ASP.NET

标准教程

卓越文化 UNIQUE ZHUOYUE WENHUA

主编 熊松明

航空工业出版社

ASP.NET 标准教程

主　编　熊松明

编　委　刘　春　迟振春

　　魏　霞　杨志利

航空工业出版社

内 容 提 要

ASP.NET 是 ASP 的新一代版本，然而 ASP.NET 又并非从 ASP 3.0 自然演化而来，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，同时也提供给 Web 开发人员更好的灵活性，有效地缩短了 Web 应用程序的开发周期。ASP.NET 与 Windows 2000 Server/Advanced Server 的完美组合，为中小型乃至企业级的 Web 商业模型提供了一个更为稳定、高效、安全的运行环境。

本书主要面向熟悉 VB 语法而对 ASP.NET 比较陌生的初、中级用户，本书由浅入深、层层深入地讲解了 ASP.NET 技术，书中涉及到的一些语法都用 C#、VB、Jscript 三种语言来分别阐述，但是范例程序都使用了 VB.NET 的语法。所以，如果读者对 VB 的语法很熟悉的话，将很快就可以上手。如果是精通别的编程语言的用户，通过对本书的阅读，也会让读者跟上编程技术发展的前沿。

图书在版编目（CIP）数据

ASP.NET 标准教程 / 熊松明主编. —北京：航空工业出版社，2002.10
ISBN 7-80183-042-3

Ⅰ.A… Ⅱ.熊… Ⅲ.主页制作—程序设计—教材
Ⅳ.TP393.092

中国版本图书馆 CIP 数据核字（2002）第 064719 号

航空工业出版社出版发行
（北京市安定门外小关东里 14 号 100029）
北京云浩印刷厂印刷　　全国各地新华书店经售
2002 年 10 月第 1 版　　2002 年 10 月第 1 次印刷
开本：787×1092　1/16　　印张：20.25　　字数：378 千字
印数：1-8000　　定价：28.00 元

前　言

ASP.NET 也叫做 ASP+，是 Microsoft 推出的新一代 Active Server Pages 脚本语言，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。需要特别指出的是，ASP.NET 不仅仅只是有了一个新界面和修复了一些缺陷的 ASP 3.0 的升级版本，更为重要的是，ASP.NET 除吸收了 ASP 以前版本的最大优点外，还参照 Java、VB 语言的开发优势加入了许多新的特色，并且修正了以前的 ASP 版本的运行错误。

ASP.NET 是建立在公共语言运行库上的编程框架，可用于在服务器上生成功能强大的 Web 应用程序。与以前的 Web 开发模型相比，ASP.NET 提供了数个重要的优点：

- 增强的性能

ASP.NET 是在服务器上运行的编译好的公共语言运行库代码。与被解释的“前辈”不同，ASP.NET 可利用早期绑定、实时编译、本机优化和盒外缓存服务，这相当于在编写代码行之前便显著提高了性能。

- 世界级的工具支持

ASP.NET 框架补充了 Visual Studio 集成开发环境中的大量工具箱和设计器。所见即所得编辑、拖放服务器控件和自动部署，是这个强大的工具所提供的众多功能中的几种。

- 威力和灵活性

由于 ASP.NET 基于公共语言运行库，因此 Web 应用程序开发人员可以利用整个平台的威力和灵活性。.NET 框架类库、消息处理和数据访问解决方案都可从 Web 无缝访问。ASP.NET 也支持多种程序语言，所以可以选择最适合应用程序的语言，或跨多种语言分割应用程序。另外，公共语言运行库的交互性保证在迁移到 ASP.NET 时保留基于 COM 的开发中的现有投资。

- 简易性

ASP.NET 使执行常见任务变得容易。从简单的窗体提交和客户端身份验证到部署和站点配置都变得更加容易。例如，ASP.NET 页框架使用户可以生成将应用程序逻辑与表示代码清楚分开的用户界面，类似在 Visual Basic 的简单窗体处理模型中处理事件。另外，公共语言运行库利用托管代码服务（如自动引用计数和垃圾回收）简化了开发。

- 可管理性

ASP.NET 采用基于文本的分层配置系统，简化了将设置应用于服务器环境和 Web 应用程序的过程。由于配置信息是以纯文本形式存储的，因此可以在没有本地管理工具帮助的情况下应用新设置。此“零本地管理”哲学也扩展到了 ASP.NET 框架应用程序的部署。只需将必要的文件复制到服务器，即可将 ASP.NET 框架应用程序部署到服务器。不需要重新启动服务器，即使是在部署或替换运行的编译代码时。

- 可缩放性和可用性

ASP.NET 在设计时考虑了可缩放性，增加了专门用于在聚集环境和多处理器环境中提

高性能的功能。另外，进程受到 ASP.NET 运行库的密切监视和管理，以便当进程行为不正常（泄漏、死锁）时，可就地创建新进程，以保持应用程序始终可用于处理请求。

- 自定义性和扩展性

ASP.NET 随附了一个设计周到的结构，它使开发人员可以在适当的级别“插入”代码。实际上，可以用自己编写的自定义组件扩展或替换 ASP.NET 运行库的任何子组件。

- 安全性

借助内置的 Windows 身份验证和基于每个应用程序的配置，可以保证应用程序是安全的。

Microsoft.NET 的策略是将互联网本身作为构建新一代操作系统的基础，对互联网和操作系统的设计思想进行合理延伸。这样，开发人员必将创建出摆脱设备硬件束缚的应用程序，以便轻松实现互联网连接。Microsoft.NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

本书第 1 章给大家详细介绍了 ASP.NET 中用到的一些语法，并用 C#、VB、Jscript 三种语言分别描述了各自的语法特点和差别，在以后的章节里将从最简单的 Web 表单开始到窗体验证、服务器数据绑定、数据访问以及自定义控件等，每章的内容都是由浅入深，用大量的实例来讲解，全面阐述 ASP.NET 的 Web 应用。

本书由熊松明主编，由于作者水平有限，书中错误之处在所难免，希望广大读者能够不吝指正。

编 者

2002 年 8 月

目 录

第 1 章　ASP.NET 基本语法

安装 ASP.NET 不会破坏现有的 ASP 应用程序，它使用单独的文件扩展名（.aspx 而不是.asp）、单独的配置设置以及完全独立的公共语言运行库（Asp.dll 没有修改）。ASP 页和应用程序可以继续使用现有的 ASP 引擎，不会受 ASP.NET 的干扰。这表明，将现有应用程序移植到 ASP.NET 的好处是巨大的。ASP.NET 提供了比传统 ASP 多许多倍的功能，而且对 ASP 应用程序移植到新的平台也提供了极大的改进机会。可以利用的新功能包括：

- 提高的性能和可缩放性
- 网络场支持和 XCopy 部署
- 输出缓存和自定义安全
- Web 窗体页控件
- XML Web 服务基础结构

ASP.NET 旨在帮助用户保留在传统 ASP 和 COM 技术上的投资。用户希望支持现有的 ASP 语法和语义，同时需要可以很好地持续到下一代 Internet 应用程序开发的前瞻性平台，而 ASP.NET 正是在这两者间实现平衡。尽管 ASP.NET 保留了 ASP 功能集的多数内容，但如果平台发展，就不可能实现二者间的完全兼容，因此它会使以往的工作方式发生一些改动。

好的方面是用户的 ASP 技能可以很容易转换为 ASP.NET 技能。仅存在少数差异，通常很容易解决。但是，将 ASP 应用程序移植到 ASP.NET 确实需要做一些工作。相对简单的页可能不需更改即可移植，但较为复杂的应用程序可能需要一些修改。本章将描述这些更改，以及它们可能影响现有应用程序代码的方式，还将介绍一些可以在 ASP.NET 中重用 ASP 和 COM 代码的方法。

1.1　ASP.NET 与 ASP 的语法差异

从 ASP 到 ASP.NET 在 API 方面的兼容性会不会有所不同呢？二者的语法语义又会有什么变化呢？本节将介绍有关这方面的内容。

1.1.1　ASP.NET 与传统的 ASP 在 API 方面的兼容性

ASP.NET 与传统的 ASP 在 API 方面完全兼容，但有以下三处不同：

- Request()：ASP 返回字符串数组；ASP.NET 返回字符串。
- Request.QueryString()：ASP 返回字符串数组；ASP.NET 返回字符串。
- Request.Form()：ASP 返回字符串数组；ASP.NET 返回字符串。

在 ASP 中，Request、Request.QueryString 和 Request.Form 集合从查找返回字符串数组。例如，在传统的 ASP 中，按如下所示访问从请求到 http://localhost/test.asp?values=45&values=600 的查询字符串值。

```
<%
Response.Write Request.QueryString("values")
' 输出结果："45 600"
Response.Write Request.QueryString("values")(1)
' 输出结果："45"
%>
```

在 ASP.NET 中，这些集合需要显式方法来获取数组访问，这些数组现在也是从 0 开始索引。例如，在 ASP.NET 中，按如下所示访问从请求到 http://localhost/test.aspx?values=45&values=600 的查询字符串值。

```
//C#
<%
Response.Write Request.QueryString("values");
// 输出结果："45 600"
Response.Write Request.QueryString("values")(1);
// 输出结果："45"
%>
'-------------------------------------------------------------------------------
` VB
<%
Response.Write Request.QueryString("values")
' 输出结果："45 600"
Response.Write Request.QueryString("values")(1)
' 输出结果："45"
%>
'-------------------------------------------------------------------------------
//Jscript
<%
Response.Write Request.QueryString("values");
// 输出结果："45 600"
Response.Write Request.QueryString("values")(1);
// 输出结果："45"
%>
```

这些数组最常用于从多项选择列表框（<select multiple>）传递窗体值或多个复选框具有相同名称的情况。

1.1.2 ASP.NET 和 ASP 之间的语义差异

ASP.NET 页与现有的 ASP 页相比还有几处语义变化，主要表现在以下方面：

1. ASP.NET 页仅支持单语言

ASP 允许在单页上使用多种语言，这对脚本库方案有用。由于 ASP.NET 的已编译特性，所以它在一页上仅支持单语言。然而，在单个应用程序内仍然可以有多个使用不同语言的页。用户控件还可以具有不同于包含它们的页所使用的语言，这使读者能够在单页内集成用不同语言编写程序的功能，这足以替代传统 ASP 应用程序中普遍使用的多语言包含文件。

2. ASP.NET 页函数必须在<script runat=server>块中声明

在 ASP 中，页函数可以在<% %>块中声明。

```
<%
    Sub DoSomething()
        Response.Write "Hello World!"
    End Sub

    DoSomething
%>
```

在 ASP.NET 中，页函数必须在<script runat=server>块中声明。

```
// C#
<script language="C#" runat=server>

    void DoSomething() {
        Response.Write("Hello World!");
    }

</script>
```

```
' VB
<script language="VB" runat=server>

    Sub DoSomething()
        Response.Write ("Hello World!")
    End Sub

</script>

<%
    DoSomething()
%>
```

```
//Jscript
<%
    DoSomething();
%>
<script language="JScript" runat=server>

    function DoSomething() : void {
        Response.Write("Hello World!");
    }

</script>

<%
    DoSomething();
```

```
%>
```

3. ASP.NET 不支持页呈现函数

在 ASP 中，可以用<% %>块声明页呈现函数。

```
<% Sub RenderSomething() %>
        <font color="red"> Here is the time: <%=Now %> </font>
<% End Sub %>

<%
   RenderSomething
   RenderSomething
%>
```

在 ASP.NET 中，这必须重写。

```
<script language="VB" runat=server>
      Sub RenderSomething()
          Response.Write("<font color=red> ")
          Response.Write("Here is the time: " & Now)
      End Sub
</script>
<%
   RenderSomething()
   RenderSomething()
%>
<script language="C#" runat=server>
      void RenderSomething() {
          Response.Write("<font color=red> ");
          Response.Write("Here is the time: " + DateTime.Now);
      }
</script>
<%
   RenderSomething();
   RenderSomething();
%>

<script language="JScript" runat=server>
      function RenderSomething() : void {
          Response.Write("<font color=red> ");
          Response.Write("Here is the time: " + DateTime.Now);
      }
</script>
<%
   RenderSomething();
   RenderSomething();
%>
```

除以上三方面，ASP.NET 与传统的 ASP 在 API 方面完全兼容。

1.2　ASP.NET 的运行环境及配置

ASP 的运行需要一个环境，可以是 PWS，也可以是 IIS，而 ASP.NET 需要的是 NGWS。软硬件要求如下：

- 硬件要求（最低）：PII300，RAM 为 96MB。
- 软件要求：Windows 2000，IE 5.5，也许还要打个补丁。

安装完成后首先来看一下配置问题。ASP.NET 采用 XML 格式的文件 CONFIG.WEB 来进行配置（有点像 PHP 里的 PHP.INI），不同之处在于这是一个分级的配置结构，就是说在每个目录下都可以有一个 CONFIG.WEB 文件，在继承上级目录的所有配置的同时，可以给该级目录提供一些特殊需要的配置。这种结构将给用户带来以下一些便利：

- 配置信息存储在 XML 的配置文件中，管理员易于更新配置设定，而开发者则易于理解配置内容。
- 配置系统易于扩展，用户可以在配置系统中存放自己的配置标准以及设置。
- 对 ASP.NET 配置文件的更改由系统自动检测，系统管理员不需要重启系统以使改变生效。
- 分级配置，可以使不同的应用程序或单个应用程序的不同部分具有不同的设定。

下面是 MSDN 中提供的例子：

```
<!-- CONFIG.WEB FILE -->
<configuration>
<configsections>
 <add names="httpmodules"
type="System.Web.Config.HttpModulesConfigHandler"/
>
 <add names="httphandlers"
type="System.Web.Config.HttpHandlerConfigHandler"
/>
<add names="sessionstate"
type="System.Web.Config.SessionStateConfigHandler"
/>
<add names="globalization"
type="System.Web.Config.GlobalizationConfigHandle
r"/>
<!-- ADDITIONAL CONFIGSECTION DECLARATIONS GO HERE -->
</configsections>
<httpmodules>
 <!-- http module subelements go here -->
</httpmodules>
<httphandlers>
 <!-- http handlers subelements go here -->
</httphandlers>
<sessionstate>
  <!-- session state subelements go here -->
 </sessionstate>
```

```
<globalization>
<!-- session state subelements go here -->
</globalization>
<!-- ADDITIONAL CONFIGSECTIONS GO HERE -->
</configuration>
```

所有的配置信息都必须居于<configuration>和</configuration>标记之间。配置文件有两个主要部分：

（1）配置部分节的处理程序声明（包括在<configsections>和</configsections>标记中）。

（2）实际的配置小节（为了清楚，它们的子元素已被移除）。注意，下面的每一个配置小节都必须对应一个<configsections>中的声明存在。每一个声明赋予了配置小节名称，并且指出了将处理其配置信息的 NGWS Framework Assembly 及 Class。每一配置小节包含 ASP.NET 细节配置设定的内容。

如果用户已经安装了 NGWS，可以在 WINDOWS 目录下找到 CONFIG.WEB 文件，这是系统级的配置文件，可以浏览它的设置内容作为参考。

下面有几个例子，通过它们可以了解几个较为重要的设置的使用。

（1）<compilation debugmode="true"/>

在这里，调试模式被打开（设置成 True）。设置为 True 后可以使用 MS FRAMEWORK SDK 的 DEBUG 工具来调试代码中的错误。

（2）<globalization requestencoding="us-ascii"responseencoding="iso-8859-1" />

在 globalization 小节，设置了请求（Request）和回应（Response）的编码方式。

注意这里 response 的编码方式是 iso-8859-1，这样是无法显示中文的，将之改为 GB2312 以显示中文。

（3）<assemblies>

```
<add assembly="System.Data.dll"/>
</assemblies>
```

在 assemblies 小节，加入了一个 assemblies，这一设置使用户可以在程序中以<%@ import namespace="system.data"%>方式引用该类库。

（4）<security>

```
<security>
<authentication mode="Windows" />
</security>
```

在 security 小节，可以设置站点的验证方式。

这里将验证方式设定为 windows-based，也就是原来的 HTTP 验证。如果将 mode 设置为 cookie，则验证方式为 form-based。另外，还可以设置 mode="PASSPORT"（PASSPORT 验证），这需要安装 PASSPORT SDK。

（5）DNS 的设置

```
<appsettings>
<add key="MyConn" value="server=localhost;uid=sa;pwd=mypassword;
Database=somedatabase"/>
</appsettings>
```

这里定义了一个连接字符串“MyConn”，以供调用。

1.3　ASP.NET 的语言支持及语法

Microsoft.NET Platform 目前提供对以下三种语言的内置支持：C#、Visual Basic 和 Jscript，下面就向读者说明这三种语言的语法以及它们之间的语法差异。

1. 变量声明

```
//C#
int x;
String s;
String s1, s2;
Object o;
Object obj = new Object();
public String name;
'-------------------------------------------------------------------------
'VB
Dim x As Integer
Dim s As String
Dim s1, s2 As String
Dim o 'Implicitly Object
Dim obj As New Object()
Public name As String
'-------------------------------------------------------------------------
//Jscript
var x : int;
var s : String;
var s1 : String, s2 : String;
var o;
var obj : Object = new Object();
var name : String;
```

2. 语句

```
//C#
Response.Write("Hello,World");
'-------------------------------------------------------------------------
'VB
Response.Write("Hello,World ")
'-------------------------------------------------------------------------
//Jscript
Response.Write("Hello,World ");
```

3. 注释

```
//C#
// This is a super language
/*
```

```
    This
    is
    a
    super
    language
    */
'-------------------------------------------------------------------------------
    'VB
    ' This is a super language
    ' This
    ' is
    ' a
    ' super
    ' language
'-------------------------------------------------------------------------------
    //Jscript
    // This is a super language
    /*
    This
    is
    a
    super
    language
    */
```

4. 访问索引属性

```
    //C#
    String s = Request.QueryString("Name");
    String value = Request.Cookies("key");
'-------------------------------------------------------------------------------
    'VB
    Dim s, value As String
    s = Request.QueryString("Name")
    value = Request.Cookies("Key").Value
'-------------------------------------------------------------------------------
    //Jscript
    var s : String = Request.QueryString("Name");
    var value : String = Request.Cookies("key");
```

5. 声明索引属性

```
    //C#
    // 声明索引属性
    public String this(String name) {
        get {
            return (String) lookuptable(name);
        }
    }
```