



创作效果百例丛书

ASP.NET

精彩编程百例

姚治安 周志宏 杨 柳 等编著



Programming



中国水利水电出版社
www.waterpub.com.cn

ASP.NET 精彩编程百例

姚治安 周志宏 杨柳 等编著

中国水利水电出版社

内 容 提 要

本书通过 100 个生动实例介绍了 ASP.NET 编程知识,尤其是相对于 ASP 新增的功能。主要包括 Web 控件的使用、HTML 控件的使用、常见对象的使用、编写用户控件、编写应用程序、ADO.NET 数据库操作、表单验证等内容。

本书的实例中都给出了详细的实例介绍、技术指南、实例功能、程序代码和结果演示。图文并茂,便于理解,具有很强的可操作性。

图书在版编目(CIP)数据

ASP.NET 精彩编程百例/姚治安等编著. —北京:中国水利水电出版社, 2003

(万水创作效果百例丛书)

ISBN 7-5084-1463-2

I. A… II. 姚… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2003) 第 021996 号

书 名	ASP.NET 精彩编程百例
作 者	姚治安 周志宏 杨柳 等编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (发行部)、82562819 (万水)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京北医印刷厂
规 格	787×1000 毫米 16 开本 22.875 印张 543 千字
版 次	2003 年 4 月第一版 2003 年 4 月北京第一次印刷
印 数	0001—5000 册
定 价	32.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

前 言

ASP.NET 刚出现时被称为 ASP+, 它是 ASP 的下一代版本, 但并不仅仅是 ASP 的简单升级。在许多方面, 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激动人心的新技术, 正是这些具有革新意义的新特性, 让 ASP.NET 远远超越了 ASP。ASP.NET 吸收了 ASP 以前版本的最大优点并参照 Java、Visual Basic 语言的开发优势加入了许多新的特色, 同时也修正了以前的 ASP 版本的运行错误。

ASP.NET 是微软发展的新型体系结构 .NET 的一部分, 它的全新技术架构会让每一个人的网络生活都变得更简单。ASP.NET 建立在 CLR (Common Language Runtime, 通用语言运行机) 基础之上, 它主要用于在服务器上开发功能强大的 Web 应用。与以往的 Web 技术相比, ASP.NET 具有许多重要特性, 例如: 优良的性能、大量工具箱的支持、强大的功能及良好的机动性、易用性、崭新的配置管理方式、可移植性和良好的适用性、超强的扩展性。

本书面向初中级用户, 通过实例全面介绍了 ASP.NET 的基础知识和具体的应用。全书共有 100 个实例, 分为三大部分:

- ◇ 第一篇: 基础控件详解
- ◇ 第二篇: 对象应用与数据库操作
- ◇ 第三篇: 高级应用

每个实例包括以下内容:

- ◇ 实例说明: 简单描述实例中将介绍的 ASP.NET 控件、方法或属性。
- ◇ 技术指南: 列示实例程序编写过程中的技术问题。
- ◇ 实例功能: 实例程序实现的功能。
- ◇ 程序代码: ASP.NET 程序的源代码。
- ◇ 结果演示: 详细演示程序运行时的 Web 页面。

书中实例的运行环境为:

- ◇ 操作系统: Windows 2000 professional
- ◇ 浏览器: Internet Explorer 5.5
- ◇ .NET Framework SDK Beta 2

本书所有例子的源代码均可在中国水利水电出版社网站 www.waterpub.com.cn 上下载。

参加本书编写工作的除姚治安、周志宏、杨柳外, 还有王景华、李蓉、王明凯、孔玲飞、张远、蔡振华、赵晓明、李明、赵海斌、王鹏、韩天雷、刘建峰、马明宇、刘欣、邹建鹏、范晨毅、罗静、周鑫华、刘建华、王逸凡、吴玉青、耿云、马纪明、张明、高一波、杜马辰等, 在此感谢他们所做的工作。

由于时间仓促及作者水平有限, 书中难免会有不足和错误之处, 敬请读者批评指正。

作者

2003 年 2 月

目 录

前言

第一篇 基础控件详解

实例 1 第一个 ASP.NET 实例	2
实例 2 页面处理—— Page_Load 事件	4
实例 3 页面处理—— Page_Unload 事件	7
实例 4 使用 HyperLink 控件	9
实例 5 使用 Button 控件	11
实例 6 处理 OnClick 事件	14
实例 7 使用 LinkButton 控件	16
实例 8 使用 ImageButton 控件	18
实例 9 使用 Label 控件	21
实例 10 使用 TextBox 控件	23
实例 11 使用 CheckBox 控件	25
实例 12 使用 CheckBoxList 控件	28
实例 13 使用 RadioButton 控件	32
实例 14 使用 RadioButtonList 控件	35
实例 15 使用 DropDownList 控件	39
实例 16 使用 ListBox 控件	42
实例 17 使用 Image 控件	46
实例 18 广告旋转板	49
实例 19 Web 控件处理表格	53
实例 20 制作日历	57
实例 21 群组控件	63
实例 22 使用 InnerHtml 和 InnerTest 属性	67
实例 23 使用 Disable 属性	69
实例 24 使用 Visible 属性	71
实例 25 使用 Attributes 属性	73
实例 26 使用 Style 属性	75

实例 27	使用 HtmlAnchor 控件	77
实例 28	使用 HtmlImage 控件	79
实例 29	使用 HtmlButton 控件	81
实例 30	使用 HtmlForm 控件	84
实例 31	使用 HtmlInputButton 控件	87
实例 32	使用 HtmlInputTest 控件	89
实例 33	使用 HtmlInputRadio 控件	92
实例 34	使用 HtmlInputCheckBox 控件	94
实例 35	使用 HtmlSelect 控件	97
实例 36	使用 HtmlInputHidden 控件	100
实例 37	使用 HtmlInputFile 控件	102
实例 38	使用 HtmlTextArea 控件	106
实例 39	使用 HtmlTable 控件	109
实例 40	使用 HtmlInputImage 控件	114
实例 41	使用 HtmlGenericControl 控件	117

第二篇 对象应用与数据库操作

实例 42	Request 对象的使用	121
实例 43	获得环境变量值	124
实例 44	获取客户端信息	127
实例 45	取出用户所上传的参数	129
实例 46	用 Form 方法发出和接收数据	132
实例 47	Response 对象的使用	135
实例 48	Server 对象的使用	138
实例 49	Application 对象的使用	141
实例 50	Session 对象的使用	146
实例 51	设定 Session 对象的有效期限	151
实例 52	Cookies 对象的使用	154
实例 53	使用 DataReader 对象	159
实例 54	使用 Update 对象	162
实例 55	使用 DataSet 对象	166
实例 56	使用 DataSetDelete 对象	168
实例 57	使用 DataSetInsert 对象	171
实例 58	使用 DefaultView 对象 1	174

实例 59	使用 DefaultView 对象 2.....	176
实例 60	使用 DataGrid 控件 1.....	179
实例 61	使用 DataGrid 控件 2.....	182
实例 62	使用 DataGrid 控件 3.....	185
实例 63	使用 DataGrid 控件 4.....	189
实例 64	使用 DataGrid 控件 5.....	193
实例 65	使用 ListBoxWeb 控件绑定数组.....	196
实例 66	使用 Web 控件绑定至 ArrayList.....	198
实例 67	绑定至 DataView.....	200
实例 68	RepeaterWeb 控件.....	203
实例 69	使用 DataList 控件 1.....	206
实例 70	使用 DataList 控件 2.....	211

第三篇 高级应用

实例 71	建立用户控件.....	217
实例 72	可设置属性的用户控件.....	219
实例 73	含有事件的用户控件.....	222
实例 74	表单的验证 1.....	228
实例 75	表单的验证 2.....	231
实例 76	表单的验证 3.....	235
实例 77	表单的验证 4.....	238
实例 78	表单的验证 5.....	242
实例 79	表单的验证 6.....	246
实例 80	读取文本文件.....	249
实例 81	列出目录内容.....	252
实例 82	文档上传操作.....	257
实例 83	多档上传.....	261
实例 84	邮件发送程序.....	264
实例 85	使用视图状态.....	269
实例 86	应用程序的配置文件.....	278
实例 87	管理瞬态状态.....	281
实例 88	配置 global.asax 文件.....	286
实例 89	单位类别.....	293
实例 90	为 HTML 控件添加风格.....	295

实例 91 为 Web 控件添加风格.....	298
实例 92 页面输出缓存.....	300
实例 93 数据缓存.....	302
实例 94 Web 控件的基础属性.....	308
实例 95 多按钮事件.....	311
实例 96 发送带有附件的邮件.....	316
实例 97 ASP.NET 生成图形和图像.....	322
实例 98 制作留言板.....	336
实例 99 Cookie 验证方式.....	345
实例 100 购物网站注册页面.....	350

第一篇

基础控件详解

ASP.NET 服务器控件是 ASP.NET 网页表单技术的核心内容，它提供了在网页上实现页面显示和事件处理的能力。从本质上说，控件就是一种程序设计对象，它可以作为用户界面的一个部分加以显示。ASP.NET 引入了两套控件：HTML 控件和 Web 控件，它们都是“服务器控件”。这些控件弥补了 HTML 的先天不足，此外还在往返传递中提供了保留浏览器状态的方法。当用户提交一个表单或发生引起向服务器发送消息的事件时，就会出现往返的传递。

在这个部分里将首先通过 3 个实例介绍 ASP.NET 程序的基本特征和处理过程，然后详细地介绍 18 个 Web 控件和 15 个 HTML 控件的使用。

另外还介绍一些最常见的控件事件和属性。而对数据库操作的 Web 控件和验证控件将在后面的部分中介绍。

Let's GO! 

实例

1

第一个 ASP.NET 实例



实例说明

该实例作为本书的第一个实例，将简单介绍 ASP.NET 窗体页的特点和基本语法。



技术指南

❏ ASP.NET 窗体页是带 .aspx 文件扩展名的声明性文本文件。

❏ `<% @Page %>` 指令

Page 指令往往放在程序的最开始，用来设定 ASP.NET 程序的个别属性，包括 Language、ResponseEncoding、Trace、TraceMode 等。

❏ 声明代码语法：`<script Language="..." runat="server">`

Language 属性指定 `<script></script>` 之间的代码使用的编程语言，这里指定的编程语言必须与 `<% Page Language="..." %>` 中指定的语言相同，ASP.NET 支持的编程语言包括 VB.NET、C#、JavaScript 等。

runat="server" 属性表明在 `<Script></Script>` 之间的代码将不会在客户端运行，这些代码将会直接在服务器上运行。

❏ 服务器端注释语法：`<%--注释--%>`

在 `<%--注释--%>` 之间定义的代码将会被视为注释语句，不予执行。服务器端注释使页开发人员能够防止服务器代码（包括服务器控件）和静态内容执行或呈现。

❏ 对象标记语法：`<object runat="server" />`

对象标记使页开发人员能够用基于标记的声明语法声明和创建变量的实例。

❏ `<% @Import>` 指令

Import 指令只有一个属性 Namespace，被用来声明一个范围，这个范围是惟一的。在这个 Namespace 范围内，允许开发者使用属于 Namespace 范围内的类。

□ 还有一些其他的语法，将在后面的例子中陆续介绍：

ASP.NET 服务器控件语法

ASP.NET HTML 服务器控件语法

数据绑定语法：<%# %>

服务器端包含语法：<-- #include File="Location.inc" -->



实例功能

该实例是一个简单的 ASP.NET 窗体页，从中可以看到 ASP.NET 的一些基本特征，用到了<%@Page %>指令、声明代码语法、服务器端注释语法和服务器控件语法。



程序代码

```
<%@ Page Language="C#" %>
<html><H3>第一个 ASP.NET 实例</H3>
<script language="C#" runat=server>
    void Page_Load(Object sender, EventArgs e)
    {
        Message.Text = "欢迎使用本 ASP.NET 实例教材";
    }
</script>
<body>
    <asp:label Id="Message" font-size=24
runat=server/>
<%--注释：这是一个 Label 控件--%>
</body></html>
```



结果演示

在记事本中输入以上代码，保存为 First.aspx，用 HTTP 的方式访问。显示结果如图 1-1 所示。



图 1-1 演示结果

实例

2

页面处理——Page_Load 事件



实例说明

该实例将介绍 ASP.NET 页面处理的第一个过程，它是通过 Page_Load 事件来完成的。网页每次在加载时会先发生 Page_Load 事件，网页内容全部下载完毕时则会发生 Page_Unload 事件，Page_Unload 事件将在下一个实例中介绍。



技术指南

- ❏ 服务器接收客户端的请求，页面的状态被重建，然后触发 Page_Load 事件。在这个过程中，可以读取或者重置页面的属性和控件的属性。
- ❏ Page_Load 是一个事件。事件 (Event) 是对象所认识的动作，如果希望对象的某些状况有所改变时，可以发出讯息通知我们执行一些相应的动作，这时就可以靠事件来完成。
- ❏ 以下为事件程序的语法：

```
Sub 对象名称_事件名(参数 1 As 类型, 参数 2 As 类型,...)
    程序代码...
End Sub
```
- ❏ 事件程序必须被 Script 标注括起来，事件程序是以 Sub 关键词声明的程序，必须指明希望响应哪个对象所发生的事件，而对象和事件之间要用下划线来作分隔。事件程序一样可以接收参数，这些参数由系统提供，表示发生某个事件时的一些状态。
- ❏ 处理 Page_Load 事件时，可以根据 IsPostBack 属性判定页面是否为第一次被请求，执行数据绑定。
- ❏ Page_Load 事件可以用来对数据库进行连接，可以利用 Page_Load 事件先将要处理的数据从数据库读取出来，再利用 Page_Unload 事件将使用完毕的对象清除，或是将连到数据库的连接关闭。



实例功能

该实例为常见的登录界面，利用 Page_Load 事件判断页面是不是第一次被请求，并判断接收用户名和密码信息，显示登录成功或失败。用到了 Page_Load 的 IsPostBack 属性。



程序代码

```
<html>
<head>
<title>Page_Load 事件</title>
<script language="VB" runat="Server">
Sub Page_Load()
If Not IsPostBack Then
message.InnerHtml="欢迎您，" & _
"请输入用户名和密码。"
End If
End Sub
Sub Button1_Click(source As Object, e As EventArgs)
If userName.Value.Equals("yang") And _
password.Value.Equals("123456") Then
message.InnerHtml="登录成功，欢迎！"
Else
message.InnerHtml="登录失败，请重试。"
End If
End Sub
</script>
</head>
<body>
<H3>处理 Page_Load 事件<Hr></H3>
<form runat=server>
用户名: <input Id="username" runat=server>
<BR>密__码: <input Id="password" type=password
runat=server>
<BR><input type=submit OnServerClick="Button1_Click"
runat=server>
</form>
<p><span Id="message" runat=server/>
</body>
</html>
```

结果演示

① 在记事本中输入以上代码，保存为 Page_Load.aspx。

② 用 HTTP 的方式访问 Page_Load.aspx，结果如图 2-1 所示。

③ 输入用户名：“yang”，密码：“123456”，登录成功，如图 2-2 所示。

④ 如果用户名或密码错误，将登录失败，如图 2-3 所示。

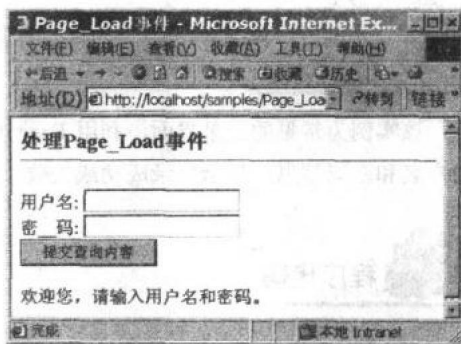


图 2-1 演示结果一

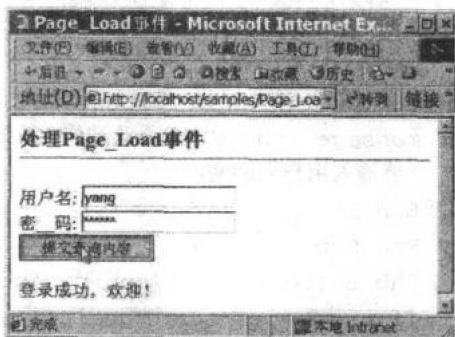


图 2-2 演示结果二

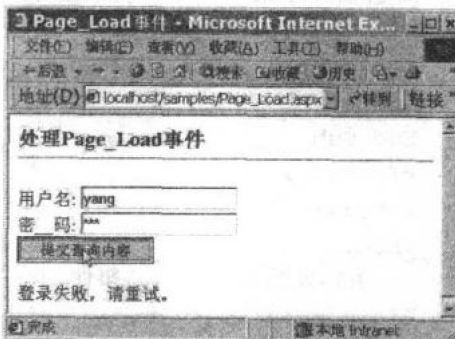


图 2-3 演示结果三

实例

3

页面处理——Page_Unload 事件



实例说明

该实例将介绍 ASP.NET 页面处理的下一个过程，它是通过 Page_Unload 事件来完成的。Page_Unload 事件是在网页内容全部下载完毕时发生的。



技术指南

- ❑ 页面处理完毕，需要做些清理工作，一般可以关闭打开的文件和数据库，或释放对象。
- ❑ 由于网页是 Page 对象，所以在触发 Load 事件时程序会到 Script 标注中寻找名称为 Page_Load 的事件程序，若找到此事件程序则执行，若没找到则不执行。事件程序执行完毕后，程序会回到正常的程序中继续执行程序，等所有的数据下载完毕后即触发 Unload 事件。
- ❑ 由于 Unload 事件最主要的工作是做一些收尾，并且是在所有数据完成下载后触发，所以不可以使用 Response.Write() 方法将数据再输出到浏览器，否则会发生错误。



实例功能

该实例为一个创建文件并写入的程序，用 Page_Load 事件创建并写入文件，用 Page_UnLoad 事件关闭文件。



程序代码

```
<%@ import namespace="system.io" %>
<html>
<head><title>Page_UnLoad 事件</title></head>
<body>
    <H3>处理 Page_UnLoad 事件<Hr></H3>
    <script language="vb" runat="server">
        public Dim writeFile As StreamWriter
```

```
Sub Page_Load(Sender As Object,E as EventArgs)
writeFile = File.CreateText( "C:\Page_UnLoad.txt" )
writeFile.WriteLine( "这是一个测试文件!" )
writeFile.WriteLine( "使用了 Page_Load 事件和 Page_Unload 事件!" )
Response.Write( "创建并写入 Page_UnLoad.txt" )
End Sub
Sub Page_UnLoad(Sender AS Object, E as EventArgs)
writeFile.Close
End Sub
</script>
</body>
</html>
```



结果演示

- ① 在记事本中输入以上代码，保存为 Page_UnLoad.aspx。
- ② 用 HTTP 的方式访问 Page_UnLoad.aspx，结果如图 3-1 所示。



图 3-1 演示结果一

- ③ 在 C 盘根目录下找到一个文件名为 Page_UnLoad.txt 的文本文件，将看到如图 3-2 所示的内容。

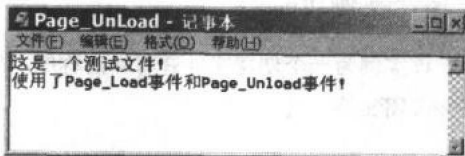


图 3-2 演示结果二

实例

4

使用 HyperLink 控件



实例说明

该实例介绍了 HyperLink（超链接）控件的使用，HyperLink 控件用于从一个页面定向到另一个页面，不过和 HTML 中的超级链接不同的是，HyperLink 控件可以使用数据绑定，可以用于文字链接和图像链接。



技术指南

- 设定 NavigateUrl 属性为要浏览的地址，在用户单击此链接时即可连至指定的地址。
- Target 属性可以在设有框架（Frame）的网页上，决定此链接要开启在哪个框架。或另外开启新的窗口。
- 设定 ImageUrl 属性则可以产生一个图形链接。
- 图形模式下的 HyperLink 控件如果设定了 Text 属性，则鼠标移到图形上时会出现小提示。
- 其使用语法为：

```
<ASP:Hyperlink Id="控件
Id" Runat="Server" Text="超级链接文字或小提示文字"
ImageUrl="图片所在地址" Target="超级链接所要显示的窗口" />
或 <ASP:Hyperlink Id="被程序代码所控制的名称" Runat="Server"
ImageUrl="图片所在地址" Target="超级链接所要显示的窗口" />
超级链接文字
</ASP:Hyperlink>
```



实例功能

该实例包括一个文字链接和一个图片链接，均是链接到微软主页 www.microsoft.com。Text 属性分别设为 Microsoft 和“链接到微软主页”，当鼠标停在图片上时，出现小提示“链接到微软主页”。用到了 HyperLink 控件的 NavigateUrl 属性、Text 属性、ImageUrl 属性和 Target 属性。

