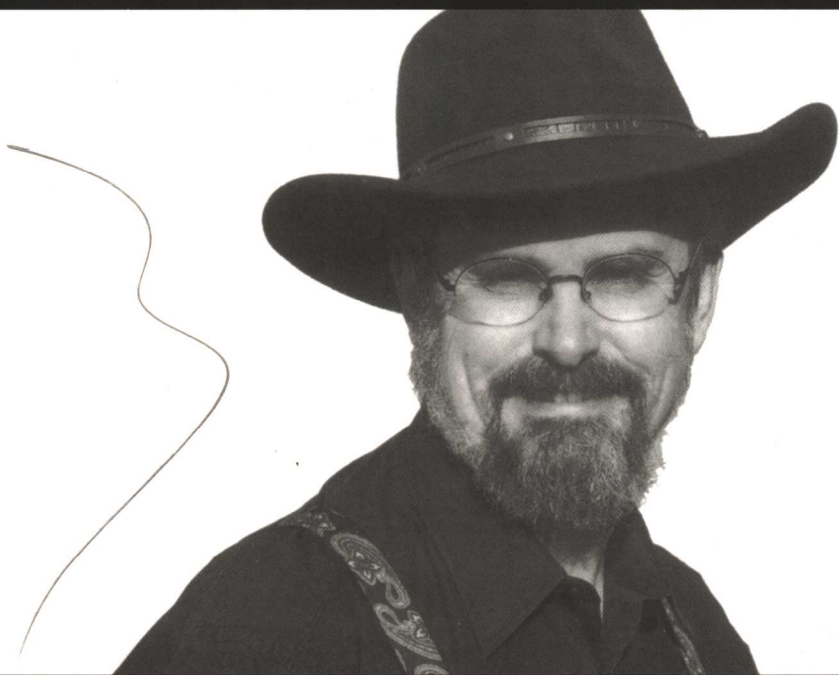


PROGRAMMER TO PROGRAMMER™



Professional Web Parts and Custom Controls
with ASP.NET 2.0

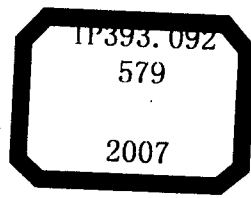
Web Parts 与自定义控件 高级编程

(ASP.NET 2.0 版)

(美) Peter Vogel 著
张建伟 王建东 译



清华大学出版社



Web Parts 与自定义 控件高级编程

(ASP.NET 2.0 版)

(美) Peter Vogel 著
 张建伟 王建东 译

清华大学出版社

北 京

Peter Vogel
Professional Web Parts and Custom Controls with ASP.NET 2.0
EISBN: 0-7645-7860-X
Copyright © 2006 by John Wiley & Sons, Inc.
All Rights Reserved. This translation published under license.

本书中文简体字版由 John Wiley & Sons, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2006-2447

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

Web Parts 与自定义控件高级编程(ASP.NET 2.0 版)/(美)沃其尔(Vogel, P.)著; 张建伟, 王建东译.

—北京: 清华大学出版社, 2007. 1

书名原文: Professional Web Parts and Custom Controls with ASP.NET 2.0

ISBN 978-7-302-14182-2

I. W… II. ①沃… ②张… ③王… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2006)第 138903 号

责任编辑: 王 军 徐燕萍

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 王秀菊

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编: 100084

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 北京市世界知识印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 26.5 字 数: 645 千字

版 次: 2007 年 1 月第 1 版 印 次: 2007 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 49.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 021929-01

前 言

自定义控件和 Web Part 可以为 Web 窗体创建可重用的(re-usable)组件,从而大大提高效率。我们可以打包一部分用户界面,在 Web 站点内部或许多不同的 Web 站点中重复使用它。但这只是这些控件提高效率的一个方面。如果有一个 ASP.NET 控件以稍有不同的方式工作,就可以创建符合我们需要的控件。在建立自己的自定义控件和 Web Part 时,可以让控件尽量完成用户所需完成的工作。

如果读者以前使用过 ASP.NET,就应很熟悉 ASP.NET 服务器控件。如果读者曾经尝试建立用户控件或自己的自定义控件,就会明白 ASP.NET 控件的内部工作原理。而 Web Part 是 ASP.NET 2.0 的新增部分,它提供的功能完全不同于 ASP.NET 控件。Web Part 可以为用户提供定制应用程序的功能,以满足用户的要求。使用 Web Part 可以让用户创建自己的界面,这是提高效率的另一种方式。

本书讨论如何高效地使用 Visual Studio 2005 创建自己的控件(也包括使用任意文本编辑器创建自定义控件、用户控件和 Web Part 等内容)。Visual Studio 2005 为创建 ASP.NET 应用程序和自定义控件提供了最佳支持。

本书采用的方法非常简单,即“如果需要,就编写代码;如果编写好代码,就会得到需要的功能”。全书的重点是创建控件所需的代码,所编写出的控件可以在建立 Web 应用程序时使用。本书不会列举像“计算小于 2000 的数的平方根”这样的代码。本书的代码完全基于与业务相关的应用程序。

也就是说,本书不会介绍 ASP.NET 的工作原理等背景知识,除非编写代码所需的基础知识与 ASP.NET 的工作原理有所不同。如果读者熟悉 ASP.NET 的工作原理,将有助于加深对本书内容的理解。另一方面,如果可以在不了解这些知识的情况下创建应用程序,那么也就是说,读者不需要通过这些知识来理解本书的示例。

达到目标并不只有一种方式,本书将介绍每种方式的优缺点,指出各种技术的常用场合,以便读者做出明智的选择。

读者可能并不需要了解本书的某些内容,至少刚开始时是不需要的。但在需要了解这些内容时,本书给出了许多示例(用 Visual Basic 2005 和 C#编写),以说明如何使用它们。例如,在第一次建立自定义控件时,可能并不需要在公司的外部发布该控件,所以不需要阅读有关许可的一节。但如果以后需要许可方面的知识,就可以在第 7 章的内容中找到它。

本书的读者

本书适合于有经验的 Web 开发人员。我们假定读者知道如何创建 Web 窗体,现在要利用 ASP.NET 的功能创建自己的控件。但我们并不会要求读者是有经验的对象开发人员。

如果读者在创建对象方面有一定的经验，本书将有一些章节可用于复习面向对象的基本概念，以及在 Visual Basic 2005 和 C# 中实现它们的方式。本书还假定读者不知道如何在 Visual Studio 2005 中创建对象。所以在第 3 章的开头介绍了这方面的知识，探讨了如何建立开发环境，以提高效率。

在语法上要注意：本书的许多示例代码都非常简明扼要——在示例中，三四行代码常常压缩为一行。本书的代码很容易理解，其作用也很明确，读者在阅读示例代码时不会感到迷惑。全书使用的命名约定(例如所有的变量名都有一个 3 字母前缀)也设计得一目了然。较新的命名约定不再使用这种样式(以前称为 Hungarian 命名法)，但在学习环境下，这种命名约定仍非常有益。即使不查看变量的声明，仍能看出 <strTextName> 是一个字符串。

另外，C# 和 Visual Basic 2005 代码示例使用相同的编程风格，以易于比较这两个版本。在大多数情况下，本书使用完整的对象名(例如使用 `System.Web.UI.WebControls.WebControl` 而不是 `WebControl`)，可确保代码按照预期的方式运行。虽然这有助于理解技术的工作原理，但使用 Visual Basic 2005 和 C# 提供的许多缩写方式，可以编写出更简洁的代码。

本书的结构

本书分为 4 部分。第 I 部分(第 1 章和第 2 章)介绍了自定义控件、用户控件和 Web Part 的基本概念。第 II 部分(第 3 章~第 7 章)介绍了实现这三种控件所需理解的知识。第 III 部分(第 8 章~第 11 章)说明了如何利用这三种控件的基本知识，添加业务特定的功能，建立专业化的控件，并讨论了其他高级主题。第 IV 部分包含第 12 章，这是自定义控件的一个案例分析。

下面列出了每一章的主要内容。

- 第 1 章“创建自定义控件”：本章描述了用户控件、自定义控件和 Web Part 的作用，还阐述了这三种控件的区别和使用场合。本章讨论了如何确定何时可以用这三种控件解决问题，以及如何选择正确的控件。
- 第 2 章“创建定制页面”：ASP.NET 2.0 的新增功能是创建定制页面，这需要使用 Web Part。本章将通过一个案例分析，说明如何在应用程序中使用定制页面，它将如何改变建立 Web 页面的过程。本章包含要与 Web Part 一起使用的 Web Part 架构中的组件。
- 第 3 章“创建自定义控件”：本章介绍了实现自定义控件的基础知识，陈述了如何创建给页面添加 HTML 代码的控件，如何创建合并其他 ASP.NET 控件的控件。本章还描述了如何在 Visual Studio 2005 中建立自定义控件项目。
- 第 4 章“建立用户控件”：如果读者知道如何建立 Web 窗体，就代表掌握了创建用户控件的大部分知识。本章以创建 Web 窗体的知识为基础，为 Web 应用程序创建用户控件。
- 第 5 章“创建 Web Part”：Web Part 是 ASP.NET 2.0 中的一个新增功能。本章介绍如何利用创建自定义控件的知识，创建一个 Web Part，如何给 Web Part 添加新功能，以扩展架构。本章还将讨论如何激活自定义控件上的属性定制功能。

- 第 6 章“用 ViewState 保存状态”：Web 应用程序中的一个关键问题是在用户对页面的请求之间跟踪信息。ASP.NET 2.0 提供了一些新功能，以管理控件中的状态。本章将介绍这方面的所有 ASP.NET 工具，包括 ASP.NET 以前版本中的工具和 ASP.NET 2.0 的新特性。
- 第 7 章“开发工具”：当然，建立控件仅是控件生存期的开头，我们还需要部署和更新它。本章将介绍如何以公开和私密的方式部署控件，如何在不重新部署应用程序的情况下更新控件。本章还会讨论如何编写管理 Web Part 的个性化系统。
- 第 8 章“添加业务功能”：建立了自定义控件后，就要给它插入应用程序的一些功能。本章将说明如何给自定义控件添加属性、方法和事件，并探讨在控件生存期的哪些阶段插入与应用程序相关的代码。
- 第 9 章“添加高级功能”：本章将介绍各种专业化主题，包括如何给定制用户添加客户端代码、建立自己的验证控件、创建数据绑定控件、给控件添加设计期间的支持等。
- 第 10 章“Web Part 之间的通信”：除了支持定制之外，Web Part 还有一个其他控件没有的特性，即在 Web Part 之间传送信息。本章将讨论如何创建可以相互交流的 Web Part，如何在页面的代码中管理这种通信。
- 第 11 章“使用 Web Part 体系架构”：Web Part 仅是支持定制页面架构的一部分。本章将跳出 Web Part 的圈子，描述可以在 Web Part 架构中使用的其他定制特性。
- 第 12 章“自定义控件的案例分析”：本章将前面章节的内容综合起来，建立一个完整的自定义控件/Web Part。前面的章节解释了自定义控件和 Web Part 技术的各个部分，本章则将它们综合起来，说明如何使用组成控件、可定制的属性 and Verb 菜单，建立一个自定义控件。除了这个案例分析之外，在 Wrox 网站 www.wrox.com 上还有一个案例分析，它描述了如何用模板编辑器创建数据绑定控件，本章的源代码也可以在 Wrox 网站上找到。

使用本书的要求

要运行本书中的示例，需要以下环境。

- .NET Framework 2.0
- Windows 2000 或更新版本上的 Internet Information Services (IIS)

本书使用的是 Visual Studio 2005，但读者可以用任意文本编辑器建立自定义控件。

用户支持

我们总是想知道您对本书的看法，您喜欢哪些内容，您不喜欢哪些内容，这些信息将有助于我们下一次做得更好。如果您有什么意见和建议，请向 fwkbook@tup.tsinghua.edu.cn 发邮件。但是，请您一定要在您的信中注明本书的书名。

如何下载本书的示例代码

在您登录到 Wrox 站点 <http://www.wrox.com/> 时, 只需使用 Search 工具或使用书名列表就可以找到本书。接着在 Code 栏中单击 Download 链接, 或单击本书信息页面上的 Download Code 链接, 就可以获得本书所有的源代码。另外, 您也可以从本书的合作站点 www.tupwk.com.cn/downpage 上下载本书的所有源代码。

从以上站点上下载的文件使用 WinZip 进行了压缩。在把文件保存到硬盘的一个文件夹中时, 需要使用解压缩软件(如 WinZip 或 PKUnzip)对该文件解压缩。在解压缩时, 代码常常放在各自的章节文件夹中。在开始解压缩过程时, 一定要将解压缩软件 WinZip 或 PKUnzip 设置为使用文件夹名。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误, 但是错误总是难免的, 如果您在本书中找到了错误, 例如拼写错误或代码错误, 请告诉我们, 我们将非常感激。通过勘误表, 可以让其他读者避免受挫, 当然, 这还有助于提供更高质量的信息。请给 fwkbook@tup.tsinghua.edu.cn 发电子邮件, 我们会检查您的信息, 如果是正确的, 就把它发送到该书的勘误表页面上, 或在本书的后续版本中采用。

要在网站上找到勘误表, 可以登录 <http://www.wrox.com>, 通过 Advanced Search 工具或书名列表查找本书, 然后在本书的信息页面上, 单击 Book Errata 链接。

E-Mail 支持

如果您希望直接就本书的问题向对本书知之甚多的专家咨询, 那么, 就向 support@wrox.com 发电子邮件, 在电子邮件的“主题”(Subject)栏中, 加上本书的名称和 ISBN 的最后 4 位号码。典型的电子邮件应该包括下列内容:

- 在“主题”栏加上书的名称、ISBN 的最后 4 位数字和问题所在的页码。
- 在邮件的正文中加上您的姓名、联系信息和问题。

我们不会发给您垃圾邮件。我们只需要详细的情况以节省您的宝贵时间和我们的时间。当您发送电子邮件时, 它会直接链接到以下支持链:

- 客户支持——您的消息会传送到我们的客户支持人员, 他们是阅读信息的第一人。他们有常见问题的文件, 会迅速回答一般性的问题。他们回答关于本书和网站的一般性问题。
- 编辑支持——更深的问题会转发到负责本书的技术编辑处。他(或)她具有编程或特殊产品的经验, 能够回答某个主题的详细技术问题。
- 作者支持——最后, 在编辑都不能回答问题的情况下(这种情况很少出现), 这些问题将转发到作者。我们试图保护作者不要从写作中分心, 但是, 我们也很愿意将特殊的问题转发给他们。所有的 Wrox 作者帮助支持他们的书籍。他们向客户和编辑回复电子邮件, 所有的读者都会从中受益。

Wrox 支持过程只能提供直接与已出版的图书相关的问题。对于超出此范围的问题可以通过 <http://p2p.wrox.com/> 论坛的团体列表来提供支持。

p2p.wrox.com

P2P 邮件列表是为作者和同行的讨论而设立的。我们在邮件列表、论坛和新闻组中提供“程序员到程序员的支持”(programmer to programmer support), 还包括一对一的电子邮件支持系统。如果把问题发送给 P2P, 就可以相信, 您的问题不仅是由支持专家解答, 而且还要提供给我们邮件列表中的许多 Wrox 作者和其他业界专家。在 p2p.wrox.com 上, 可以从许多不同的列表中获得帮助, 不仅在阅读本书时获得帮助, 还可以在开发应用程序时获得帮助。在网站的.NET 类别中, 最适合本书的是 beginning_vb 和 vb_dotnet 列表。

要订阅一个邮件列表, 可以遵循下面的步骤:

- (1) 进入 <http://p2p.wrox.com>。
- (2) 从左侧的菜单栏中选择合适的列表。
- (3) 单击想加入的邮件列表。
- (4) 按照指示订阅和填写电子邮件地址和密码。
- (5) 回复接收到的确认电子邮件。
- (6) 使用订阅管理器加入更多的列表, 设置自己的邮件设置。

为什么这个系统提供最好的支持

您可加入该邮件列表中, 也可以每周分类接收它们。如果您没有时间或设备接收该邮件列表, 可以搜索我们的在线文档。垃圾邮件和广告邮件会被删除, 您自己的电子邮件地址会被独特的 Lyris 系统保护起来。任何加入或退出列表的查询, 或者与列表相关的一般问题, 都应发送到 listsupport@p2p.wrox.com。

目 录

第 I 部分 控件简介

第 1 章 创建自己的控件	3
1.1 三种控件	3
1.1.1 用户控件	3
1.1.2 自定义控件	4
1.1.3 Web Part	5
1.2 可重用控件的优点	5
1.2.1 Web Part 除可重用性 之外的优势	6
1.2.2 用 Web Part 进行定制	7
1.3 用控件实现可重用性	8
1.4 控件	9
1.5 不同类型的控件	12
1.5.1 使用用户控件的场合	13
1.5.2 使用自定义控件的场合	13
1.6 Web Part: 定制	14
1.7 提供个性化	16
1.8 个性化框架	16
1.9 小结	18
第 2 章 创建定制页面	19
2.1 Web Part 框架控件	19
2.2 创建定制页面	20
2.3 给区域添加控件	23
2.3.1 在 Web Part 中封装控件	24
2.3.2 访问区域中的控件	24
2.4 配置区域	27
2.4.1 配置定制动词	27
2.4.2 配置 Verb 菜单	28
2.4.3 给区域设置样式	29
2.5 打开定制功能	30
2.6 定制工具控件	33

2.6.1 用 EditorZone 控件 编辑控件	33
2.6.2 用 CatalogZone 控件 添加新控件	36
2.6.3 配置工具区域	41
2.7 小结	42

第 II 部分 创建控件

第 3 章 创建自定义控件	45
3.1 创建 Visual Studio 解决方案	46
3.1.1 开始一个自定义控件项目	46
3.1.2 配置解决方案	47
3.1.3 保持同步	49
3.1.4 组织自定义控件	50
3.1.5 组织案例分析	51
3.2 扩展已有的控件	55
3.3 创建完整的自定义控件	56
3.3.1 添加组成控件	57
3.3.2 编写 HTML	65
3.3.3 合并控件和 HTML	73
3.3.4 分解 Render 方法	75
3.3.5 支持 AutoPostBack	76
3.4 处理返回的数据	80
3.5 控制相关的 HTML	83
3.5.1 设计期间的 HTML	83
3.5.2 运行期间的 HTML	85
3.6 提取和控制 WebControl 属性	89
3.6.1 服务器端属性	89
3.6.2 确定标记	89
3.7 管理自定义控件的样式	90
3.7.1 主题和 skin	91
3.7.2 使用层叠样式表类	91
3.7.3 更新样式	92

3.8	处理设计模式	99	5.6.5	组成控件的 name 和 id 特性	134
3.8.1	管理设计期间的代码	99	5.7	配置 Web Part	135
3.8.2	控制 Visual Studio 处理控件的方式	101	5.7.1	控制 Web Part 的用户界面	135
3.9	小结	102	5.7.2	查找 Web Part	138
第 4 章	建立用户控件	103	5.7.3	关闭个性化功能	139
4.1	用户控件、WebForm 和自定义控件	103	5.7.4	提供帮助	140
4.2	启用用户控件	104	5.8	小结	141
4.3	给控件编写代码	107	第 6 章	用 ViewState 保存状态	143
4.3.1	模拟构造函数	108	6.1	使用 ViewState	143
4.3.2	动态接口	109	6.1.1	直接访问 ViewState	144
4.3.3	使用 Web Part	110	6.1.2	高效地管理 ViewState	145
4.4	使用用户控件	111	6.2	管理控件的状态	148
4.5	访问控件	113	6.2.1	清理状态	152
4.6	用户控件 HTML	114	6.2.2	创建区分大小写的键	153
4.7	小结	114	6.3	与高速缓存集成	154
第 5 章	建立 Web Part	117	6.4	对象的高效串行化	157
5.1	激活定制属性的个性化功能	117	6.4.1	使用 TypeConverter	160
5.1.1	打开定制功能	118	6.4.2	把 TypeConverter 关联到类上	161
5.1.2	保存定制的内容	119	6.5	小结	162
5.1.3	共享定制的内容	119	第 7 章	开发工具	163
5.1.4	激活继承的属性的定制功能	120	7.1	在设计期间调试控件	163
5.1.5	给属性添加说明	121	7.2	部署控件	165
5.2	创建功能全面的 Web Part	121	7.2.1	已部署的 Web 站点格式	165
5.3	检查可个性化的数据	122	7.2.2	更新用户控件	168
5.4	访问特性	122	7.2.3	更新自定义控件和 Web Part	169
5.5	添加新的动词	124	7.2.4	共享自定义控件和 Web Part	170
5.5.1	创建动词列表	124	7.2.5	强名	173
5.5.2	共享例程	127	7.3	给控件设置许可	176
5.5.3	创建动词例程	128	7.4	管理个性化子系统	180
5.5.4	配置动词	129	7.4.1	识别用户	180
5.6	Web Part 的 HTML	130	7.4.2	建立个性化提供程序	183
5.6.1	顶层: WebPartZone	131	7.5	小结	184
5.6.2	中间层: Web Part	132			
5.6.3	底层: 标题栏	132			
5.6.4	底层 2: Web Part	134			

第 III 部分 扩展控件

第 8 章 添加业务功能..... 187

8.1 分解代码.....187

8.1.1 方法和事件..... 188

8.1.2 方法和属性..... 190

8.1.3 事件的作用..... 190

8.1.4 在 Web 环境下使用事件..... 192

8.2 自定义控件的生存期.....192

8.2.1 自定义控件的事件..... 193

8.2.2 主页的事件..... 193

8.2.3 组成控件的事件..... 193

8.2.4 处理事件..... 195

8.2.5 在设计期间运行事件..... 197

8.3 给组成控件添加代码.....197

8.3.1 创建例程..... 197

8.3.2 编写控件的例程..... 198

8.4 创建定制方法、属性和事件.....199

8.4.1 访问级别..... 199

8.4.2 定制属性..... 200

8.4.3 保存设计期间进行的修改..... 204

8.4.4 配置属性..... 205

8.4.5 共享/静态方法和属性..... 206

8.4.6 给 Visual Studio 2005 配置方法和属性..... 207

8.4.7 添加定制事件..... 208

8.4.8 为 Visual Studio 配置事件..... 218

8.4.9 管理多个控件中的事件..... 219

8.5 管理控件集合.....221

8.6 小结.....222

第 9 章 添加高级功能..... 225

9.1 集成客户端代码.....225

9.1.1 ASP.NET 客户端代码..... 225

9.1.2 静态和动态的客户端代码..... 226

9.1.3 引发客户端代码..... 227

9.1.4 动态生成代码..... 230

9.1.5 支持客户端脚本..... 233

9.1.6 建立回调函数..... 237

9.2 专用控件.....247

9.2.1 验证控件..... 248

9.2.2 模板控件..... 255

9.3 多个记录的数据绑定.....260

9.3.1 数据绑定属性..... 261

9.3.2 提取数据..... 263

9.3.3 显示数据..... 266

9.3.4 进行更新..... 267

9.4 设计期间的支持.....269

9.4.1 处理复杂的属性..... 269

9.4.2 设计器..... 288

9.5 小结.....295

第 10 章 Web Part 之间的通信.....297

10.1 使用可连接的 Web Part.....297

10.1.1 建立页面..... 297

10.1.2 建立连接..... 298

10.1.3 管理连接..... 301

10.2 创建可连接的 Web Part.....301

10.2.1 创建 Web Part 提供程序..... 301

10.2.2 创建 Web Part 使用程序..... 304

10.2.3 双向连接..... 306

10.2.4 预定义的接口..... 308

10.2.5 处理不兼容的接口..... 317

10.2.6 连接点参数..... 319

10.3 建立自己的连接.....320

10.3.1 连接的类型..... 321

10.3.2 通过代码建立连接..... 321

10.3.3 创建静态连接..... 325

10.3.4 永久管理和静态连接..... 326

10.3.5 连接事件..... 328

10.4 配置 ConnectionsZone 文本.....328

10.5 小结.....329

第 11 章 使用 Web Part 体系结构 331

- 11.1 在 WebPartManager 上设置个性化选项 331
 - 11.1.1 控制 WebPartManager 个性化选项 331
 - 11.1.2 把变化应用于其他用户 332
 - 11.1.3 进行授权 333
- 11.2 管理 Web Part 的个性化 335
 - 11.2.1 检查是否支持 DisplayMode 336
 - 11.2.2 管理个性化的改变 337
- 11.3 在代码中处理 Web Part 347
 - 11.3.1 关闭 Web Part 348
 - 11.3.2 移动 Web Part 348
 - 11.3.3 添加一个关闭的 Web Part 349
 - 11.3.4 从类别中添加 Web Part 349
 - 11.3.5 创建 Web Part 351
 - 11.3.6 导出导入 Web Part 353
- 11.4 小结 359

第 IV 部分 控件的应用

第 12 章 自定义控件的案例分析 363

- 12.1 设计控件 364
- 12.2 建立解决方案 364
- 12.3 添加控件 366
- 12.4 设置默认样式 371

- 12.5 控件的定位 372
 - 12.5.1 重写 Render 方法 373
 - 12.5.2 使用表格 374
 - 12.5.3 使用绝对定位功能 377
- 12.6 在显示模式和更新模式之间切换 380
- 12.7 给开发人员调整控件 384
- 12.8 保存状态 386
 - 12.8.1 为保存状态定义一个数据结构 386
 - 12.8.2 保存到 ControlState 中 387
- 12.9 提取用户数据 389
- 12.10 引发事件 393
 - 12.10.1 定义定制事件参数对象 394
 - 12.10.2 定义事件 396
 - 12.10.3 引发事件 396
 - 12.10.4 支持下一个控件开发人员 397
- 12.11 在回送时显示用户数据 399
- 12.12 显示和接受数据 401
- 12.13 支持定制 403
- 12.14 添加动词 405
- 12.15 通信 407
 - 12.15.1 定义接口 407
 - 12.15.2 实现提供程序 408
 - 12.15.3 实现使用程序 409
- 12.16 小结 410

第 I 部分

控件简介

第 1 章 创建自己的控件

第 2 章 创建定制页面

创建自己的控件

ASP.NET 带有服务器端自己的控件，为什么还要创建你自己的控件？为什么需要三种不同的控件(自定义控件、Web Part 和用户控件)？

创建自己的控件，可以为 Web 应用程序的用户界面建立强大的、可重用的可视化组件，包括允许用户定制和个性化应用程序的组件。除了用户控件之外，本章还将介绍两类主要的控件：自定义控件和 Web Part，并说明如何在创建控件的同时，提高 Web 应用程序的质量、提高开发效率、改善用户界面。

1.1 三种控件

为什么需要三种不同的控件？其部分原因是历史遗留下来的：自定义控件和用户控件在 ASP.NET 的第一版中被引入，而 Web Part 是 ASP.NET 2.0 的新增内容，它添加了用户控件和自定义控件所不能提供的功能。所以，从这种角度来看，Web Part 不同于用户控件和自定义控件，因为 Web Part 比较新，但这并不是最重要的区别。如后文所述，Web Part 实际上只是对自定义控件的一种扩展，但其新功能为开发人员提供了表达新东西的机会：让用户定制 Web 页面。只是这种说法对区分这三种控件没有什么实际帮助，Web Part 是一种特殊类型的控件，可以把用户控件和自定义控件用作 Web Part(只是它们不具备功能全面的 Web Part 的所有定制功能)。

Web Part、自定义控件和用户控件都允许创建可重用的组件，这些组件可以在 ASP.NET 中用于创建 Web 页面。在使用 ASP.NET 2.0 中的 Web Part、自定义控件和用户控件创建 Web 页面时，它们看起来非常类似。例如，它们都可以在设计视图中使用，可以把它们拖放到页面上，重新设置其大小，在属性窗口中设置其属性。主要区别是，自定义控件和 Web Part 位于 Visual Studio 的工具箱中，而用户控件位于 Solution Explorer 中。但因为用户控件和定制用户都可以用作 Web Part，所以可以从工具箱和 Solution Explorer 中拖放定制组件(功能全面的 Web Part 位于工具箱中)。无论是建立 Web Part、自定义控件还是用户控件，都可以给它们添加自己的属性、方法和事件。

1.1.1 用户控件

本书的重点是自定义控件和 Web Part，但不应忽略用户控件。对于开发人员来说，用户控件和自定义控件的主要区别是便于开发，所以强烈建议使用用户控件。用户控件为创

建重用控件提供了一种简单的方式：如果知道如何在 ASP.NET 中创建 Web 页面，就知道该如何创建用户控件。例如，给用户控件添加控件的方式与给 Web 页面添加控件的方式相同，都是把控件拖放到设计界面上。图 1-1 显示了 Visual Studio .NET 中设计视图上的一个用户控件。

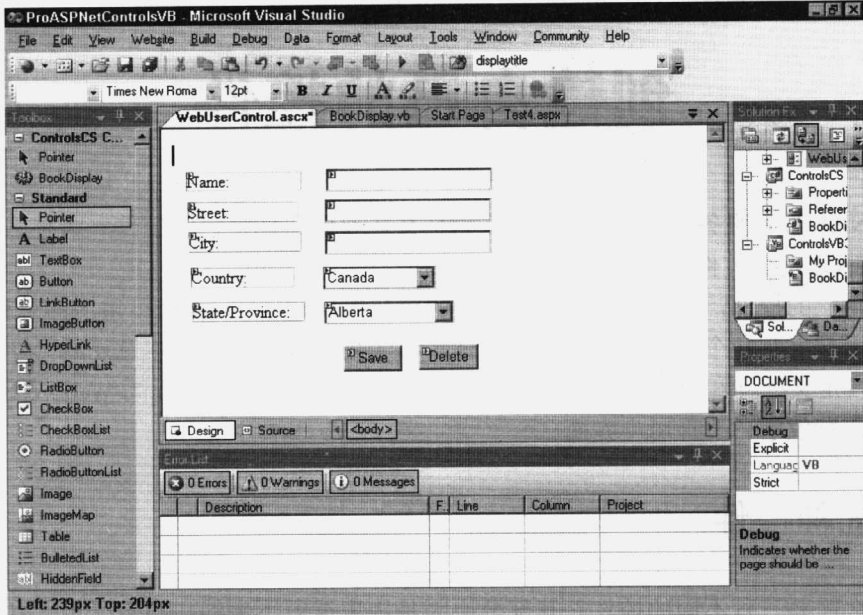


图 1-1

当然，这是有代价的：为了获得这种开发的便利性，用户控件有几个限制。在用户控件的功能上，第一个主要限制是它们不能继承其他 ASP.NET 控件，而自定义控件可以继承其他控件。继承其他控件的优点是，假如创建一个继承自 ASP.NET 控件 `ListBox` 的自定义控件，就可以扩展该控件。

但是，这种限制常常意味着，要以不同的方式考虑问题：如果要创建功能类似于列表框的用户控件，就可以在用户控件上拖放一个列表框，再添加需要的新方法、属性或事件(对于自定义控件，则必须自己完成所有的开发工作，而不能通过拖放设计器获得这些功能)。如本书后面所述，在自定义控件中可以利用的所有特性都可以在用户控件中使用，只是用户控件有拖放设计器，所以不需要用户自己添加它们。

用户控件的第二个主要限制是它们的可重用性：用户控件不能在项目或 Web 站点之间共享。用户控件只能在它所在项目的 Web 页面上使用，这个限制是无法克服的。

1.1.2 自定义控件

自定义控件是为 Web 应用程序创建可重用组件的一种代码密集型方式。例如，要给自定义控件添加控件，必须编写代码，创建该控件，再把它添加到自定义控件的 `Controls` 集合中，它没有用户控件的拖放功能。尽管创建自定义控件的时间较长，但其功能也比较强大。

自定义控件比用户控件更灵活。可以让所创建的自定义控件继承自另一个服务器端控件，再扩展该控件。例如，可以根据一个自定义控件创建另一个自定义控件(甚至可以创建

像 TreeView 控件那样复杂的控件), 再添加新的方法或属性, 以创建更强大的新控件。

自定义控件的重用性比用户控件高。自定义控件可以在项目之间共享。一般情况下, 可以在 Web 自定义控件库中创建自定义控件, Web 自定义控件库与应用程序是分开编译的, 之后, 就可以把该库添加到任意项目中, 在该项目中使用自定义控件了。

1.1.3 Web Part

将 Web Part 与用户控件和自定义控件进行比较并不正确。用户控件和自定义控件可用作 Web Part, 只是它们不具备功能全面的 Web Part 的所有特性。但如果要充分利用 Web Part 的功能集, 就必须从一开始就把控件建立一个 Web Part。如后面所述, Web Part 是自定义控件的扩展, 可以把功能全面的 Web Part 看作有超能力的自定义控件。

实际上, Web Part 最初并不是在 ASP.NET 中引入的, 而是在 Windows SharePoint Services 中引入的(2003 年, SharePoint 重新构建为运行在 ASP.NET 上, 此时引入了 Web Part)。SharePoint 是 Microsoft 的一个基于 Web 的工具, 用于创建基于文档的、用户可以定制的解决方案。在访问 SharePoint 站点时, 用户可以在 SharePoint 中建立页面, 给 SharePoint 页面添加 Web Part, 或修改页面上已有的 Web Part。在 ASP.NET 2.0 版本中, 添加了 Web Part 的一个版本。

对于大多数开发人员来说, “用户可以在 SharePoint 中创建页面”这句话有点自相矛盾。通常的分工是: 开发人员创建 Web 页面, 用户使用这些页面。而 SharePoint 设计为给用户授予建立页面的权限, 让用户在需要时建立页面, 而无须 IT 人员过问。在 ASP.NET 中, Web Part 可用于完成相同的功能: 让用户在需要时通过页面的可用 Web Part 清单来建立页面。有了这种功能, Web Part 开发人员就有了一种比较有趣的新工作, 他们不仅要建立应用程序, 还要建立组件, 允许用户建立应用程序。

提示:

上面描述 Web Part 在 SharePoint 站点上的使用方式时, 省略了重要一步。在创建 Web Part 后, 它会添加到 SharePoint 站点上的一个可用 Web Part 列表中。当 Web Part 可以在站点上使用, 开发人员就把该 Web Part 添加到页面上。只有 Web Part 在 SharePoint 站点的一个列表中时, 用户才能把它添加到 SharePoint 页面上。Web Part 的 ASP.NET 2.0 实现方式不采用这种 Web Part 列表方式。

注意:

Web Part 已成为 ASP.NET 开发人员工具集中的一个工具, 但 ASP.NET 2.0 目前开发的 Web Part 不能在 SharePoint 上使用, 为 SharePoint 建立的 Web Part 也不能在 SharePoint 外部使用。Microsoft 已升级了 SharePoint, 在不久的将来, 就可以把 ASP.NET 2.0 的 Web Part 用于 SharePoint 了。

1.2 可重用控件的优点

创建了自己的控件后, 就可以建立控件的工具集, 并在创建 Web 应用程序时使用它。