 SoftKeyLabel 属性	81
7.2.2 利用<mobile:Link>控件创建一个菜单	83
7.2.3 给移动站点布线	87
7.3 <mobile:Command>控件	91
7.4 设备支持的图像	96
7.5 <mobile:Image>控件	97
7.6 小结	101

第 8 章 列表控件和数据绑定	103
8.1 <mobile:List>控件	103
8.2 与<mobile:List>控件的交互作用	105
8.3 <mobile:List>的导航与样式	107
8.4 <mobile:SelectionList>控件	111
8.5 小结	129
第 9 章 数据绑定	130
9.1 什么是数据绑定	130
9.2 小结	142
第 10 章 验证控件	143
10.1 验证控件如何工作	144
10.1.1 共同属性	145
10.1.2 特殊控件的属性	146
10.2 让验证控件工作	146
10.3 RangeValidator 控件	149
10.4 RegularExpressionValidator 控件	152
10.5 RequiredFieldValidator 控件	155
10.6 CustomValidator 控件	159
10.7 ValidationSummary 控件	164
10.8 小结	168
第 11 章 Rich 控件	169
11.1 使用 Rich 控件	169
11.1.1 AdRotator 控件	169
11.1.2 属性	170
11.1.3 Calendar 控件	176
11.2 小结	181
第 12 章 设计页面的输出样式	183
12.1 样式属性(attribute)	183
12.2 样式属性(property)	185
12.3 样式表单	187
12.4 设备的功能	190
12.5 设备的特殊处理	194
12.6 小结	196



第 13 章 编写控件	198
13.1 用户控件	198
13.2 定制控件	202
13.3 小结	207
第 14 章 新设备的支持	208
14.1 配置一台设备	209
14.2 请求头	212
14.3 定制控件适配器	213
14.4 小结	220
第 15 章 Web 服务	222
15.1 Web Services Refresher	222
15.2 Smart Devices	229
15.3 小结	230
第 16 章 安全机制、Cookies 和状态管理	231
16.1 .NET 的安全机制概述	231
16.1.1 代码访问的安全机制	233
16.1.2 基于角色的安全机制	234
16.2 Mobile Internet Toolkit 的安全机制	234
16.3 会话状态	248
16.3.1 移动页面中的 ViewState	248
16.3.2 会话过程中的 ViewState 历史信息	249
16.3.3 Cookieless 会话	249
16.3.4 其他注意事项	249
16.4 小结	250
第 17 章 Visual Studio.NET 介绍	251
17.1 安装	251
17.2 Start Page	252
17.2.1 Get Started	253
17.2.2 What's New	253
17.2.3 Online Community	253
17.2.4 Headline	253
17.2.5 Search Online	254
17.2.6 Downloads	254
17.2.7 Web Hosting	255

17.2.8 My Profile	255
17.3 创建项目	257
17.4 Designer 窗口	269
17.5 Toolbox 窗口	270
17.6 Solution Explorer 窗口	271
17.7 Properties 窗口	272
17.8 Dynamic Help 窗口	272
17.9 测试一个项目	273
17.9.1 Output 窗口	273
17.9.2 Task List 窗口	274
17.9.3 调试窗口	274
17.10 高级问题	280
17.10.1 Server Explorer	280
17.10.2 Object Browser	281
17.10.3 在 Visual Studio.NET 中设置选项	281
17.11 小结	283
第 18 章 使用 Visual Studio .NET 创建移动应用程序	284
18.1 简介	284
18.2 设计一个移动应用程序	294
18.3 添加逻辑	295
18.4 测试移动应用程序	307
18.5 小结	311
第 19 章 Maître D'移动应用程序	313
19.1 Maître D'移动应用程序概述	313
19.2 Maître D'	313
19.3 收集需求信息	314
19.4 体系结构	314
19.4.1 Maître D'数据库	314
19.4.2 Maître D' Web 服务	316
19.4.3 用户界面	325
19.5 小结	337
附录 配置环境变量	338

第1章 .NET 移动天地

移动应用程序开发历程虽然只有短短的几年，但这并不意味着它才刚刚起步。实际上，移动应用程序的开发已经普及了相当一段时间；经过这段时间，开发人员足以针对现存的缺陷及对未来的展望列出长长的清单。而且令人高兴的是，随着.NET 的出现，不仅让我们有可能已经实现了上述的某些愿望，或许还能够解决清单中的某些问题。现在，Microsoft 已经开发出了新的工具 Microsoft Mobile Internet Toolkit，它比其以.NET Framework 形式出现的早期产品更为先进。Microsoft Mobile Internet Toolkit 和 ASP.NET 是一个功能非常强大的工具包，通过它可以生成在任何设备上使用的内容。

本书的目的是，通过使用 ASP.NET 和 Microsoft Mobile Internet Toolkit 使您能够更轻松地进行移动应用程序开发。等到完成了本书的学习，您应该就可以自如地利用.NET 移动所提供的许多便利了。

本章主要内容：

- 传统的 ASP 开发人员不得不面临的一些挑战，以及 ASP.NET 和.NET Framework 是如何解决他们所面临的问题的。
- Microsoft Mobile Internet Toolkit 概述，并展示在移动仿真器中演示的一个非常简单的网页示例。
- ASP.NET 应用程序和移动 Web 控件之间的兼容性问题的探讨。

为了引出这些新技术的概念，您要先将思维退回到几年前的编程概念中。

随着商业的发展，我们开始脱离传统的客户-服务器/桌面应用程序模式，而开始考虑 Web 服务(ASP、JSP 等)方面的内容。在当时，这可能被视为发展的倒退，因为应用程序存在于 Internet 的方式与旧式主机体系结构非常相似——它们都是以服务器为中心，并用浏览器来代替那些原始的哑终端。而且更糟的是，用来编写 Web 应用程序的新语言跟 Visual Basic、Delphi 这些值得信赖的中坚语言相比非常的不理想。这些问题都给开发人员带来了一系列的挑战，下面就列举出了一些：

- 早期用于用户界面(User Interface, UI)开发的 HTML 控件非常简单。因此，开发人员必须提供控件自己的“布线”编码，来模拟使用动态 HTML(Dynamic HTML, DHTML)或 JavaScript 的事件(DHTML 用 JavaScript 来动态地改变一个 HTML 元素的属性，或者对它们所引发的事件作出反应)。为了尽量减少这种负担，一个事件模型必须符合最初为直接布局定制好的语言。更糟的是，输出要交付给客户端，而这个客户端一开始是计划传递静态网页，不是仿真桌面应用程序的。

- 由于 Internet 是由无国籍的协议组成的，因此状态管理也是一个大难题。这些协议是无关的，这意味着每次用户访问服务器都可能是第一次连接服务器。开发人员必须通过隐含变量或者将状态打包成一个 cookie，来将用户的状态从一个页面传递到另一个页面。然后在每次到



达服务器之后，必须重构视图状态。也就是说，我们必须保存大量的多余编码，用户的内存中只是保存以前(在客户-服务器模式下)的状态。

- 由于 Web 应用程序的分布式特性，要进行有效可靠的调试简直是不可能。
- 应用程序在不同系统间共享数据的共同基础不存在。
- 由于要将关键的应用程序移到 Internet 上去，因此保护应用程序和数据成为更大的问题。
- 由于表示代码和商务逻辑的错综复杂的关系，代码的重用受到限制。
- 更常见的是，因为浏览器软件以及客户端工具改变得非常快，开发人员在进行程序开发时总是要面临未知的客户端性能(甚至更多未知数)。

上述短短的几个障碍却可能引发更多的问题，那么“是否值得向这个方向发展呢？”。我相信是值得的。但是开发人员的工作并不仅仅是列出这类问题，而是要发掘出业界对软件功能提出了什么样的要求。商业的发展需要此领域的工人们可以使用库存清单、日程安排以及其他无数企业级数据。商业需要灵活性。

供应商正快速地开发工具和产品来协助补偿(或者说是安抚)那些开发人员，因为他们正忙于解决如何处理这些商业战场上的突变，与此同时，开发人员也已经提出了更具创新性的方法来处理新环境下的障碍。

1.1 .NET

.NET Framework 是 Microsoft 新开发的工具包，它是为 Internet 正在而且将会面临的开发挑战准备的。在 .NET Framework 的简介中，接着要介绍的是 ASP.NET。ASP.NET 承诺将完全改变 Web 应用程序的开发现状。

.NET 并不仅仅是对我们前面所提到的问题做补丁，而是用来填补传统 Windows 开发和 Web 应用程序开发之间技术空白的一个全新的平台，同时它又是可扩展的，这样足以满足新的计算设备和其语言持续增长的需求。

.NET Framework 能够给我们提供如下的一些优势：

- 方便地建立基于组件的应用程序的能力。
- 允许不同系统共享数据的 XML Web 服务。一个 XML Web 服务可以安全地展示其在 Internet 上的编程功能，并且允许客户端应用程序将它当成本地对象或 API 处理。Web 服务的一个例子就是 Microsoft's Favorites Service，它可以让任何能够在 HTTP/S 上进行简单对象访问协议(Simple Object Access Protocol, SOAP)调用的系统实现上述功能，这样用户就能将所喜欢的内容保存在自己的站点上。
- 公共语言运行时(Common Language Runtime, CLR):
 - 可以提高安全性的运行时环境，例如它能够保证系统运行的任何代码都是可信的。
 - 采用了新的垃圾收集机制的内存管理。新机制的优点之一就是它解决了循环引用的问题。
 - 根据运行框架的处理器器的特定类型，对本机代码进行即时(Just In Time, JIT)编译。该编译过程分为两步，而且允许在平台间进行方便的移植。编译器首先将本机代

码编译为 MS 中间语言(MS Intermediate Language, MSIL), 然后再编译成该处理器/操作系统所特有的机器代码。

- 支持多种语言(标准的 VB.NET、C#和 JScript.NET)。
- 与符合 CLR 的语言的互操作性:
 - 允许交叉语言继承的公共二进制系统。
 - 跨语言的异常处理。
- 更利于应用程序的部署和维护:
 - ASP.NET 应用程序的 XCOPY 开发, 这样就无需为应用程序的安装而生成注册表项了。
 - 由处理版本的系统级机制而产生的简便的代码修正。
- 与 COM 的无缝互操作性。
- ASP.NET, 还包括了:
 - Web 窗体服务器控件——通过这套控件, 您可以快速地开发出事件驱动的 Web 应用程序。而且, 使用 Visual Studio .NET 可以让 Web 窗体的开发人员有种 Visual Basic 窗体开发的感觉。
 - Web 服务——这样开发人员可以使用远程服务器的功能。
 - 浏览器的独立性和 UI 呈现。
 - 对状态管理的强化支持。
 - 后台编码机制——它能够使开发人员将 HTML 代码从他们的程序逻辑中分离出来。
 - 另外, 由于 ASP.NET 网页是由 JIT 编译器编译成本机代码的, 这样在强类型的程序设计环境中允许早期绑定。

虽然.NET 为 Web 应用程序的开发提供了如此奇妙的新平台, 但是我们以后还是不可避免地会遇到新的困难。这是科技发展的自然规律。随着时间的流逝, 开发人员不断地探索新技术, 又不断地超越它, 这样每个连续的“技术调整”清单就能够汇集成为一个平台或一项技术。

那么现在您已经作好了准备, 可以开始具体介绍这项有趣的技术了。接下来, 首先要复习一下 ASP.NET 的相关内容, 然后再介绍移动控件。

1.2 用 ASP.NET 优化 Web 应用程序

ASP.NET 是专门用来进行 Web 开发的工具。Microsoft 已经成功地模糊了桌面和 Internet 之间的差别。它们是 Visual Basic 设计范例的精华, 而且属于 ASP.NET 设计经验的一部分。这样, Windows 桌面程序员将能够利用许多现存的技术来建立 Web 应用程序。

下面将深入讨论 ASP.NET 服务器控件的有关细节, 因为它们与本书中要用到的移动 Web 窗体控件直接相关。而其他控件则超出本书讨论的范围, 但您可以查阅清华大学出版社出版的《ASP.NET 入门经典——C#编程篇》或《ASP.NET 高级编程》。



ASP.NET 服务器控件

ASP.NET 服务器控件是一组新类，它提供了从验证窗体到激活事件的种种功能。大量的服务器控件都是 ASP.NET 提供的，但是您要创建自己的能够在其他 ASP.NET 应用程序中重用的自定义控件也是非常容易的。下面，将首先简要介绍一下 ASP.NET 所提供的控件，然后再讨论 Microsoft Mobile Internet Toolkit (MMIT) 中的类似控件。

主要考虑如下几组控件：

- HTMLControls 和 WebControls
- 服务器端用户控件
- 验证控件

1. HTMLControls 和 WebControls

HTMLControls 和 WebControls 的确让开发人员的工作容易了许多。它们可以自动保持自己的状态(通过名为 postback 的属性，在 ASP.NET 中您也会碰到该属性)，还可以检测浏览器的类型并配合它们发送的 HTML，然后根据到服务器的返回结果，自动重建视图状态。因此，我们再也不必自己编制代码来重新填充控件了。

工作原理

HTML 服务器控件与 Web 服务器控件在网页中是作为对象存在的，因此在编程时，在服务器上也是以对象的形式存在，而且对它们的处理也要在服务器上进行(从而得名服务器控件)。这样就允许在窗体 POST 中检测客户端的事件，而且可以像它们是在桌面应用程序中产生的那样对之进行处理。因此，开发人员可以使用一个控件实例的属性和方法，所采用的方法与传统的 Visual Basic 窗体控件相同。

下面我们列出了几个因素，来帮助您决定是使用 HTML 服务器控件还是 Web 服务器控件。

- 如果使用的是传统 HTML 代码，而且需要添加一些附加功能，那么选择 HTML 服务器控件可以让您在利用事件驱动的程序设计模型时能够结合与客户端代码集成。请记住，因为所有的值都以字符串传递，这样您就不能利用类型对象模型。
- 如果您的用户接口设计包含了嵌套控件，而且您需要粒度更精细的事件处理，那么 Web 服务器控件对您来说就比 HTML 服务器控件更适合一些。这样您不仅能够在容器层捕获事件，而且可以获得类型安全的附加收益。
- 同时，上述两种控件都可以检测到客户端浏览器的类型，并实施合适的标记语言。

下面是 WebControls 类的一个示例：

```
<asp:Button runat="server" Text="Click" id="MyButton"></asp:Button>
```

以及 HTMLControls 类的示例：

```
<INPUT type=button value="Click" id="cmdMyButton" name="cmdMyButton" runat="server">
```

2. 用户控件

用户控件是创建可重用逻辑包的有效方法，您可以像使用 ActiveX 控件一样尽可能多地