飞思科技
FECIT Sci-Tech
www.fecit.com.cn

开发专家之MS.NET

ASP.NET 应用开发指南

飞思科技产品研发中心 编著

系统讲解 ASP.NET 常用控件
全面接触 ASP.NET 高级技术
深入理解 ASP.NET 语法精髓

随书附赠光盘包含书中所有精彩范例的源代码

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

开发专家之MS.NET

ASP.NET 应用开发指南

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

随同.NET框架的发布，新一代的活动服务页面ASP.NET诞生了。跟前一版本相比，ASP.NET做了一系列改进，引入了相当多的新功能，例如所有的ASP.NET代码都经过编译，代码和用户界面的彻底分离，基于XML的配置等。利用ASP.NET基础框架提供的服务，开发人员可以快速高效地制作出功能强大、伸缩性良好的Web应用。

本书内容全面，基本上涵盖了ASP.NET应用开发中90%以上的知识，主要包括ASP.NET应用开发概览、ASP.NET语法、使用验证服务器控件、数据绑定、模板控件、创建和使用用户自定义控件、ASP.NET内部对象、使用Global.asax文件、使用ADO.NET、使用.NET组件、创建和使用Web服务、使用配置、使用缓存、安全性、实用技术选录等。通过本书的学习，读者将领会到ASP.NET应用开发的精华。

本书实例丰富，便于自学。对于每一个知识点，本书都提供了读者可以自己动手操作的实例，没有高深和冗长的原理说明，因而一切都变得简单明了。

本书面向ASP.NET的中、高级用户，要求读者有HTML方面的基础知识，并且至少了解一门.NET平台下的编程语言，如C#或Visual Basic。

图书在版编目（CIP）数据

ASP.NET 应用开发指南/飞思科技产品研发中心编著.—北京：电子工业出版社，2002.9
（开发专家之 MS.NET）
ISBN 7-5053-7958-5

Ⅰ.A... Ⅱ.飞... Ⅲ.主页制作—程序设计 Ⅳ.TP393.092

中国版本图书馆 CIP 数据核字（2002）第 063584 号

责任编辑：王树伟 黄建逊
印　　刷：北京东光印刷厂
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京海淀区万寿路 173 信箱　邮编：100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：29.5　字数：755.2 千字　附光盘 1 张
版　　次：2002 年 9 月第 1 版　2002 年 9 月第 1 次印刷
印　　数：4000 册　定价：42.00 元（含光盘）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系调换。联系电话：（010）68279077

出版说明

“开发专家”是电子工业出版社计算机研发部长期以来精心培育的计算机科学技术类本版品牌。这个品牌是由多个专题系列组成的横向大系列，涵盖了计算机技术的各个方面，特别是一直受到极大关注的程序开发类系列，例如《开发专家之数据库》、《开发专家之网络编程》、《开发专家之 Delphi》、《开发专家之 Sun ONE》以及《开发专家之 MS.NET》等。这些专题系列基于各自的角度，从纵向上包含了该专题的所有内容。因此，整个“开发专家”的品牌架构纵横交错，囊括了所有的计算机技术和所有的技术层面，海纳百川而又极具可扩展性。

“开发专家”的作者队伍主要依托于“飞思科技产品研发中心”。“飞思科技产品研发中心”是由专业的策划人员、权威的技术专家和精深的作者队伍共同构成。在图书的出版上，形成了以研发为基础、以出版为中心、以服务为支持的专业化出版框架和流程。通过深入的市场调查和技术跟踪，在综合了技术需求和读者焦点等因素的基础上，形成各系列丛书的写作重点和大纲，然后聘请业界的最前沿学者进行写作。同时，策划工作全程介入写作进程，严格控制写作质量，用最专业的技术背景、最深刻的理论基础、最具代表性的案例、最能为专业读者接受的形式，为读者提供品质最佳的图书产品。这体现了出版者和著作者的完美结合。

多年来，计算机研发部始终把创造社会效益摆在首位，秉承一切为国内计算机技术专业读者服务的精神，为推动国内 IT 技术发展、为体现国内技术的原创水平，穷尽所有的创意与努力，将出版者的命运与读者的支持紧紧地连在了一起。

在此，我们临出版之残酷竞争而不惧，旌旗猎猎而异军突起，这与广大读者的支持是分不开的。为了使我们的脚步更坚实、使我们的队伍永葆活力和创造力，我们期待着您能为我们的前进贡献出您的意见和建议。同时，我们也在等待着您的加入。

我们的联系方式：

电　　话：（010）68134545　　68131648

电子邮件：support@fecit.com.cn

飞思在线：http://www.fecit.com.cn　　http://www.fecit.net

勘误网址：http://www.fecit.com.cn/question.htm

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

电子工业出版社计算机研发部

前言

关于本丛书

对开发人员而言，.NET 是一个完美的开发平台。它提供了一套公共的运行库，并制定了一套公共语言规范，所有符合该规范的语言都可以无缝使用这套运行库。在.NET 平台下，除了语法上的区别，各种编程语言没有本质的不同。它们共享公共类库，具有类似的编程模型和相差无几的功能，开发人员可以自由选择自己喜爱的语言开发程序。.NET 平台提供大量的服务，包括垃圾自动收集、面向对象的多线程、基于程序集的部署、异常处理、特性编程、远程处理、ASP.NET 网页框架、互操作、安全性等，使开发人员可以快速构架任何应用，从包括传统的桌面应用到面向 Web 的大型分布式应用。.NET 将彻底改变软件的开发方式、使用方式和发行方式，.NET 将是一场软件革命。

Microsoft .NET 是一个用于构建、运行和体验下一代分布式应用程序的平台。它是跨客户端、跨服务器的开发人员工具，它由以下部分组成：

- .NET 框架编程模型，它使开发人员能够构建 Web 应用程序、智能客户端应用程序和可扩展标记语言 (XML) Web 服务应用程序，它们利用诸如 SOAP、可扩展标记语言 (XML) 和 HTTP 等标准协议以编程方式通过网络开放其功能。
- 开发人员工具，比如 Visual Studio.NET，该工具为用 .NET 框架进行编程提供了一个迅速集成开发应用程序的环境。
- 一组服务器，包括 Windows 2000、SQL Server 和 BizTalkServer，它可集成、运行、操作和管理 XML Web 服务和应用程序。
- 客户端软件，比如 Windows XP、Windows CE 和 Microsoft Office XP，它有助于开发人员跨多种设备和现有产品，为用户提供深切、引人的使用感受。

《开发专家之 MS.NET》系列丛书针对 Microsoft .NET 整个开发体系和特色组织作者进行了编写，它涵盖了 MS.NET 中的方方面面，为那些希望学习 MS.NET 技术的开发人员展示了非常完整的知识体系。在利用 MS.NET 进行项目开发的过程中，本丛书也可以为各种不同需要提供完美的解答。

关于本书

随同.NET 框架的发布，新一代的活动服务页面 ASP.NET 诞生了。跟前一版本相比，ASP.NET 做了一系列改进，引入了相当多的新功能，例如所有的 ASP.NET 代码都经过编译，代码和用户界面的彻底分离，基于 XML 的配置等。利用 ASP.NET 基础框架提供的服务，开发人员可以快速高效地制作出功能强大、伸缩性良好的 Web 应用。

本书读者对象

本书面向 ASP.NET 的中、高级用户，要求读者有 HTML 方面的基础知识，并且至少了解一门.NET 平台下的编程语言，如 C#或 Visual Basic。

本书内容

- ASP.NET 中的语法精髓，使用户在编程时有章可循，遇难不惊。
- ASP.NET 中的常用控件，包括服务器端的验证控件、模板控件和数据绑定控件。这些控件是 Web 窗体中出现频率最高的界面和功能元素，只有掌握这些控件，才能设计出高水平的 Web 窗体。
- 创建和使用用户自定义控件以及 ASP.NET 的内部对象，如 Request、Response、Application、Session、Server 等。
- ASP.NET 应用程序和 Gloabal.asax 文件。
- 使用.NET 组件创建简单的 XML Web 服务的方法以及 Web 服务的使用。
- ASP.NET 高级应用技术。

本书特点

跟同类书相比，本书有如下特色：

- 实例丰富，便于自学

对于每一个知识点，本书都提供了读者可以自己动手操作的实例，没有高深和冗长的原理说明。因为有了丰富的实例，一切都变得简单明了。

- 内容全面，自成一体

本书内容涵盖了 ASP.NET 应用开发中 90%以上的知识。通过本书的学习，读者将领会到 ASP.NET 应用开发的精华。

学习建议

学习之前，读者最好安装 Visual Studio.NET 集成开发工具。若没有，至少也要安装.NET 框架 SDK。微软官方网站提供了免费的.NET 框架 SDK 下载。

在开始学习 ASP.NET 之前，读者最好花一些时间熟悉一下 HTML 以及 ASP 编程方面的知识。实际动手编写几个 HTML 和 ASP 的例子是最好的准备。这方面的知识，读者可以参考我社已出版的两本来自我国台湾省的图书：《完全接触 ASP 之 JScript》和《完全接触 ASP 之 VBScript》。

同时对.NET 编程语言 C#或 Visual Basic 的了解是非常重要的，因为本书对这些编程语言做过很多的介绍。读者可以参考我社出版的同套图书《Visual Basic.NET 编程指南》和《C#编程指南》。

一般而言，按照本书的章节顺序阅读本书对初学者来说是比较合适的。但本书的各章

我们涉及到独立的专题，因此中、高级读者可以选择自己感兴趣的章节开始学习。对于初学者，我们希望能认真阅读本书的前 11 章，然后结合自己的编程任务再学习相关内容。

本书在写作过程中，参考了微软公司的相关资料，在此特做说明。

本书由飞思科技产品研发中心策划并组织编写，同时参与写作的还有刘晓华、李华、张宏伟、张彦超、王国华、李国志、李书德、张庆国、陈海涛、丁晖、赵洁、刁立立等。由于编者水平有限，再加上时间仓促，虽然经过再三勘误，但仍难免有纰漏之处，欢迎广大读者予以批评和指正。

我们的联系方式：

电　　话：（010）68134545　　68131648

电子邮件：support@fecit.com.cn

飞思在线：http://www.fecit.com.cn　　http://www.fecit.net

勘误网址：http://www.fecit.com.cn/question.htm

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

目录

第 1 章　ASP.NET 应用开发概览

ASP.NET 是下一代的活动服务页面框架。ASP.NET 是.NET 框架中的固有部分，利用 ASP.NET 提供的服务，可以开发出新一代的 Web 应用。

本章将介绍下面内容：

- HTML、ASP 应用和 ASP.NET 应用的比较
- ASP.NET 应用开发前的准备工作
- 用 Visual Studio.NET 开发第一个 ASP.NET 应用

1.1　HTML、ASP 和 ASP.NET

在开始 ASP.NET 介绍之前，我们先简单追述一下 Web 应用开发的历史。最初的 Web 应用是基于静态的 HTML 文本。利用一些预定义的标志，诸如 FrontPage 等工具就可以产生这样的文本。HTML 较好地解决了信息的显示问题，但这些信息和控制显示格式的标志在服务器端是编码存储的，这时应用程序的执行模式为：客户请求某一个静态 HTML 文本，服务器返回该文本。下面是一个典型的 HTML 文本的示例：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta http-equiv="Content-Language" content="zh-cn">
<title>主页</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta name="Microsoft Theme" content="strtedge 011, default">
<meta name="Microsoft Border" content="tl, default">
</head>
<body>
    <p><font size="4">欢迎光顾我的站点！ </font></p>
    <p>主页是让到访者了解本站点主旨的最佳场所。主页向到访者展示了您站点的风格。
</p>
    <p>  </p>
    <p>此网页最近更新于<!--webbot bot="Timestamp" s-type="REGENERATED" s-format=
"%y 年%m 月%d 日" -->。</p>
    <p align="right"><a href="http://www.microsoft.com/frontpage/"><img
src="images/frontpag.gif" border="0" width="88" height="44"></a></p>
</body>
</html>
```

上述 HTML 文本是用 FrontPage 工具创建的。文本的结构大致如下：

```
<html>
```

```
<head>
  <title>主页</title>
</head>
<body>
</body>
</html>
```

标志<html></html>之间的是 HTML 文件的内容。<html>标志指示上述文件为一个 HTML 文档，<title></title>标志指示文本的标题，<bod></body>之间的是 HTML 文件的正文。

IE 浏览器能够解释结构不完整的 HTML 文件，所以上述文件中去掉第一行的<html>和最后一行的</html>，也能在浏览器中正确地显示。但建议读者在设计 HTML 文件时，保证其结构的完整性，因为这样可以增强代码的可读性。

观察上述 HTML 文件，会发现每个标志都有若干属性，例如，<font size="4">中的 font 标志有 size 属性，表示 size 字体为 4 号字。格式信息正是通过标志和标志属性设置的。

在 IE 浏览器下，上述 HTML 文件的浏览效果如图 1-1 所示。

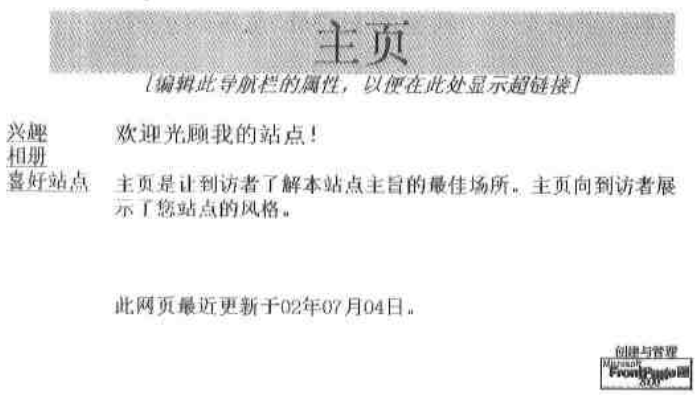

图 1-1　浏览器下查看 HTML

从浏览效果来看，HTML 文件提供了比较丰富的界面元素。其外观效果远非文本效果可以比拟。但 HTML 的内容都是静态准备好的，如果打算制作网页反映股票信息，实现这样的任务几乎是不可能的，因为股票信息时刻在变，而为了反映这种变化，就必须时刻重新编写 HTML。为了解决动态信息的发布问题，活动服务页面 ASP 诞生了。

活动服务页面在 HTML 的基础上引入了一些额外的标志，这些标志指示服务器在返回用户请求的页面之前进行进一步的处理。对客户而言，它最终仍然会得到一个 HTML 文件，但该 HTML 文件是在服务器端动态生成的。对服务器而言，它要根据用户的请求把活动页面中的内容翻译成 HTML 代码，开发 ASP 应用工具有 InterDev 等。下面是一个 ASP 文件（demo.asp）的示例：

```
<html>
<head>
<title>演示 ASP</title>
</head>
<body>
    <% Response.Write("Hello," +Request("user")) %>
</body>
</html>
```

ASP 文件跟 HTML 文件类似，也可以包括<html>等标志，但其中有一种特殊的<% %>标志，该标志指示期间的代码要在服务器上执行，执行的结果将作为 HTML 文件的一部分返回给客户。

其中 Response 对象是 ASP 内置的对象之一，表示对用户的回应，Response.Write 方法用以向客户端写入回应消息。Request 对象是另一个内置对象，表示用户的请求，例如通过该对象可以获得用户获取的 Form 表单（如果通过 Post 方法提交）或者通过 Get 方法提交请求时的参数。代码 Request("user")表示获得 Get 参数列表中的 user 参数。

下面的命令用以请求 demo.asp，并指定 user 参数为“晓华”：

http://localhost/demo.asp?user=晓华

浏览器的显示结果如图 1-2 所示。

图 1-2 客户端浏览器显示请求响应

- 界面和代码没有分开

ASP 应用中的代码包含在界面元素之间，这样导致程序结构不清晰，增加了维护代码的难度。

- 解释性的脚本语言

ASP 中所用到的编程语言是解释性的脚本语言，如 VBScript 或 JScript。每次服务器响应客户的请求，都要调用某种语言引擎解释脚本代码，这样降低了程序的执行效率，会影响对客户的响应。

ASP.NET 克服了 ASP 上述不足，利用 Visual Studio.NET 集成开发工具，开发人员甚至可以像设计 Visual Basic 程序那样开发 ASP.NET 应用。

通过一种所谓的隐藏代码技术，彻底实现了界面和代码分离。下面是一个利用 Visual Studio.NET 产生的空白页面示例，它由两部分组成，界面和代码，两部分都被保存为一个独立的文件。界面文件以.aspx 为后缀，下面是其内容：

```
[Visual Basic]
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="WebForm1.aspx.vb"
Inherits="CH1.WebForm1"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
    <head>
        <title>WebForm1</title>
        <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.0">
        <meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
        <meta name="vs_defaultClientScript" content="JavaScript">
        <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
    </head>
    <body MS_POSITIONING="GridLayout">
        <form id="Form1" method="post" runat="server">
        </form>
    </body>
</html>
```

```
        <form id="Form1" method="post" runat="server">
98      </form>
    </body>
</HTML>
```

首先注意.aspx 文件的第一行，该行是一个页面 Page 指令。Language 指示页面的编程语言，可以为 C#、Visual Basic 或任意一种.NET 平台下的编程语言。Codebihand 属性指示于界面元素关联的代码文件，Inherits 属性用以指示当前页面的父类，AutoEventWireup 属性指示是否在页面加载时，自动调用页面中定义的 Page_Init 和 Page_Load 方法。

一个.aspx 文件实际上隐含定义一个 Page 派生类，它们直接或间接地从 Page 类派生。

我们再看看代码隐藏文件，隐藏代码定义了一个页面类，它直接从 System.Web.UI.Page 派生。下面是其具体内容：

```
[Visual Basic]
Public Class WebForm1
    Inherits System.Web.UI.Page
#Region " Web 窗体设计器生成的代码 "
    '该调用是 Web 窗体设计器所必需的。
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    End Sub
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Init
        'CODEGEN: 此方法调用是 Web 窗体设计器所必需的。
        '不要使用代码编辑器修改它。
        InitializeComponent()
    End Sub
#End Region
    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        '在此处放置初始化页的用户代码
    End Sub
End Class
[C#]
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
namespace CSCH1
{
```

```
    /// <summary>
    /// WebForm1 的摘要说明。
    /// </summary>
    public class WebForm1 : System.Web.UI.Page
    {
        private void Page_Load(object sender, System.EventArgs e)
        {
            // 在此处放置用户代码以初始化页面
            //Request.Form
        }
        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            //
            // CODEGEN: 该调用是 ASP.NET Web 窗体设计器所必需的。
            //
            InitializeComponent();
            base.OnInit(e);
        }
        /// <summary>
        /// 设计器支持所需的方法 - 不要使用代码编辑器修改
        /// 此方法的内容。
        /// </summary>
        private void InitializeComponent()
        {
            this.Load += new System.EventHandler(this.Page_Load);
        }
        #endregion
    }
}
```

如果读者有 Visual Basic.NET 或 C#的编程经验，读懂上述代码应该没有任何问题。上述代码的主题是重载父类的 OnInit 方法和为 Load 事件添加处理代码。通过下面的方法挂钩事件：

```
[Visual Basic]
```

方法后跟 Handlers 关键字指示处理的事件，上例中的 Page_Load 方法即是一例。

为了保证该方法正确的处理事件，必须保证该方法的签名符合事件类型。在 Visual Basic 中，事件类型用委托来定义，例如：Web 窗体的 Load 事件由下面的委托定义：

```
<Serializable>
Public Delegate Sub EventHandler( _
    ByVal sender As Object, _
    ByVal e As EventArgs _
)
```

所以 Page_Load 方法的的第一个参数必须为 Object 类型，第二个参数必须为 EventArgs

类型，并且该方法无返回值（为过程）。

[C#]

把委托实例跟事件直接挂钩。把本例中挂钩事件的处理代码重写如下：

```
this.Load += new System.EventHandler(this.Page_Load);
```

代码右边生成一个 EventHandler 委托实例。this.Page_Load 方法的签名必须符合该委托。代码左边是要处理的事件（实际上是一个委托变量），运算符+=将左边的委托实例和右边的事件关联起来。

C#中 this 关键字表示对象自身，而 Visual Basic 中对应的关键字是 Me。

从上面的例子可以看出，ASP.NET 真正实现了将代码和内容分开，已有的开发其他.NET 应用的经验可以直接用到 ASP.NET 应用开发中来。

需要指出，代码隐藏文件并不是必须的，但利用它，可以将代码和界面分离，因此是一种值得推荐的做法。另外一种替换的编程方式跟传统的 ASP 应用开发类似，即把所有的代码和一些 HTML 标志元素混在一起，并保存为一个单独的文件（这样的文件仍以.aspx 文为后缀）。下面是单文件的窗体示例：

[Visual Basic]

```
<html>
<!--
   演示 ASP.NET
!-->
<Script language="vb" runat="server">
      public sub Page_Load( sender as Object,   e as EventArgs)
           if not IsPostBack then
              Message.Text="Hello,World!"
           end if
      end sub
</Script>
<body>
   <h4><font face="verdana">
      <asp:Label id="Message" runat="server"/>
   </font></h4>
</body>
</html>
```

[C#]

```
<html>
<!--
   演示 ASP.NET
!-->
<Script language="C#" runat="server">
      public void Page_Load(Object sender, EventArgs e) {
           if(!IsPostBack)
              Message.Text="Hello,World!" ;
      }
```