PROGRAMMER TO PROGRAMMER™

Fast Track ASP.NET C# Edition

高效掌握 ASP.NET

—— C# 编程篇

Marco Bellinaso
Brady Gaster
Kevin Hoffman 著 冉晓旻 译

清华大学出版社
http://www.tup.com.cn

高效掌握 ASP.NET

—— C#编程篇

Marco Bellinaso
Brady Gaster 著
Kevin Hoffman

冉晓旻 译

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：01-2002-3174

内 容 简 介

ASP.NET 是.NET 中一项优秀的新型 Web 开发技术，本书详细介绍了这项技术的主要内容，具体包括 ASP.NET 的新特性以及与.NET Framework 的集成、ASP.NET 应用程序的配置、ASP.NET 页面的工作原理和控件的使用、ASP.NET 页面与外部数据源的绑定以及如何利用身份验证和授权开发安全的 ASP.NET Web 应用程序和如何处理 Web 服务等，并重点讨论一些示例代码，以帮助读者快速掌握这项技术。

本书非常适合具备 Web 开发经验的 ASP 开发人员；另外，本书还适合那些具有一定服务器端 Web 开发知识、并且正在考虑把基于 Web 的应用程序迁移到 ASP.NET 的开发人员。

图书在版编目(CIP)数据

高效掌握 ASP.NET——C#编程篇/(美)贝利纳索等著；冉晓旻译.—北京：清华大学出版社，2002
书名原文：Fast Track ASP.NET— C# Edition
ISBN 7-302-06026-6
Ⅰ. 高… Ⅱ. ①贝…②冉… Ⅲ. ①主页制作—程序设计②C 语言—程序设计 Ⅳ.TP393.092

中国版本图书馆 CIP 数据核字(2002)第 083376 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
责任编辑：夏兆彦
印 刷 者：北京昌平环球印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：22　**字数**：522 千字
版　　次：2002 年 11 月第 1 版　2002 年 11 月第 1 次印刷
书　　号：ISBN 7-302-06026-6/TP・3593
印　　数：0001～4000
定　　价：43.00 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 是 Microsoft 公司的一项新技术，用于创建服务器端的 Web 应用程序。这项技术建立在 Microsoft .NET Framework 之上，这种新的开发平台在经过长时间的测试之后于 2002 年初正式发布。.NET Framework 只是 Microsoft 公司.NET 战略的一个组成部分，它通过把 XML 合并到每个应用程序中，从而改变建立应用程序的方式。ASP.NET 是.NET Framework 的一个重要组成部分，其作用是让我们能够开发 Web 应用程序和 Web 服务。在.NET 当中，关于创建 Web 应用程序的一些传统概念仍然被保留了下来：即每个应用程序都由 Web 页面组成，这些 Web 页面把信息显示在客户端；我们可以编写组件以封装程序设计逻辑；可以配置应用程序，以便它们根据具体不同的设置完成不同的工作；还可以重用代码。但是，在后台工作方面，ASP.NET 的工作方式与脚本技术(例如著名的 ASP 或 PHP)相比差别非常大，另外，它与 JSP 的工作方式差别也非常大。

ASP.NET 有许多功能特性，从而吸引 Web 开发人员在进行 Web 开发时采用它，这些特性包括：

- 完全面向对象的程序设计模型。ASP.NET 具有事件驱动和基于控件的体系结构，进而可以实现代码的封装和重新使用。
- 可以使用.NET 支持的任何语言(例如，Visual Basic .NET、C#、Jscript .NET 和许多其他的语言，当然那些语言必须由下载的第三方编译器把它们编译成为适合于 Common Language Runtime 的代码)编写代码的能力。
- ASP.NET 页面是根据需要被编译后执行的，而不是被解释执行的，因此 ASP.NET 页面的性能就有了很大的提高。此外，也可以对组件进行预编译，这样与 ASP 3 模型相比，组件的使用就相对容易一些。

在以后的章节中，我们将会详细地介绍这些改进之处。此外，我们还将引入 **Web 服务**的概念(Web 服务这个概念正在迅速成为 Web 开发的讨论热点)，Web 服务使用 SOAP 和 UDDI 等技术，通过 Web 提供应用程序逻辑。

本书主要内容

本书中各章的主要内容如下：

- 第 1 章：ASP.NET 简介。本章高度概括了 ASP.NET 是怎样集成到.NET

Framework 中的。

- 第 2 章：ASP.NET 应用程序。本章详细介绍了 ASP.NET 应用程序的组成，其中包括配置的详细内容、全局设置、以及 Web 服务和控件是怎样协同工作的。
- 第 3 章：ASP.NET 页面。本章详细介绍了 ASP.NET 页面的工作原理。在本章中，我们将会学习页面的处理过程、基类 Page 以及怎样在页面中使用控件。
- 第 4 章：使用 ASP.NET 控件进行程序设计。本章继续介绍基于控件的 ASP.NET 架构，同时继续介绍一些控件，那些控件可以为 ASP.NET 页面提供功能丰富的元素。此外，本章还将介绍怎样创建用户控件和定制服务器控件，以便把可重新使用的可视元素封装到随时可以重新使用的控件中。
- 第 5 章：数据绑定控件。本章将介绍开发人员处理数据时可以使用的控件，其中包括列表控件和数据网格控件。
- 第 6 章：数据源。本章将介绍数据访问方面的知识，即介绍怎样使用 ADO.NET 类连接数据和操作数据。在.NET 中，所有的数据访问都是在断开连接的情况下进行的，与 ADO 2.x 中“总是处于连接状态”特性相比，这样做可以减轻服务器的负担。
- 第 7 章：安全性。本章将讨论一些安全方面的理论，即怎样使用 ASP.NET 开发安全的 Web 应用程序，其中包括认证和授权方面的讨论。
- 第 8 章：Web 服务。本书的最后一章将介绍 ASP.NET Web 服务方面的知识，此外，还将讨论怎样创建和使用 Web 服务。

本书读者对象

本书的主要读者对象是具有一定服务器端 Web 开发知识、并且正在考虑把基于 Web 的应用程序开发迁移到 ASP.NET 的开发人员。创建功能完善的 ASP.NET Web 应用程序涉及到的知识很多，本书对此并没有全部介绍，但是在创建 ASP.NET 应用程序中所涉及到的重要主题，本书都将一一介绍。

使用本书所需要的条件

首要的条件是一台已经安装了.NET Framework 的计算机。为了运行 ASP.NET 页面，需要使用的操作系统为 Windows 2000 Professional 或更高版本，也可以使用 Windows XP Professional，并且在系统中至少要有.NET 的最小安装版本。目前，.NET 有以下两种版本：

- .NET Framework Redistributable——这是一个完整的框架，在这个框架中可以运

行任意的.NET 应用程序，安装这个框架大约需要 20MB 硬盘空间。

- .NET Framework SDK(Software Development Kit) ——这个框架除了具备上面框架的所有功能外，还包括一些示例和指南，以便在学习.NET 时进行参考。安装这个框架大约需要 130MB 空间。

这两个框架都可以从http://www.asp.net/免费下载。

虽然本书的代码确实是使用 Visual Studio 创建的，但是因为本书中的所有示例都可以使用一般的文本编辑器创建、使用命令行进行编译、以及使用 Web 浏览器进行查看，所以 Visual Studio.NET Professional 或更高版本，或者 Visual C#.NET 并不是必需的。MSDN 订阅者可以把 Visual C# .NET 和 Visual Studio .NET 作为 MSDN Professional 或更高订阅版本中的一部分来购买，当然，也可以单独在线或到商店购买它们。

客户支持

我们一贯重视读者的意见，并想知道每位读者对本书的看法，包括读者喜欢和不喜欢的内容，以及读者希望我们下一次完善的地方。您可以通过发送电子邮件(地址为 feedback@wrox.com)向我们反馈意见。请确保反馈信息提到本书的书名。

如何下载本书的示例代码

当您访问 Wrox 公司站点(地址为 http://www.wrox.com/)时，通过 Search 工具或书名列表，可以方便地定位需要的书目。然后，单击 Code 栏中的 Download 超链接，或者单击本书的详细页面中的 Download Code 超链接，就可以下载相应的范例代码。

当您单击下载本书中的代码时，将会看到带有以下 3 个选项的 Web 页面：

- 如果您是 Wrox Developer Community 的成员(即如果您已经在 ASPToday、C#Today 或者 Wroxbase 上注册)，就可以使用您通常的用户名和密码进行登录以下载代码。
- 如果不是它们的成员，则会向您询问是否愿意注册为会员，以便可以免费下载代码。此外，也可以从 Wrox Press 下载免费的文章。注册为会员后，可以得到本书升级版本和新版本的有关信息。
- 第三个选项是完全绕过注册过程，直接下载代码。

对于本书而言，不注册也能下载代码，但是，如果您愿意注册后下载代码，您的注册信息也不会泄漏给第三方。关于这方面的详细条款和条件，可以通过单击下载页面上的相关链接来查看。

从我们的站点上下载的文件都是使用 WinZip 压缩过的文档。保存文件到本地磁盘上的文件夹中后，需要使用一个解压缩程序(例如 WinZip 或 PKUnzip)来解压缩文件。在解压缩文件时，通常将代码解压缩到每一章所在的文件夹中。在解压缩的过程中，

应确保解压缩程序(WinZip、PKUnzip, 其他)被设置为使用原有文件夹名。

勘误表

我们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍然在所难免。如果您发现本书存在错误，例如拼写错误或不正确的代码段，请给我们反馈信息，我们将不胜感激。勘误表的发送可以节约其他读者学习本书的时间，而且能够帮助我们提供更高质量的信息。您的反馈信息将被核查，如果正确，将被粘贴到本书的勘误页面上，或者在本书的后续版本中使用。

要在我们的站点上找到勘误表，请访问 http://www.wrox.com/，并通过 Advanced Search 或者书名列表轻松定位本书页面。然后，单击 Book Errata 超链接即可，该链接位于本书的详细信息页面中的封面图形下面。

E-mail 支持

如果您希望直接向详细了解本书内容的专家咨询书中问题，可以发送电子邮件到 support@wrox.com，要求在邮件的主题栏中带上本书的书名和 ISBN(国际标准图书编号)的后 4 位数字。一个典型的电子邮件应包括下面的内容：

- 在主题栏中必须有本书的书名、ISBN 的后 4 位数字和问题所在的页数。
- 邮件正文中应包括读者的名字、联系信息和问题。

我们将不向您返回无用邮件，因为我们仅仅需要有用的详细资料，以便节约您和我们的时间。当您发送一个电子邮件信息时，它将得到以下环节的技术支持：

- 客户支持：首先，您的信息将被递送到我们的客户支持人员手中，并由他们阅读。对于一些被频繁提到的问题将被归档，并将立即回答有关本书或者 Web 站点的任何常见问题。
- 编辑支持：其次一些有深度的问题将被送到对本书负责的技术编辑手中，他们在程序设计语言或者特定的产品上有着丰富的经验，能够回答相关主题的详细技术问题。
- 作者支持：最后，如果编辑不能回答您的问题(这种情况很少发生)，他们将请求本书的作者。我们将尽量保护作者免受干扰，以便不影响其写作。然而，我们也非常高兴转寄给他们一些特殊的问题。所有 Wrox 公司的作者都为他们的书提供技术支持。作为回应，他们将发送电子邮件给用户和编辑，进而使所有的读者都受益。

Wrox 公司的支持过程仅仅对那些与我们出版的书目内容直接相关的问题提供支持，对于超出常规书目支持的问题，您可以从 http://p2p.wrox.com/forum 中的公共列表

中获得支持信息。

p2p.wrox.com 站点

为了便于作者和其他人讨论，特将讨论内容加入到 P2P 站点的邮件列表中，而且我们惟一的系统将 programmer to programmer™(从程序员到程序员而著)的编程理念与邮件列表、论坛、新闻组以及所有其他服务内容(一对一的邮件支持系统除外)相联系。如果您向 P2P 发送一个问题，应该相信它一定会被登录邮件列表的 Wrox 公司作者和其他相关专家所检查到。无论您是在阅读本书，还是在开发自己的应用程序，都可以在 p2p.wrox.com 站点中找到许多对自己有所帮助的邮件列表。

按照下面的步骤可以预订一个邮件列表：

(1) 登录 http://p2p.wrox.com/站点。

(2) 从左边的主菜单栏选择一个适当的类别。

(3) 单击希望加入的邮件列表。

(4) 按照说明订阅并填写自己的邮件地址和密码。

(5) 回复您收到的确认邮件。

(6) 使用预订管理程序加入更多的邮件列表并设置自己的邮件首选项。

本系统提供最好支持的原因

您可以连接到整个邮件列表，也可以只接收每周的邮件摘要。如果您没有时间和工具来接收邮件列表，可以直接查找我们的在线文档。独特的 Lyris 系统可以将一些没有用的垃圾邮件删除，并保护您的电子邮件地址不被侵扰。当存在加入和离开列表、以及任何有关列表的其他常见问题时，请发送邮件到 listsupport@p2p.wrox.com。

目　录

第 1 章　ASP.NET 简介 ……1
1.1　ASP.NET 简介 ……1
1.2　需要一个新版本的原因 ……2
1.2.1　像意大利面条一样混杂的代码 ……2
1.2.2　ASP 仅仅是脚本 ……4
1.2.3　Microsoft 的新思想和 COM 的死亡 ……5
1.3　关于 ASP.NET 的 7 个要点 ……5
1.3.1　ASP.NET 集成在.NET Framework 中 ……6
1.3.2　ASP.NET 是被编译的，而不是被解释的 ……8
1.3.3　ASP.NET 支持多种语言 ……9
1.3.4　ASP.NET 支持多种设备和多种浏览器 ……11
1.3.5　ASP.NET 是真正面向对象的语言 ……12
1.3.6　ASP.NET 是控件驱动的 ……13
1.3.7　ASP.NET 包含有自己的可移植配置和安全模型 ……15
1.4　ASP.NET 的工作原理 ……16
1.4.1　.NET Framework 对于分布式体系结构的意义 ……16
1.4.2　状态管理 ……20
1.4.3　生成输出结果 ……21
1.5　小结 ……26

第 2 章　ASP.NET 应用程序 ……27
2.1　ASP 应用程序和 ASP.NET 应用程序的比较 ……27
2.2　剖析 ASP.NET 应用程序 ……29
2.2.1　配置 ……30
2.2.2　Global.asax 文件 ……42
2.2.3　页面的配置标记 ……45
2.2.4　Web 服务 ……48
2.2.5　控件 ……51
2.2.6　.NET 组件 ……55
2.3　小结 ……58

第 3 章　ASP.NET 页面 ……60
3.1　页面的处理 …… 60
3.1.1　视图和状态管理 ……61
3.1.2　事件驱动 ……62
3.1.3　Web 窗体的处理步骤 ……63
3.2　Page 类 …… 69
3.2.1　Session 对象 ……70
3.2.2　Server 对象 ……70
3.2.3　Request 对象 ……71
3.2.4　Response 对象 ……72
3.2.5　User 对象 ……74
3.2.6　Application 对象 ……75
3.2.7　Cache 对象 ……75
3.2.8　Trace 对象 ……77
3.3　作为控件容器的页面 …… 78
3.4　小结 …… 81

第 4 章　使用 ASP.NET 控件进行程序设计 ……82
4.1　HTML 服务器控件 …… 83
4.1.1　HTML 服务器控件的类 ……84
4.1.2　HtmlControl 基类 ……86
4.1.3　HtmlContainerControl 基类 ……87
4.1.4　使用 HTML 服务器控件 ……87
4.2　ASP.NET Web Form 控件 …… 96
4.2.1　WebControl 基类 ……97
4.2.2　基本的 Web Form 控件类 ……98
4.2.3　ASP.NET 列表控件 ……105
4.2.4　ASP.NET 输入验证控件 ……110
4.2.5　ASP.NET Rich 控件 ……120
4.3　创建自定义控件 ……124
4.3.1　用户控件 ……125
4.3.2　自定义服务器控件 ……133
4.4　小结 ……139

第 5 章　数据绑定控件 ……141
5.1　数据绑定及其得到的支持 ……141

5.1.1 支持数据绑定的数据结构……142
5.1.2 支持重复绑定的控件……142
5.1.3 数据绑定的语法……143
5.2 支持重复值绑定的一些简单控件……145
5.3 Repeater 控件……150
5.3.1 DataBinder.Eval 方法……152
5.3.2 其他模板……153
5.3.3 Repeater 控件的事件……154
5.4 DataList 控件……158
5.4.1 项的选择……160
5.4.2 项的编辑……162
5.4.3 项的删除……167
5.4.4 动态地加载模板……167
5.5 DataGrid 控件……170
5.5.1 DataBound 列……171
5.5.2 行的排序……173
5.5.3 行的选择……175
5.5.4 分页显示记录……177
5.5.5 自动的分页显示……177
5.5.6 自定义的分页显示……180
5.5.7 使用模板的列……182
5.5.8 行的编辑和删除……184
5.5.9 高级定制……188
5.6 小结……190

第 6 章 数据源……191
6.1 基本的 ADO.NET 类……194
6.1.1 Connection 对象(SqlConnection 和 OleDbConnection)……194
6.1.2 Command 对象(SqlCommand 和 OleDbCommand)……195
6.1.3 DataReader 对象(SqlDataReader 和 OleDbDataReader)……195
6.1.4 DataAdapter 对象(SqlDataAdapter 和 OleDbDataAdapter)……196
6.1.5 DataSet 对象……197
6.1.6 DataTable 对象……199
6.2 通过 ADO.NET 访问和使用数据……201
6.2.1 连接数据存储……201
6.2.2 使用 Command 和 DataReader……202

6.2.3 DataSet 的使用 ……212
6.2.4 DataView 的使用 ……217
6.2.5 使用 DataTable 更新数据 ……220
6.2.6 更新数据库中的数据 ……228
6.3 处理 XML 数据 ……238
6.3.1 DataSet 的 XML 支持 ……238
6.3.2 编写 XML 文件 ……240
6.3.3 XML 的读取和导航 ……243
6.4 小结 ……254

第 7 章 保障 ASP.NET 应用程序安全 ……256
7.1 安全的含义 ……256
7.1.1 身份验证 ……257
7.1.2 授权 ……257
7.1.3 角色扮演 ……257
7.2 ASP 的安全性—— 一个历史教训 ……258
7.3 .NET 安全性本质 ……258
7.4 配置文件 ……270
7.5 ASP.NET 角色扮演 ……273
7.6 身份验证选项 ……276
7.6.1 Windows 身份验证 ……276
7.6.2 基于窗体的身份验证 ……278
7.6.3 Passport 身份验证 ……283
7.7 授权 ……285
7.7.1 HTTP 模块和自定义的身份验证 ……285
7.7.2 ASMX Web Services 中必需的题头 ……288
7.7.3 通过 web.config 文件禁止对确定文件的访问 ……291
7.8 小结 ……295

第 8 章 Web 服务 ……296
8.1 Web 服务概述 ……297
8.2 创建 ASP.NET Web 服务 ……299
8.2.1 编写 Web 服务 ……300
8.2.2 使用 Web 服务 ……307
8.3 Web 服务详述 ……316
8.3.1 WebMethod 属性 ……317

8.3.2　异步 Web 服务 ……323
8.3.3　确保 Web 服务安全 ……326
8.3.4　SOAP 头 ……327
8.4　小结 ……331

第1章 ASP.NET简介

10年前，Tim Berners-Lee第一次通过HTTP协议实现了数据传输。很快计算机科学工作者就接受了HTTP协议，并且国际教育机构宣称HTTP协议是传输计算机化信息的一种非常好的方式。HTTP的应用也扩展到了商业领域，不久就成为一个家喻户晓的名词。

为了最大限度地发挥HTTP协议的潜能，开发人员应该设计能够很容易地相互进行发现和集成的应用程序。为此，人们提出了一些标准，例如HTML和后来的XML等，目的是让人们可以在任何地方的任何类型的计算机操作系统上使用Web。

同样，软件厂商不但必须开发一些与Web集成的语言和程序设计工具，而且还必须开发完整的架构，以便为Web开发人员和企业开发人员提供一些容易构建、开发和部署应用程序所需要的工具。直到最近，由于采用了XML作为数据传输和描述的标准，以及使用Web服务范型的人的迅速增多，Web的潜力才真正得到挖掘。

这种需求引发许多大的软件厂商(例如IBM、Sun Microsystems和Microsoft等)开发了许多产品，以满足Web开发人员和企业开发人员的需要。其中Microsoft公司推出的产品为.NET Framework，在这个软件开发模型中，Microsoft公司进行了重大改进，创建了一个集成化的组件套件，把Web的基础(即标识语言和HTTP协议)和面向对象的编程思想结合了起来。

本书将把ASP.NET和ASP.NET Web服务作为.NET Framework的一个子集来介绍，此外，本书还将介绍这些技术怎样使World Wide Web实现了真正意义上的交互性和平台无关性。

1.1 ASP.NET简介

以前的基于服务器的Web应用程序架构主要依赖于脚本语言或晦涩的专用标记法约定去提供对基于服务器资源和组件的访问。大多数Web开发范型通过把解释型或专用型脚本集成到那些运行在服务器上的应用程序和组件中而进行访问。ASP.NET之前的Web开发架构有以下两类：

- 服务器端资源解释型脚本
- 通过服务器端调用来执行的独立小应用程序

传统ASP和ColdFusion程序员大多数使用的是第一类Web开发架构。如果您对传

统 ASP Web 开发比较熟悉，就会理解脚本化应用程序的执行速度通常比被编译的应用程序的执行速度要慢许多。此外，脚本化的平台也带来了其他一些问题，例如，与安全设置的集成能力比较弱(或者不具备那样的能力)、以及资源的利用效率不高等。

Perl 和 CGI 程序员广泛使用第二类 Web 开发架构，第二个架构的问题与第一个架构的问题完全不同。虽然使用 Perl 和 CGI 编写的应用程序的执行速度比对应的脚本化应用程序的执行速度快一些，但是它们需要很大的内存空间(原因是如果多个客户对服务器发出请求，则需要应用程序的多个实例)，并且它们的编写和调试通常比较困难，最重要的是没有集成的概念，考虑的仅仅是可移植性。

但是，ASP.NET 不属于上面两类架构的任何一类，因为它提供了真正集成化的 Web 应用程序范型。ASP.NET 并不是上面两类架构的简单发展，而是打破了 Web 应用程序开发的发展趋势，它是第一种集成基础性架构的 Web 应用开发模型。此外，ASP.NET 也不是.NET Framework 的扩展和修正，通过松散耦合的挂接使用.NET Framework 提供的功能，而是架构本身的一个子集，是.NET 运行时管理的.NET Framework 的一个实实在在的组成部分。从本质上来讲，由于 ASP.NET 把原来桌面开发人员所垄断的工具和技术扩充到了 Web 开发领域中，因此，它打破了传统应用程序开发和 Web 开发之间曾经存在的界限。

1.2 需要一个新版本的原因

对于许多传统 ASP 程序员而言，ASP.NET 的变化使他们在开发 Web 应用程序时要适应新的范型。在使用 Microsoft 公司的技术开发 Web 应用程序时，传统 ASP 是一个相当优秀的工具。但是，对于其他大多数的开发模型而言，传统 ASP 有一些基础性的设计问题，这些问题使 Web 应用程序的开发面临许多困难。

1.2.1 像意大利面条一样混杂的代码

如果您已经用 ASP 创建过许多应用程序，就会非常熟悉长的 Web 页面，其中，服务器端脚本代码是与 HTML 混杂在一起的。通常，在一个特定的 Web 页面内，Web 页面要在 ASP 和 HTML/客户端脚本之间来回转换许多次。其结果是在一次响应中，为了提供所有的服务器代码，运行在所有 IIS 服务器上的 ASP ISAPI 筛选器需要开关许多次。

下面的示例从数据库中查询数据，并把数据赋予 HTML 下拉式菜单：

```
<%
Set dbConn = Server.CreateObject("ADODB.Connection")
```

```
Set rs = Server.CreateObject("ADODB.Recordset")
dbConn.Open "Provider=SQLOLEDB.1;UID=sa;PWD=pass4Sql;Server=bglaptop;Initial
Catalog=pubs"
%>

<select name="cboAuthors">
<%
rs.Open "SELECT * FROM Authors",dbConn,3,3
do while not rs.EOF
%>
<option value="<%=rs("au_id")%>"><%=rs("au_lname") & ", " & rs("au_fname")%></option>
<%
rs.movenext
loop
%>
</select>
```

这 16 行代码就可以生成一个简单的 HTML 控件。这种编写代码的方法降低了应用程序的性能，因为运行在服务器上的脚本引擎必须在脚本和 HTML 代码之间来回转换。如果 ASP 页面把大量的报文发送给客户端的浏览器，而此 ASP 页却未启用 Response.Buffer 属性，则将给客户端浏览器发送大量小的分组，这样将导致性能更低。

说明：

关于优化 ASP 应用程序的详细信息，请在 MSDN 中参阅标题为“25+ ASP Tips to Improve Performance and Style”的文章；在 Microsoft MSDN 的站点上，文章存放的地址为：http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnasp/html/ asptips.asp。

如果 Web 开发小组中的图形设计人员需要回头在其中一个 WYSIWYG(所见即所得)编辑器中对代码进行改动，最好的情况是保留有代码的备份。但是，大多数 HTML 编辑器不能识别 ASP 代码，这样，HTML 编辑器就会完全忽略服务器端的代码。

使用这些结构编写的代码很容易导致无法控制的局面。并且，当创建的 COM 对象与 ASP 脚本一起使用时，情况可能会更加复杂，管理负担将更重，因为管理人员必须解释在为复杂 ASP 脚本所创建的组件中发生了什么。ASP.NET 页面是按照传统的面向对象思想编写的，因此，页面本身就包含了控件，控件的编写与桌面应用程序的编写相似。这样就消除了传统 HTML 标记与页面内代码结合之间的依赖性，大大减少了“像意大利面条一样混杂的代码的现象”。如果选择使用后台编码的方法，程序设计的逻辑和表示实际上就分为两个不同的文件，以后对程序进行维护时，就不需要在一个地