求是科技

ASP.NET

通用模块及典型系统开发

● 求是科技
张　蓓　编著

实例导航

附源代码光盘
CD-ROM

人民邮电出版社
POSTS & TELECOM PRESS

ASP.NET

通用模块及典型系统开发

● 求是科技
张　蓓　编著

实例导航

人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目（CIP）数据

ASP.NET 通用模块及典型系统开发实例导航 / 求是科技编著.
—北京：人民邮电出版社，2006.2

ISBN 7-115-14328-5

Ⅰ. A… Ⅱ. 求… 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 006968 号

内 容 提 要

本书先用两个章节简要介绍了用 ASP.NET 进行编程时要掌握的基础知识，然后分别介绍了实际开发中几个常用模块的开发方法和技巧，这些模块包括登录模块、注册模块、网页计数器模块、讨论区 BBS 模块、投票系统模块、聊天室模块、新闻自动发布系统模块、搜索引擎和网上书店系统。

书中在对每一个系统的设计时都以系统的易用性、安全性、健壮性、高效性、可维护性和可扩展性为原则，先实现最简单的功能，然后逐步提高，以达到逐步求精的目的。

读者可以将书中所介绍的很多常用模块的实现与自己所开发的相关系统进行对比，找到需要改进的地方。也可将书中介绍的实例直接应用到实际项目中。

本书适合 ASP.NET 初、中级程序员在进行系统开发时参考和阅读。

ASP.NET 通用模块及典型系统开发实例导航

◆ 编　　著　求是科技　张　蓓
　责任编辑　张立科

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
　邮编　100061　　电子函件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京顺义振华印刷厂印刷
　新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
　印张：24.5
　字数：597 千字　　2006 年 2 月第 1 版
　印数：1 – 6 000 册　　2006 年 2 月北京第 1 次印刷

ISBN 7-115-14328-5/TP・5186

定价：42.00 元（附光盘）

读者服务热线：(010)67132692　印装质量热线：(010)67129223

前　言

ASP.NET 与 ASP 有着本质的不同。ASP.NET 完全基于模块与组件，具有更好的可扩展性与可定制性，数据处理方面更是引入了许多新技术，正是这些具有革新意义的新特性，让 ASP.NET 远远超越了 ASP，同时也提供给 Web 开发人员更好的灵活性，有效缩短了 Web 应用程序的开发周期。

本书为想使用 ASP.NET 进行开发的读者而编写，由浅入深地介绍了用 ASP.NET 开发各个常用模块的方法和技巧。

本书分为 11 章，具体安排如下。

第 1 章：介绍了 ASP.NET 的基础知识，包括 Web 窗体和 Web 应用程序等基本的概念，通过这一章的学习，读者可以对 ASP.NET 有一个初步的了解，并可以编写简单的 ASP Web 应用程序。

第 2 章：介绍了.NET 中的数据访问技术——ADO.NET，并讨论了数据访问中的一些高级话题。通过这一章的学习读者可以使用 ASP.NET 实现基本的数据访问功能，并了解在.NET 中使用连接池、数据缓存和事务管理等方面的基本方法和原则。

第 3 章和第 4 章：分别介绍登录和注册这两个任何电子商务网站都要用到的基本功能模块。首先从最简单的设计开始建立一个可以运行的系统，然后逐步完善，逐步优化，逐步求精。这样，读者比较容易理解，同时这也是软件开发的一个自然的过程。本书的后续章节会多次使用这两章的结果。

第 5 章：介绍网页计数器的设计方法，实现了基于 Application、数据库和 XML 三种方式的网页计数器，并且使用.NET 的反射技术实现了网页计数器类型的动态配置。

第 6 章：介绍一个简单的 BBS 讨论区系统的开发。这是本书第一个完整的应用系统，在设计实现上采用了典型的三层应用程序模型，可以更好地分离界面和业务逻辑的编程。这个系统比较复杂，其中大量使用了存储过程。除此还介绍了对 ASP.NET 应用程序进行性能优化的基本方法。总体上讲本章的内容比较丰富，技巧性也比较强，通过本章的学习，读者将深入了解 ASP.NET 编程技术。

第 7 章～第 11 章：这些章节中介绍的系统来自于真实的网络应用。其中第 7 章为网上投票系统，第 8 章为聊天室系统，第 9 章为新闻发布系统，第 10 章为搜索引擎的实现，第 11 章为网上书店系统。这些章节中不仅讲解了系统实现的过程，而且每一章都会重点介绍一项 ASP.NET 编程的关键技术。第 7 章是一个 DataGrid 控件全面应用的范例；第 8 章使用了现在非常热门的 AJAX 技术实现无刷新的动态页面；第 9 章介绍了如何使用 Web Service 技术对业务逻辑进行封装；第 10 章的核心内容为搜索引擎的基本实现模型；第 11 章是本书规模最大的例子，详细介绍了系统的设计、实现和部署的过程，第 11 章中还使用了微软公司提供的企业库模块，也是一个企业库模块在多层系统中应用的范例。

本书主要由施伟伟编写，特别感谢张蓓在本书的资料收集和内容编排上给予的大力协助。

由于时间仓促和作者的水平有限，书中错误和不妥之处在所难免，敬请读者批评指正。如果有任何问题，可以发送 E-mail 到 shiweiwei97@gmail.com，作者将尽快予以答复。

编者

2006 年 1 月

目　录

第 1 章　ASP.NET 基础

ASP.NET 是建立在公共语言运行库上的 Web 编程框架，相对于它的前身 Active Server Pages（ASP）而言，ASP.NET 提供了更强的性能、更方便的工具支持、更好的平台支持和灵活性。与以往的 Web 编程框架相比，ASP.NET 的一大革命性进步是可以将应用程序逻辑与表示代码清楚地分开，用类似于 Visual Basic 的简单窗体处理模型处理事件。这样一来 Web 应用程序的开发人员可以使用和 Windows 桌面程序开发类似的编程模型，从而大大降低了开发难度。

本章将首先介绍 ASP.NET 的开发环境，然后通过创建一个基于 ASP.NET 的 Web 应用程序介绍创建应用程序的方法，第 3、4 节将分别介绍 ASP.NET 中两个重要的概念：Web 窗体和 Web 应用程序，最后第 5 节中给出了一些 ASP.NET 编程常用的资源，供读者参考。

1.1　ASP.NET 开发环境

运行 ASP.NET 应用程序的基本要求是 IIS 服务器和.NET 框架，当然也可以在其他 Web 服务器（如 Apache）上运行 ASP.NET 应用程序，但是 IIS 服务器对于 ASP.NET 的支持是最好的。大多数 Web 应用程序都会涉及数据库的访问，这就要求服务器上必须安装微软数据访问组件（MDAC），读者可以从微软公司的主页上下载该组件。以上所讲的是运行 ASP.NET 应用程序的基本条件，要开发基于 ASP.NET 的应用，还必须安装.NET 框架的软件开发包，即.NET Framework SDK。

1. .NET Framework SDK

.NET Framework SDK 提供了开发.NET 应用所必需的所有内容。SDK 附带了用于各种语言的命令行编译器、公共语言运行库和相关文件以及详细的文档。.NET Framework 的安装文件可以从微软网站上获取。安装完成以后，在 SDK 安装路径的 Bin 子目录下，提供了很多.NET Framework 工具，按照它们功能的范畴可以分为配置和部署工具、调试工具、安全工具以及常规工具 4 类。Bin 子目录下的 StartTools.htm 页面对这些工具的作用进行了简单的介绍，需要更详细内容的读者可以参考 MSDN。

2. Visual Studio .NET

安装了.NET Framework SDK 后就已经可以进行.NET 的开发了，但是此时还存在很多不方便的地方，比如：用普通文本编辑器编写代码效率比较低，而且容易出错。Visual Studio .NET 是微软公司推出的针对.NET 的集成开发环境（IDE），它集成了大量的工具箱和设计器，可以进行所见即所得（WYSIWYG）的编辑、拖放服务器控件和自动部署等；它提供了很多新建项目模板，可以帮助开发人员快速建立应用程序的框架；此外，它还可以通过外挂插件的形式扩展新功能。安装 Visual Studio .NET 的过程非常简单，按照安装盘的提示进行即可，大致可以分为 3 步：首先是安装系统必备的组件，包括.NET Framework 等；第 2 步是安装 Visual Studio .NET 的组件，读者可以根据需要在需要安装的组件列表中进行选择；最后是安装

MSDN 文档，使用过微软开发工具的读者都了解，MSDN 文档十分详尽，是进行.NET 开发的必备工具，建议读者在空间允许的情况下进行完全安装。

3. 其他常用辅助工具介绍

Visual Studio .NET 的功能虽然比较强大，但是还是有欠缺之处，如单元测试、重构、代码生成、代码规范检查等。常见的.NET 开发辅助工具如下。

NUnit：单元测试工具。

NDoc：文档生成工具。

NAnt：构建工具。

Refractor：重构工具。

FxCorp：代码规范检查工具。

CodeSmith：基于模板的代码生成工具。

The Regulator：正则表达式工具。

Nprof：性能分析工具。

使用这些工具作为 Visual Studio .NET 的补充（其中部分工具可以通过插件的形式集成到 Visual Studio .NET 环境中去），可以大大提高编码的速度和质量。有关这些工具的使用方法，可以参考相关的网站。

1.2 ASP.NET 及 C#语法简介

ASP.NET 是 Web 编程的框架，虽然它的编程模型和 Windows 窗体（Winform）程序非常类似，但在本质上还是有所区别的：Winform 运行在本机，它的事件响应和程序状态的维护是非常方便的；而 Web 页则是在服务器和客户端之间进行交互的，它有一定的生存周期，要维护页面上的各种状态信息，必须通过一定的方式在提交表单时保存这些信息。以往这个工作必须编写相应的代码实现，ASP.NET 提出了 Postback 机制来解决这个问题，该机制利用页面的视图状态（ViewState）保存页面中控件的状态，开发人员只需将页面或者控件的视图状态启用即可。要深入理解这些概念，需要在实际应用中仔细体会 ASP.NET 页面的运行机制，必要的时候可以利用一些.NET 的反编译工具查看.NET Framework 的源代码，这样不仅对 ASP.NET 编程有帮助，对设计理念的提升也是大有裨益的。

.NET Framework 为 ASP.NET 应用程序提供了强大的类库支持，目前 ASP.NET 支持 3 种开发语言 C#、VB.NET 和 Jscript.NET，这 3 种语言编译的程序集（Assembly）是相同的，各种不同语言编写的程序集是可以相互调用的。C#语言是.NET 平台首推的开发语言，吸取了 C/C++和 Java 语言的优点，读者只要有 C/C++或 Java 其中一种语言的编程经验，理解 C#的语法是非常简单的。本书的例子都是使用 C#语言编写的。

C#语言的基本语法如变量声明、条件判断、循环等都与其他语言类似，这里就不具体叙述了，例 1-1 所示为一些 C#语言中特有语法的示例代码，供读者参考。

【例 1-1】C#示例代码

```
//声明简单属性
public string name
{
```

```
        get {return ...;}
        set {... = value;}
    }

    //声明索引属性
    public String this[string name]
    {
        get {return (string) lookuptable[name];}
    }

    //自定义属性
    [STAThread]
    [Obsolete("Obsolete message here")]
    [Obsolete("Obsolete message here", true)]

    //声明事件
    public event EventHandler MyEvent;
    ......
    protected void OnMyEvent(EventArgs e)
    {
        MyEvent(this, e);
    }

    //向事件添加事件处理程序或从事件移除事件处理程序
    Control.Change += new EventHandler(this.ChangeEventHandler);
    Control.Change -= new EventHandler(this.ChangeEventHandler);
```

代码中涉及到的一些概念，如自定义属性、事件等对初学者来说可能不太容易理解，没有关系，在后续的章节中将结合具体的实例讲解这些概念。

1.3　第一个 ASP.NET 应用实例

下面我们用 Visual Studio .NET 创建一个最简单的 ASP.NET Web 应用程序。首先需要启动 IIS 服务器的 WWW 服务，如图 1-1 所示。打开 Visual Studio .NET，选择“文件”→“新建”→“项目”菜单，弹出如图 1-2 所示的对话框，在“位置”输入框中输入“http://localhost/book01”，单击“确定”按钮。

Remote Access Connection Manager	创建网络连接。	已启动	手动	本地系统
Telephony	提供 TAPI 的支持，...	已启动	手动	本地系统
Terminal Services	允许多位用户连接并...	已启动	手动	本地系统
World Wide Web Publishing	通过 Internet 信息...	已启动	手动	本地系统
Alerter	通知所选用户和计算...		已禁用	本地服务
ClipBook	启用“剪贴簿查看器...		已禁用	本地系统
Error Reporting Service	服务和应用程序在非...		已禁用	本地系统
Human Interface Device Access	启用对智能界面设备 ...		已禁用	本地系统
Messenger	传输客户端和服务器...		已禁用	本地系统
Network DDE	为在同一台计算机或...		已禁用	本地系统
Network DDE DSDM	管理动态数据交换 (D...		已禁用	本地系统
Remote Registry	使远程用户能修改此...		已禁用	本地服务
Routing and Remote Access	在局域网以及广域网...		已禁用	本地系统
Security Center	监视系统安全设置和...		已禁用	本地系统
Telnet	允许远程用户登录到...		已禁用	本地系统

图 1-1　启动 IIS 的 WWW 服务

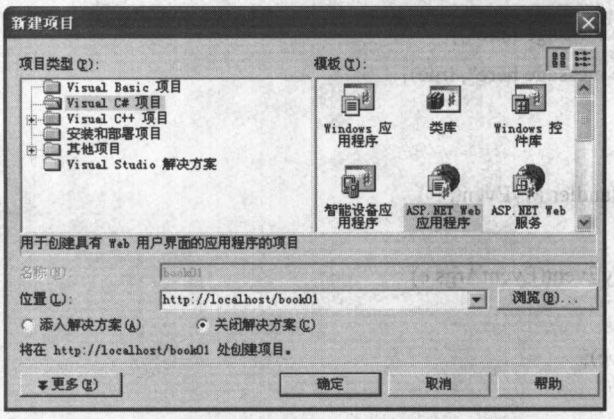

图 1-2　新建 ASP.NET Web 应用程序项目

在“解决方案管理器”中出现了“book01”节点，在该节点上单击鼠标右键，选择“添加”→“添加 Web 窗体”项，如图 1-3 所示，在弹出的“添加新项”对话框的“名称”一栏中输入“Hello.aspx”，单击“打开”按钮打开 Web 窗体。

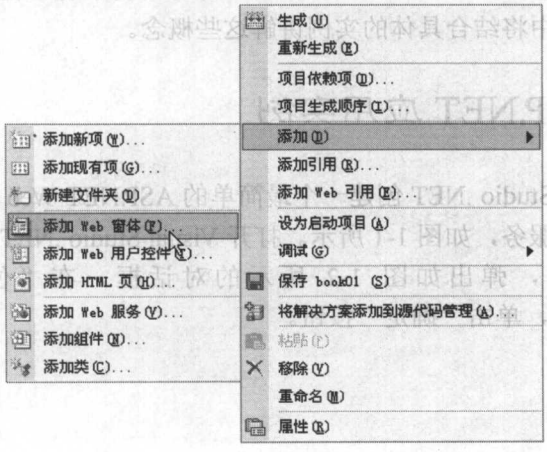

图 1-3　新建 Web 窗体

从工具箱中分别拖拽 TextBox、Button 和 Label 控件至 Web 窗体中，调整控件布局和属性，如图 1-4 所示（3 个控件的 id 分别为 tbName、btnClick 和 lblHello）。双击 Button 控件，Visual Studio .NET 自动添加 Button 控件的 Click 事件响应函数，在其中输入例 1-2 所示的代码。

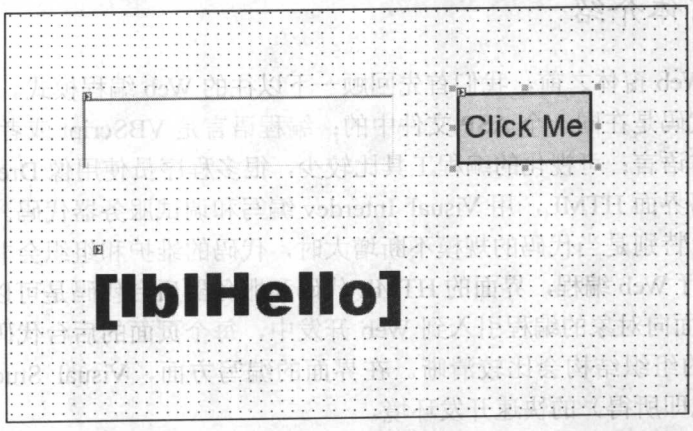

图 1-4　Web 窗体对话框

【例 1-2】Hello.aspx 后台代码

```
private void btnClick_Click(object sender, System.EventArgs e)
{
    if (tbName.Text.Trim().Length > 0)
    {
        lblHello.Text = "Hello, " + tbName.Text;
    }
}
```

最后，运行该页面，在输入框中输入一个名字，单击“Click Me”按钮，效果如图 1-5 所示。

图 1-5　页面运行效果

1.4 ASP.NET Web 窗体

1.4.1 Web 窗体介绍

在开始介绍 Web 窗体之前，我们首先回顾一下以往的 Web 编程模式。在 ASP 时代，服务器代码和页面代码是在同一个 ASP 文件中的；编程语言是 VBScript 或者 JavaScript，它们属于解释型的脚本语言；可视化的编程工具比较少，很多程序员使用像 Dreamweaver 这样的网页设计工具编写界面 HTML，用 Visual Interdev 编写和调试服务器代码。这种编程模式的效率是比较低的，特别是当代码的规模不断增大时，代码的维护和组织会非常的不方便。而使用 ASP.NET 进行 Web 编程，界面的 HTML 代码和服务器后台代码是可以分离的，更重要的是 ASP.NET 将面向对象的编程引入到 Web 开发中，每个页面的后台代码实际上对应了一个类，这样代码的组织结构会比较清晰。在界面的编写方面，Visual Studio .NET 提供了 WYSIWYG（所见即所得）的快速开发环境。

这种编程模式对于了解 VB 编程的读者一定不会陌生，在 VB 中每个程序窗口对应一个 VB 窗体。所谓窗体就是一个容纳各种控件的容器，各种控件都必须直接或者间接地和它有依存关系。ASP.NET 也引入了窗体的概念即 Web 窗体，每一个 Web 窗体都对应于一个页面，在 Web 窗体中可以方便地增加各种控件，对控件的属性和事件进行编程。

ASP.NET Web 窗体页是扩展名为 aspx 的文本文件，当浏览器客户端请求 aspx 资源时，ASP.NET 运行库分析目标文件并将其编译为一个.NET Framework 类，这个类将动态处理传入的请求。这里需要说明的是只有在第一次访问 aspx 文件时，ASP.NET 运行库会对其进行编译，当客户端再次请求该文件时，ASP.NET 运行库会直接执行已编译的类型实例。这就是为什么 ASP.NET 页面在第一次被访问时会相对比较慢，以后访问速度会大大提高的原因。这一点和 ASP 的情况完全不同，因为 ASP 只支持 VBScript 和 JavaScript 这样的脚本语言，页面是解释执行的，当用户发出请求后，无论是第一次，还是第 1000 次，ASP 的页面都会被动态解释执行；而 ASP.NET Web 窗体是一次编译多次执行的，它支持的可编译语言包括 VB.NET、C# 和 JScript.NET。

1.4.2 服务器控件

ASP.NET 的前身是 ASP，从语法上 AS 控件 P.NET 和 ASP 是兼容的，也就是说在 Web 窗体中仍然可以使用<%...%>这样的标记。除了使用<%...%>代码块编写动态内容代码外，还可使用服务器控件编写 Web 页代码。服务器控件的特征是包含 runat="server"属性值。例如：

```
<asp:TextBox id="TextBox1" runat="server"></asp:TextBox>
```

前面提到过，ASP.NET 运行库可以将 aspx 文件编译为一个类，含有 runat="server"标记的控件将会成为类的成员，而页面中的其他部分（例如下面的代码）会直接输出。

```
<INPUT type="button" value="Button">
```

ASP.NET 提供的常见服务器控件包括两种：HTML 服务器控件和 Web 服务器控件。表 1-1 和 1-2 所示为它们中所有控件的列表，控件的具体使用方法将在后面的章节中结合实例讲解。

表 1-1　　**HTML 服务器控件列表**

HtmlAnchor	HtmlButton	HtmlForm	HtmlGenericControl
HtmlImage	HtmlInputButton（按钮）	HtmlInputButton（重置）	HtmlInputButton（提交）
HtmlInputCheckBox	HtmlInputFile	HtmlInputHidden	HtmlInputImage
HtmlInputRadioButton	HtmlInputText（密码）	HtmlInputText（文本）	HtmlSelect
HtmlTable	HtmlTableCell	HtmlTableRow	HtmlTextArea

表 1-2　　**Web 服务器控件列表**

AdRotator	Button	Calendar	CheckBox
CheckBoxList	CompareValidator	CustomValidator	DataGrid
DataList	DropDownList	HyperLink	Image
ImageButton	Label	LinkButton	ListBox
Panel	PlaceHolder	RadioButton	RadioButtonList
RangeValidator	RegularExpressionValidator	Repeater	RequiredFieldValidator
Table	TableCell	TableRow	TextBox
ValidationSummary	XML		

1.4.3　用户控件

除了 ASP.NET 提供的内置服务器控件外，开发人员还可以使用与编写 Web 窗体页相同的方式来定义控件。基本上只需要对普通 Web 窗体页进行少量的修改，就能得到用户自己的服务器控件，简称为用户控件，通常使用.ascx 扩展名来标识这样的控件。用户控件通过 Register 指令包括在 Web 窗体页中。例如：

```
<%@ Register TagPrefix="Acme" TagName="Message" Src="pagelet1.ascx" %>
```

其中 TagPrefix 确定了用户控件的惟一命名空间，防止多个同名的用户控件发生混淆；TagName 是用户控件的惟一名称，Src 属性是用户控件的虚拟路径，例如“MyPagelet.ascx”或“/MyApp/Include/MyPagelet.ascx”。注册了用户控件后，可以像放置普通服务器控件那样，将用户控件标记放置在 Web 窗体页中，例如：

```
<Acme:Message runat="server"/>
```

1.4.4　自定义控件

用户控件实际上更接近于 Web 窗体，可以把它看作窗体的一部分。在 ASP.NET 中，开发人员还可以自定义控件。要创建简单的自定义控件，开发人员要做的只是从 System.Web.UI.Control 类派生出一个控件类，然后重写它的 Render 方法。例 1-3 给出了呈现消息字符串的简单控件。

【例 1-3】Simple.cs

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace book02
{
    public class Simple : Control
```

```
    {
        protected override void Render(HtmlTextWriter output)
        {
            output.Write("{0} {1} {2}","<H2>", "大家好！ ", "</H2>");
        }
    }
}
```

自定义控件的使用方法和用户控件类似，如例 1-4 所示。其中 Register 指令的 Assembly 属性指明了控件所在的程序集名称，如果 Simple.cs 代码和 WebForm1.aspx 在同一个项目中，那么 Assembly 就是整个项目的程序集名称；如果 Simple.cs 位于单独的控件库项目中，Assembly 属性则是它所在控件库的程序集名称。

【例 1-4】WebForm1.aspx

```
<%@ Page language="c#" Codebehind="WebForm1.aspx.cs" AutoEventWireup="false"
 Inherits="book02.WebForm1" %>
<%@ Register TagPrefix="cc" Namespace="book02" Assembly="book02" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
 <HEAD>
  <title>WebForm1</title>
  <meta name="GENERATOR" Content="Microsoft Visual Studio .NET 7.1">
  <meta name="CODE_LANGUAGE" Content="C#">
  <meta name="vs_defaultClientScript" content="JavaScript">
  <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
 </HEAD>
 <body>
  <form id="Form1" method="post" runat="server">
  <cc:Simple runat=server></cc:Simple>
  </form>
 </body>
</HTML>
```

自定义控件中还可以加入已有的 HTML 服务器控件或者 Web 服务器控件，这样的自定义控件被称为复合控件。实际上，目前.NET Framework 提供的所有服务器控件都可以看作是自定义控件，只是它们已经由框架实现，开发人员可以直接使用，而自定义控件一般是在现有控件不能满足要求的情况下进行开发。一般来说，自定义控件可以实现更灵活的功能，但是编程难度相对于用户控件就要大一些。更多的关于自定义控件开发的内容，读者可以参考 MSDN 等相关资料，本章不做详细介绍，在后面的章节中，会结合相应的实例进行具体的讲解。

1.5　ASP.NET Web 应用程序

1.5.1　Web 应用程序概述

在 ASP.NET 中，可以这样来定义一个应用程序：能够在一个 Web 应用服务器的子目录或者虚拟目录上运行的所有的文件、页面、操作、模块或者能被执行的代码。Web 服务器上的 ASP.NET 应用程序在一个被称作应用程序域运行空间（AppDomain）的环境中被执行，以保证类的隔离（没有版本、名称上的冲突）、安全屏蔽（防止未授权的情况下访问某些机器/网络的资源）和静态变量的隔离。

在 Web 应用程序的生存期内，ASP.NET 维护了 HttpApplication 的实例池，它会自动指派其中的某个实例处理应用程序接收到的 HTTP 请求。所指派的特定 HttpApplication 实例负责管理请求的整个生存期，直到请求完成后才被重新使用。ASP.NET Framework 应用程序在第一次向服务器发出请求时创建，此时会引发 Application_Start 事件。

应用程序（Application）对象在整个 Web 应用程序范围内有效，与之相对的是会话（Session）对象。一次会话是指从用户打开浏览器，浏览网站内的网页开始，到用户关闭浏览器结束浏览的过程。Session 对象在当前会话范围内有效，不能和其他会话共享，在会话开始时会引发 Session_Start 事件，会话结束时会引发 Session_End 事件。

应用程序区别于 Web 窗体的地方在于其整体的概念：全局类 Global.asax 和应用程序的状态对于应用程序中的每一个页面都是起作用的。

1.5.2　全局类 Global.asax

在 Visual Studio .NET 中新建一个“ASP.NET Web 应用程序”项目，在其中查看 Global.asax 的代码，如例 1-5 所示。

【例 1-5】Global.asax.cs

```
.....
protected void Application_Start(Object sender, EventArgs e)
{
}
protected void Session_Start(Object sender, EventArgs e)
{
}
protected void Application_BeginRequest(Object sender, EventArgs e)
{
}
protected void Application_EndRequest(Object sender, EventArgs e)
{
}
```