



普通高等教育“十一五”国家级规划教材
高职高专计算机类专业规划教材

基于C#的 ASP.NET 程序设计

翁健红 主编

JIYU C# DE ASP.NET
CHENGXU SHEJI

 机械工业出版社
CHINA MACHINE PRESS



普通高等教育“十一五”国家级规划教材
高职高专计算机类专业规划教材

基于C#的ASP.NET 程序设计

主 编 翁健红
副主编 彭勇 马绍惠
参 编 孙桂芝 刘小华 刘志成

基于C#的ASP.NET程序设计\翁健红主编. 北京: 机械工业出版社, 2007.3.
ISBN 978-7-111-30728-8

I. 基… II. 翁… III. ①C语言—程序设计—教材 IV. TP393.095

中国版本图书馆CIP数据核字(2007)第019942号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑: 王五鑫 封面设计: 吕敬明 责任编辑: 张世琴 文字编辑: 刘文

北京京丰印刷厂印刷
2007年5月第1版·第1次印刷
184mm×260mm·16.75印张·415千字
0.001—4 000册

ISBN 978-7-111-30728-8
定价: 22.00元



凡购本书, 如蒙赠页, 请向
邮购部或发行部联系。电话: (010) 88379643
邮购部电话: (010) 88379643
发行部电话: (010) 88379643

机械工业出版社

ASP.NET 是 Web 开发的主流技术之一。本书使用 C# 作为 ASP.NET 开发语言, 主要介绍使用 ASP.NET 进行 Web 应用系统编程的方法。

本书共分 8 章, 分别是 ASP.NET 开发环境、C# 语言基础、服务器端控件、ADO.NET 数据库访问技术、VS.NET 开发会员管理系统、数据窗体设计、内置对象与应用程序配置、设计实例 (在线考试系统与新闻发布系统)。

本书内容丰富, 结构清晰, 叙述深入浅出, 适合作为高职高专院校计算机及相关专业 Web 应用系统设计的教材, 也可作为 ASP.NET 培训教材或从事 ASP.NET 编程和网站开发者的参考书。

为方便教师教学, 本书配有电子教案, 请发邮件至 wangyx@mail.machineinfo.gov.cn 索取。

图书在版编目(CIP)数据

基于 C# 的 ASP.NET 程序设计/翁健红主编. —北京: 机械工业出版社, 2007.2

ISBN 978-7-111-20778-8

I. 基… II. 翁… III. ①C 语言-程序设计②主页制作-程序设计 IV. TP312 TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 019945 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 王玉鑫 孔喜峻

责任编辑: 李学峰 版式设计: 张世琴 责任校对: 刘志文

封面设计: 马精明 责任印制: 洪汉军

北京京丰印刷厂印刷

2007 年 2 月第 1 版 · 第 1 次印刷

184mm × 260mm · 16.75 印张 · 412 千字

0 001—4 000 册

标准书号: ISBN 978-7-111-20778-8

定价: 25.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 68354423

封面防伪标均为盗版

前言

ASP.NET 是 Web 开发的主流技术之一。本书使用 C# 作为 ASP.NET 开发语言, 主要介绍使用 ASP.NET 进行 Web 应用系统编程的方法。

本书共分 8 章, 分别是 ASP.NET 开发环境、C# 语言基础、服务器端控件、ADO.NET 数据库访问技术、VS.NET 开发会员管理系统、数据窗体设计、内置对象与应用程序配置、设计实例 (在线考试系统与新闻发布系统)。

本教材有如下特色:

1. 贯彻够用为度

在知识点上, 不求多和深。根据作者对 ASP.NET 的透彻了解以及广泛的市场调查, 借鉴现有的各种 IT 教学体系, 严格把握“够用”原则, 以免学生困难而却步, 因多而分心。

2. 合理讲解单文件的 Web Form 方式与基于 VS.NET 的代码分离方式

本书以单文件的 Web Form 方式为主讲解基本的原理并简单地举例, 适当地讲解了 VS.NET 的使用并以 VS.NET 为工具开发项目案例。而现在的教材往往要么只讲单文件的 Web Form 方式, 要么只讲在 VS.NET 平台下的开发。但是, 单文件的 Web Form 方式便于学生理解原理但不便于操作, 与实际的开发方式有一段距离; VS.NET 虽是实际软件开发的主要平台, 但对初学者来说, 容易限于简单琐碎的操作却不透彻理解原理。

3. 与案例结合, 强调项目、模块开发

本书使用了“在线投票系统”、“会员管理系统”、“在线考试系统”与“新闻发布系统”四个精炼的案例。与一般教材把案例放在最后章节不同, 本书强调案例为内容服务, 在学生有足够知识的情况下, 及时引入案例。例如第 5 章安排“会员管理系统”, 第 6 章引入“在线投票系统”, 第 8 章综合讲述“在线考试系统”与“新闻发布系统”, 这样既强化了所学的知识, 又提高了学生兴趣。

在案例的组织上, 引入目标—编程的方式, 即在把案例分解为模块的基础上, 对每个模块先以“模块设计”说明目标; 然后讲解实现这个目标的步骤。这就避免了案例的讲解变成代码的堆积, 学生无论学习, 还是上机练习都可以带着明确的目标。这种方式与现实开发非常接近。

4. 一线教师编写, 定位准确, 符合教学规律

本书作者是长期在第一线从事教学、培训工作的高校教师, 有丰富的教学经验、理论水平和项目开发经验。

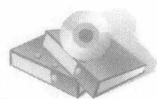
本书由翁健红任主编, 彭勇、马绍惠任副主编。具体分工如下: 第 1 章由刘志成编写, 第 2 章、第 6 章、第 7 章由翁健红编写, 第 3 章由马绍惠编写, 第 4 章由刘小华编写, 第 5 章由孙桂芝编写, 第 8 章由彭勇编写。

由于时间仓促, 作者水平有限, 书中不足与疏漏之处在所难免, 敬请广大读者批评指正。

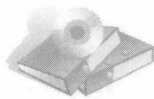
编者

目 录

前言	33
第 1 章 ASP.NET 开发环境	1
1.1 Web 基础知识	1
1.1.1 Web 服务器	1
1.1.2 静态网页	1
1.1.3 动态网页	2
1.2 配置 ASP.NET 的运行环境	3
1.2.1 ASP.NET 的运行环境	3
1.2.2 安装 IIS	3
1.2.3 安装 .NET Framework	5
1.2.4 测试 ASP.NET 环境	5
1.2.5 创建虚拟目录	6
1.2.6 第一个 Web Form 程序	8
1.3 ASP.NET 页面结构	10
1.4 ASP.NET 页面事件	11
习题 1	13
第 2 章 C# 语言基础	14
2.1 C# 语言简介	14
2.2 变量	14
2.3 数据类型	15
2.4 运算符	17
2.5 程序流程控制	20
2.5.1 选择语句	20
2.5.2 循环语句	23
2.5.3 跳转语句	26
2.6 方法	27
习题 2	28
第 3 章 服务器端控件	30
3.1 常用 Web Server 控件	30
3.1.1 Web 控件的共用属性	31
3.1.2 Label 控件	32
3.1.3 Image 控件	32
3.1.4 HyperLink 控件	32
3.1.5 Button 控件	33
3.1.6 LinkButton 控件	35
3.1.7 ImageButton 控件	36
3.1.8 TextBox 控件	37
3.1.9 Table、TableRow 及 TableCell 控件	38
3.1.10 Panel 控件	40
3.1.11 RadioButton 控件	42
3.1.12 RadioButtonList 控件	43
3.1.13 CheckBox 控件	46
3.1.14 CheckBoxList 控件	47
3.1.15 DropDownList 控件	49
3.1.16 ListBox 控件	50
3.2 数据验证控件	53
3.2.1 RequiredFieldValidator 控件	53
3.2.2 CompareValidator 控件	54
3.2.3 RangeValidator 控件	56
3.2.4 RegularExpressionValidator 控件	57
3.2.5 CustomValidator 控件	61
3.2.6 ValidationSummary 控件	62
3.3 服务器端其他控件	64
3.3.1 AdRotator 控件	64
3.3.2 Calendar 控件	65
3.3.3 Literal 控件	69
3.3.4 Placeholder 控件	69
3.4 HTML 控件简介	70
3.5 用户控件	71
3.5.1 用户控件的建立与使用	71
3.5.2 将 Web 窗体页转换为用户控件	72
习题 3	72
第 4 章 ADO.NET 数据库访问技术	74
4.1 ADO.NET 基本概念	74
4.2 ADO.NET 对象模型	74
4.3 OleDbConnection 对象	76



4.4 Command 对象	78	5.4.1 方法、属性、变量命名规范	145
4.4.1 Command 对象的属性和方法	78	5.4.2 Web 控件的命名	146
4.4.2 增加、删除、修改数据	79	5.4.3 代码书写规范	146
4.4.3 执行带参数的 Command	82	5.4.4 注释	146
4.4.4 ExecuteScalar 方法	85	5.4.5 不规范代码实例	147
4.5 DataReader 对象	85	5.5 安装	148
4.6 DataSet 对象	87	习题 5	150
4.7 DataTable 对象	90	第 6 章 数据窗体设计	151
4.8 DataAdapter 对象	92	6.1 数据绑定的概念和方法	151
4.9 DataView 对象	95	6.1.1 绑定到变量	151
4.10 连接 SQL Server 数据库	97	6.1.2 绑定到方法	152
4.10.1 SqlConnection	97	6.1.3 绑定到数组	153
4.10.2 执行存储过程	102	6.1.4 绑定到集合和列表	154
4.11 ADO.NET 中使用 XML		6.2 CreateDataSet.inc 文件	155
文件	106	6.3 Repeater 控件	155
习题 4	108	6.4 DataList 控件	159
第 5 章 VS.NET 开发会员管理系统	109	6.4.1 DataList 控件的语法格式	159
5.1 VS.NET 的开发环境	109	6.4.2 编辑 DataList 项目	162
5.1.1 使用 VS.NET 创建 ASP.NET	109	6.4.3 在 DataList 中选择项目	167
应用程序	109	6.4.4 VS.NET 中 DataList 的模板编辑	169
5.1.2 VS.NET 项目中的文件	113	6.5 DataGrid 控件	170
5.1.3 Design 视图和 HTML 视图	114	6.5.1 一个简单的例子	170
5.1.4 ASP.NET 的代码后置	114	6.5.2 DataGrid 的语法格式	171
5.1.5 理解自动生成的代码	114	6.5.3 DataGrid 的列类型	172
5.1.6 VS.NET 中的表格操作	118	6.5.4 使用样式	185
5.1.7 VS.NET 中添加窗体	119	6.5.5 DataGrid 的排序	187
5.2 会员管理系统	120	6.5.6 DataGrid 的分页	190
5.2.1 系统分析与设计	120	6.6 在线投票系统	193
5.2.2 系统的运行界面	121	6.6.1 系统分析	193
5.2.3 登录页面	124	6.6.2 投票页面	193
5.2.4 主页面	127	6.6.3 显示投票结果页面	196
5.2.5 注册页面	128	习题 6	199
5.2.6 显示个人信息页面	131	第 7 章 内置对象与应用程序配置	200
5.2.7 修改个人信息页面	133	7.1 Application 对象	200
5.2.8 取回口令 (1) 页面	137	7.1.1 Application 对象的	
5.2.9 取回口令 (2) 页面	139	生命周期	200
5.3 程序排错与调试	140	7.1.2 Application 对象的读写	200
5.3.1 错误类型	140	7.1.3 Application 对象的锁定	200
5.3.2 使用调试器	141	7.1.4 Application 对象应用举例	201
5.3.3 调试实例	143	7.2 Session 对象	202
5.4 编程规范	145	7.2.1 Session 对象与 Application	



对象的区别	202
7.2.2 Session 对象的生命周期	202
7.2.3 Session 对象的读写	202
7.2.4 Session 对象应用举例	203
7.3 Request 对象	203
7.3.1 Request 对象的常用属性和方法	203
7.3.2 Request 对象应用举例	204
7.4 Response 对象	205
7.4.1 Response 对象的常用属性 和方法	205
7.4.2 Response 对象应用举例	206
7.5 Server 对象	207
7.5.1 Server 对象的常用属性和方法	207
7.5.2 Server 对象应用举例	207
7.6 Cookie 对象	210
7.6.1 Cookie 对象的特点	210
7.6.2 Cookie 对象的读写	210
7.6.3 Cookie 对象的生存期	210
7.6.4 Cookie 对象验证的安全性	210
7.6.5 Session 对象和 Cookie 对象的关系	210
7.6.6 Cookie 对象应用举例	211
7.7 Global.asax 文件	213
7.8 Web.config 配置文件	214
7.8.1 配置文件的结构	214
7.8.2 <system.web> 配置段	215

7.8.3 应用程序设置	217
7.8.4 配置文件的继承关系	218
习题 7	219
第 8 章 设计实例	220
8.1 在线考试系统	220
8.1.1 系统分析与设计	220
8.1.2 系统运行界面	221
8.1.3 Web.config 文件	223
8.1.4 数据库操作的公共类	223
8.1.5 考生登录页面	226
8.1.6 考试页面	228
8.1.7 成绩显示页面	231
8.2 新闻发布系统	232
8.2.1 系统分析与设计	232
8.2.2 系统的运行界面	233
8.2.3 Web.config 与公用类	236
8.2.4 浏览新闻页面	236
8.2.5 显示新闻页面	238
8.2.6 添加新闻页面	240
8.2.7 新闻列表页面	243
8.2.8 管理新闻页面	248
8.2.9 修改新闻页面	253
习题 8	255
附录 参考实验	256
参考文献	260



图 1-1 示例代码

(图 1-1 所示代码，将文本“你好”显示在网页上)

第 1 章 ASP.NET 开发环境

```
<html>
<body>
<div style="text-align: center;">
<div>
</div>
</div>
</body>
</html>
```

学习目标

- 1) 了解 Web 基础知识。
- 2) 掌握 ASP.NET 运行环境的配置。
- 3) 了解 ASP.NET 页面的结构。
- 4) 了解 ASP.NET 页面的事件。

1.1 Web 基础知识

HTTP 即超文本传输协议 (Hyper Text Transfer Protocol)，是在 Internet 中进行信息传送的协议，浏览器默认使用 HTTP。例如，当用户在浏览器的地址栏中输入 www.163.com 的时候，浏览器会自动使用 HTTP 来搜索 http://www.163.com 网站的首页。

从浏览器向 Web 服务器发出的搜索某个 Web 网页的请求叫做 HTTP 请求。当 Web 服务器收到这个请求之后，就会按照请求的要求，找到相应的网页。如果可以找到这个网页，那么就把网页的 HTML 代码通过网络传回浏览器；如果没有找到这个网页，就发送一个错误信息给浏览器。后面的这些操作称为 HTTP 响应。

1.1.1 Web 服务器

当提到 Web 服务器的时候，很多人都会认为这是一台物理的机器。但实际上，Web 服务器是一种软件，可以管理各种 Web 文件，并为提出 HTTP 请求的浏览器提供 HTTP 响应。大多数情况下，Web 服务器和浏览器处于不同的机器，但是它们也可以并存在同一机器上。比较常见的 Web 服务器有 Apache 和 IIS。IIS 是微软公司的 Windows 操作系统所提供的，由于 ASP.NET 只能在 IIS 上运行，因此在后面的章节将介绍 IIS。

1.1.2 静态网页

www 通过超链接 (Hyper Link) 技术在位于不同位置的文件之间建立了连接，从而可以为用户提供一种交叉式而非线式的访问方式。借助这种更符合思维习惯的访问方式，人们可以十分便捷地访问各种资源 (如文本信息、多媒体信息等)。HTML (Hyper Text Markup Language) 是一种标记语言，用于声明信息 (如文本、图像等) 的结构、格式，标识超链接等。在文本中嵌入适当的 HTML 标记后所得到的文件称为 HTML 文档。静态网页就是用纯 HTML 代码编写的网页。这些网页的代码是用一些编辑器输入的，或者是用一些网页设计程序生成的，保存为 .html 或 .htm 文件的形式。设计完成之后，无论是何人、何时，以何种方式进入这个网页，它的显示都不会发生任何变化。

【例 1-1】下面是一个静态网页的例子，用于显示一个红色的“你好”字符串，运行结



果如图 1-1 所示。

Hello.html (横线上加深字体表示文件名, 以后不再一一介绍)

```
<html>
<body>
<font color = red> 你好 </font>
</body>
</html>
```



图 1-1 一个简单的静态网页

仅由 HTML 页面构成的 Web 应用程序的内容是静止的, 它不会对用户的动作做出动态响应。采用静态网页会导致很大的局限性, 如果希望为用户显示一些个性化的信息, 使用静态网页就无法达到目的。例如, 如果当前的时间是新年的开始, 就在网页的最上面显示一个“新年好!”的信息; 如果当前的时间是圣诞节, 就在同样的位置显示一个“圣诞快乐!”的信息。

1.1.3 动态网页

要对用户请求做出动态响应, 就要使用动态网页, 动态网页可以为不同的用户提供个性化的服务, 而为了实现这种动态性, 就需要进行程序设计。

Web 应用程序要么在 Web 服务器 (作为服务器端脚本时) 上执行, 要么在 Web 浏览器 (作为客户端脚本时) 上执行, 它们使你能通过 Internet 和企业网共享、访问信息。另外, Web 应用程序支持在线商务事务, 也就是一般所说的电子商务。

客户端脚本使你能开发出动态响应用户输入的 Web 页面, 而无需和 Web 服务器进行交互。例如, 有一个 Web 应用程序要求用户输入用户名和口令方可显示主页, 因此用户名和口令部分不能为空白。要检查用户名和口令部分是否为空白, 你可以编写运行于用户计算机上的客户端脚本。客户端脚本有助于减少网络流量, 因为它不需要和 Web 服务器进行交互作用以动态响应用户的输入。脚本语言 (如 VBScript 和 JavaScript) 被用于编写客户端脚本。

服务器端脚本为用户提供动态内容, 这些内容是基于远程存储信息 (如后台数据库) 的。它包括用服务器端脚本语言 ASP (Active Server Pages) 和 JSP (Java Server Pages) 所写的



代码。服务器端脚本要在 Web 服务器上执行,当浏览器请求包含服务器端脚本和 HTML 页面时,收到请求的 Web 服务器先处理脚本,然后将结果发送给浏览器,这样任何来自 Web 服务器的数据都会用服务器端脚本进行处理。例如,用 Web 页面显示 Web 站点所在系统当前时间就需执行服务器端脚本。如果使用了客户端脚本的话,那么每个请求包含该脚本的文件的浏览器都会显示浏览器所在系统的当前时间。

ASP.NET 是 ASP 的 NET 版本,可以创建动态 Web 页面。ASP.NET 提供的服务器脚本有下列优点:

- 1) ASP.NET 能从数据源(如后台数据库和存储于 Web 服务器上的文本文件)访问数据。
- 2) ASP.NET 能使用被称为模板的编程代码集合来创建 HTML 文档。使用模板的优点是你可以从 HTML 文档显示给用户之前,将从数据源检索到的内容动态插入到 HTML 文档中。所以,在从数据源检索到的内容发生变化时不需要手动修改信息。
- 3) ASP.NET 能将 HTML 的设计和数据库检索机制相分离。这样,改变 HTML 设计不会影响数据库应用程序。类似的,服务器脚本确保了对数据源的修改不要求对 HTML 文档作出改动。

除了上述优点,ASP.NET 还有许多高级功能,帮助你开发 Web 应用程序。

1.2 配置 ASP.NET 的运行环境

1.2.1 ASP.NET 的运行环境

1. 软件环境

- 1) 操作系统。Windows 2000 Professional/Server、Windows XP Professional 或已安装 Service Pack 的 Windows NT 4.0。
- 2) 服务软件。Internet Information Services (IIS) 5.0、NET Framework、MDAC (Microsoft Data Access Components)。此外,如果有 Visual Studio.NET 套件,则只需在 IIS 5.0 基础上安装此套件即可。
- 3) www 客户。Internet Explorer 5.5 或 6.0。

2. 硬件环境

对运行 ASP.NET 的计算机而言,硬盘及内存越大越好,其中内存最好在 256MB 以上。

1.2.2 安装 IIS

如果操作系统是 Windows 2000/2003 Server, IIS 5.0/6.0 是默认安装上的。如果是 Windows 2000/XP Professional,则需要安装 IIS 5.0。下面以 Windows XP 操作系统为例说明 IIS 的安装步骤。

- 1) 依次打开“控制面板”→“添加/删除程序”→“添加/删除 Windows 组件”,勾选“Internet 信息服务 (IIS)”,如图 1-2 所示。然后单击“详细信息”按钮。
- 2) 在打开的“Internet 信息服务 (IIS)”窗口中选择所有选项,如图 1-3 所示。

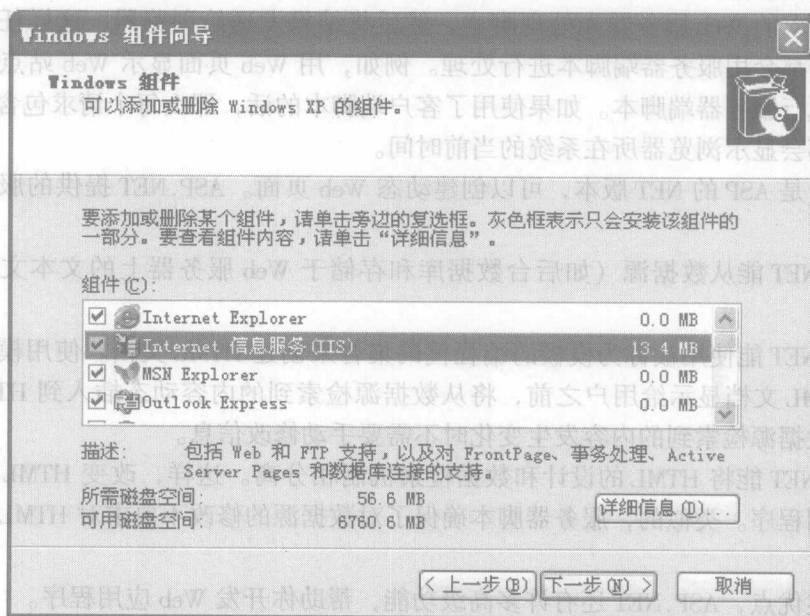


图 1-2 选择添加“Internet 信息服务 (IIS)”

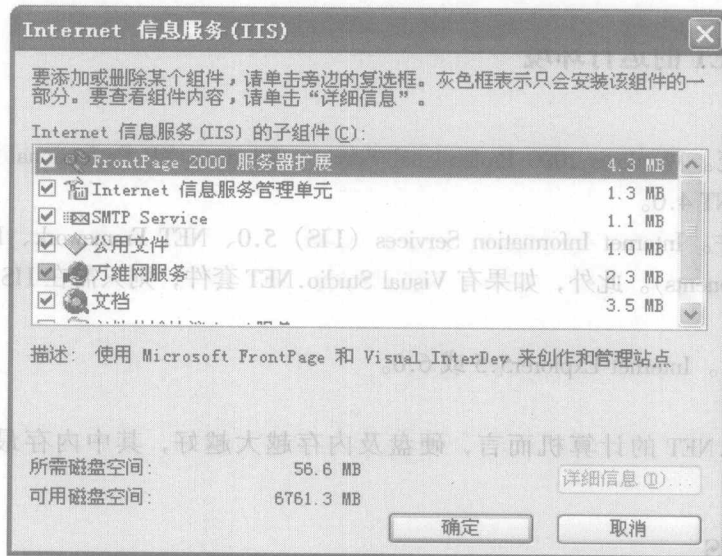


图 1-3 “Internet 信息服务 (IIS)” 窗口

3) 单击“确定”按钮后，再单击“下一步”按钮。在光驱中放入 Windows XP 的安装盘，安装完毕后，可以测试一下安装是否成功，在浏览器的地址栏中输入“http://localhost”；如果安装成功的话，将会出现欢迎画面，如图 1-4 所示。

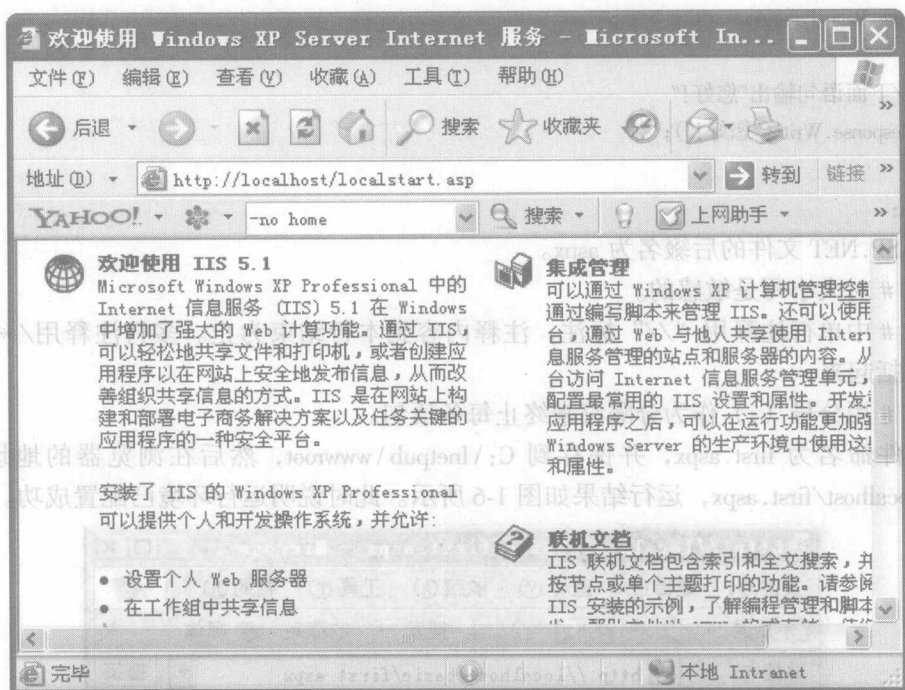
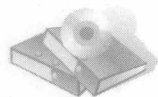


图 1-4 欢迎画面

1.2.3 安装 .NET Framework

安装完 IIS 以后, 为了支持 ASP.NET 脚本, 还必须安装 .NET Framework, 最新的版本 (Microsoft .NET Framework 1.1 版可再发行组件包 .exe) 可以在微软的网站下载。运行组件包, 如图 1-5 所示, 接下来的步骤按默认选项进行安装。

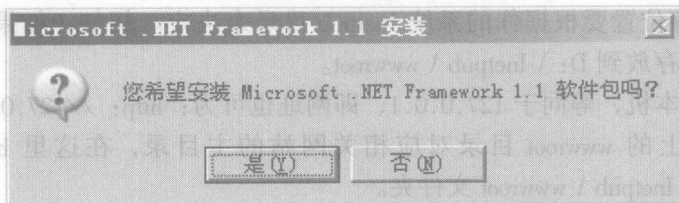


图 1-5 安装 .NET Framework

注意: 一定要先安装 IIS, 再安装 .NET Framework。

1.2.4 测试 ASP.NET 环境

安装了 .NET Framework, 系统就具备了运行 ASP.NET 的环境。我们可以通过运行一个简单的 ASP.NET 程序来测试。

【例 1-2】 建立一个简单的 ASP.NET 程序, 在记事本中输入下面程序:

```
first.aspx
```



```
<% @ Page Language="C#" %>
```

```
<%
```

```
//下面语句输出"您好!"
```

```
Response.Write("您好!");
```

```
%>
```

注意:

1) ASP.NET 文件的后缀名为 `aspx`。

2) C# 对大小写是敏感的。

3) C# 中单行注释用“//”标注,注释内容到本行结束为止。多行注释用 `/* ... */` 把注释内容括起来。

4) C# 用分号(;)作为分隔符来终止每条语句。

把文件命名为 `first.aspx`,并保存到 `C:\Inetpub\wwwroot`,然后在浏览器的地址栏输入 `http://localhost/first.aspx`,运行结果如图 1-6 所示。此时说明运行环境已配置成功。



图 1-6 测试 ASP.NET 环境

说明:

1) 文件的保存位置要根据你的系统 Inetpub 目录来决定。例如,如果 Inetpub 目录在 D 盘根目录下,则应存放到 `D:\Inetpub\wwwroot`。

2) localhost 指本机,等同于 127.0.0.1,即网址也可为: `http://127.0.0.1/first.aspx`。

3) 物理磁盘上的 `wwwroot` 目录对应相关网站的主目录,在这里 `http://127.0.0.1/first.aspx` 对应 `C:\Inetpub\wwwroot` 文件夹。

1.2.5 创建虚拟目录

要显示动态页面,需要把动态页面文件放在某个虚拟目录下。下面以 Windows 2000 的 IIS5.0 为例,在 `C:\TestWeb` 目录上创建一个名为 `MyWeb` 的虚拟目录(首先应确保 `C:\TestWeb` 目录已存在)。创建虚拟目录的步骤如下:

1) 单击“开始菜单”→“程序”→“管理工具”→“Internet 信息服务 (IIS) 管理器”,启动 IIS,如图 1-7 所示。

2) 在“默认网站”上用鼠标右击,选择“新建”→“虚拟目录”;在弹出的窗口中单击“下一步”按钮。

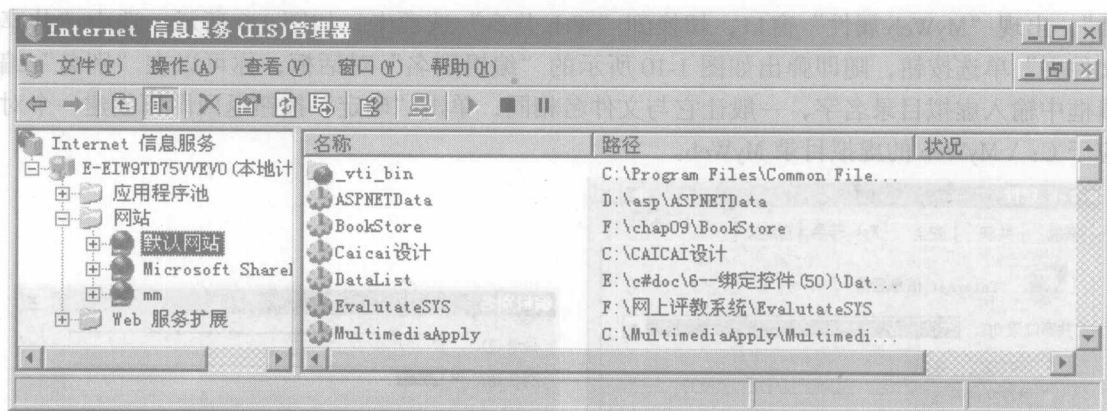


图 1-7 Internet 信息服务 (IIS) 管理器

3) 输入目录名称“MyWeb”，如图 1-8 所示，然后单击“下一步”按钮。

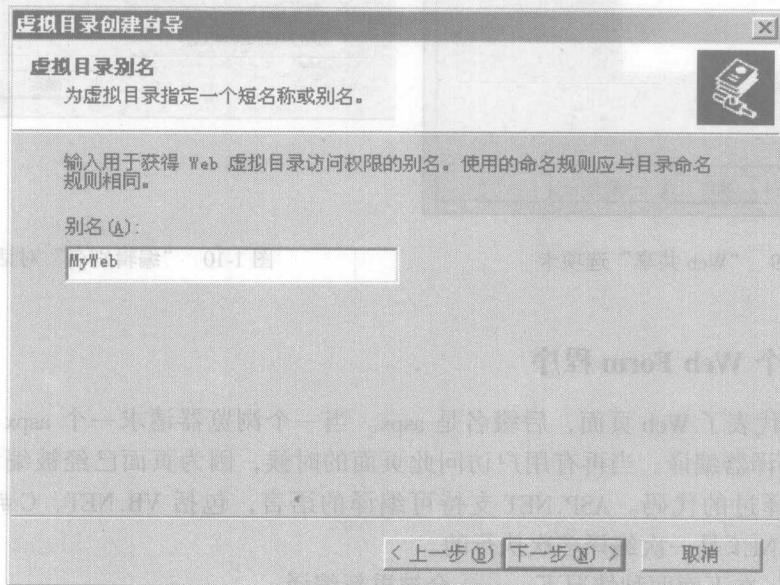


图 1-8 输入目录名称“MyWeb”

4) 输入目录路径“C: \ TestWeb”，其余无须更改，一直单击“下一步”按钮就可以完成虚拟目录的创建。

创建好虚拟目录后，我们只要把 ASP.NET 程序放到虚拟目录中，就可以在网址中浏览。网址基本路径为：<http://localhost/MyWeb/>。例如，把前面创建的 first.aspx 复制到 C: \ TestWeb 下，然后在浏览器的网址栏输入“<http://localhost/MyWeb/first.aspx>”，即可显示如图 1-6 所示的运行结果。

提示：我们也可以用快捷的方法创建虚拟目录。例如，要把 C: \ MyWeb 文件夹设为虚拟目录，只要在资源浏览器中，用鼠标右击 C: \ MyWeb 文件夹，选择弹出菜单中的“属



性”，出现“MyWeb 属性”窗口，切换到“Web 共享”选项卡，如图 1-9 所示。选中“共享文件夹”单选按钮，随即弹出如图 1-10 所示的“编辑别名”对话框，你可以在“别名”编辑框中输入虚拟目录名字，一般让它与文件名相同，单击“确定”按钮返回即可创建一个对应于 C:\MyWeb 的虚拟目录 MyWeb。

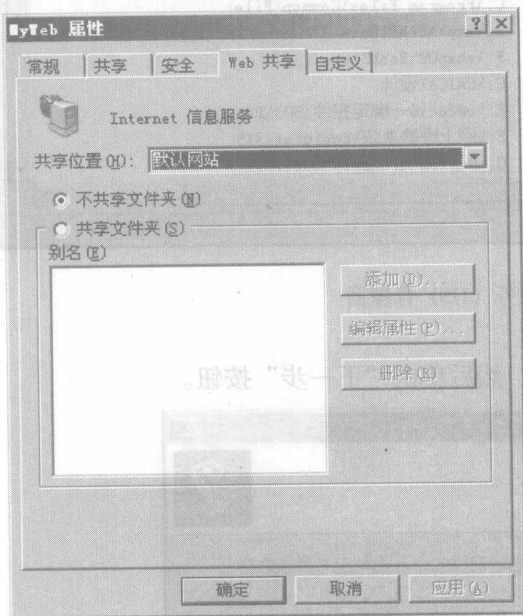


图 1-9 “Web 共享”选项卡

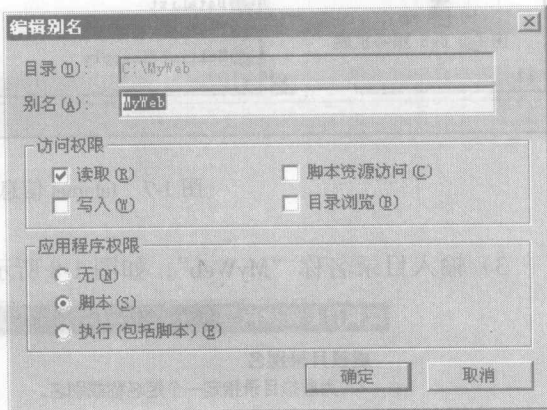


图 1-10 “编辑别名”对话框

1.2.6 第一个 Web Form 程序

Web Form 代表了 Web 页面，后缀名是 aspx。当一个浏览器请求一个 aspx 文件时，Web Form 被 CLR 编译器编译。当再有用户访问此页面的时候，因为页面已经被编译，所以 CLR 会直接执行编译过的代码。ASP.NET 支持可编译的语言，包括 VB.NET、C#、Jscript.NET 等，所以 ASP.NET 是一次编译多次执行的。

一般来说，在下面两种情况下，aspx 会被重新编译：

- 1) aspx 页面第一次被浏览器请求。
- 2) aspx 文件被改写。

因为 Web Form 第一次被浏览器请求要进行编译，所以开始速度较慢，但以后访问速度就可以很快。

【例 1-3】 下面程序在用户输入姓名并单击“确定”按钮后，将显示欢迎信息，页面如图 1-11 所示。

WebForm.aspx

```
< % @ Page Language = "C#" % >

< script runat = "server" >
protected void Button_Click(object sender, EventArgs e){
    Response.Write("欢迎您:" + TextBox1.Text);
}
```



```
</script>
```

```
< form method = "post" runat = "server" >
```

```
    姓名: < asp:textbox width = "100" id = "TextBox1" runat = "server" />
```

```
    < asp:button text = "确定" onclick = "Button_Click" runat = "server" />
```

```
</form>
```

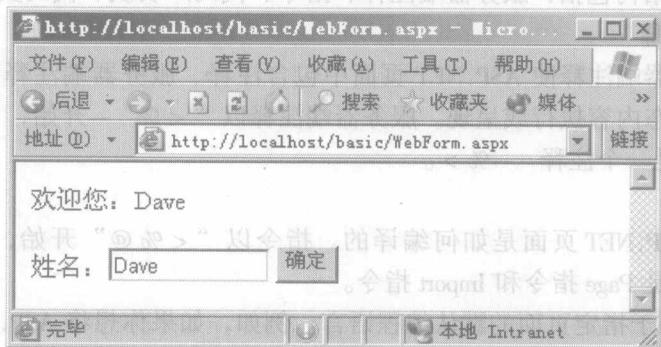


图 1-11 一个简单的 Web Form

说明:

1) 在例 1-3 中, 下列语句分别表示编辑控件与按钮:

```
< asp:textbox width = "100" id = "TextBox1" runat = "server" />
```

```
< asp:button text = "确定" onclick = "Button_Click" runat = "server" />
```

2) 下面代码输出欢迎信息:

```
protected void Button_Click(object sender, EventArgs e) {
    Response.Write("欢迎您:" + TextBox1.Text);
}
```

Button 控件中的 onclick = "Button_Click" 指定该控件被单击时, 应该执行名为 Button_Click 的函数; 因此, 当用户单击“确定”按钮时, 系统自动调用 Button_Click 函数, 该函数根据用户的输入, 返回欢迎信息; 其中 "TextBox1.Text" 返回编辑控件 TextBox1 中用户的输入。

3) 从这个程序看出, Web Form 的一些基本特点如下:

① Web 控件的标记前缀是 asp:, 然后指定 Web 控件的类别名称, 如 asp:textbox 和 asp:button。

② 指定 runat 属性为 server, 表示控件是在 Server 端执行, 是一个服务器控件。

③ 控件的 id 相当于控件的名字。如果要在代码中引用控件, 就要给控件一个 id 属性, 如 textbox 控件的 id 为 TextBox1; 而 Button 控件在代码中没用到, 可以没有 id; 但通常大多数服务器控件都有 id。

④ Form 标记在网页内形成“表单”, 对于 Form 标记, 要注意以下几点:

- 在 Web 网页中, 如果 Form 内有服务器控件, 则 Form 标记中一定要加上 runat = "server"。



- 一个页面只能有一个 Form 表单。
- 大多数服务器控件必须放在 Form 表单内。

1.3 ASP.NET 页面结构

ASP.NET 页面结构包括：服务器端注释、指令、代码声明块、代码实施块。

1. 服务器端注释

可以使用服务器端注释使 ASP.NET 页面中包含注释、服务器端注释说明文档，也可防止服务器代码和静态内容执行或呈现。服务器端注释以“<%”开始，以“%>”结束。例如，<%-- 这是一个注释 --%>。

2. 指令

指令指定了 ASP.NET 页面是如何编译的。指令以“<%@”开始，以“%>”结束。最频繁使用的指令是 Page 指令和 Import 指令。

1) Page 指令用于指定页面的默认编程语言。例如，如果你想将 ASP.NET 页面的编程语言指定为 C#，你可用 Page 指令的 language 属性，如下所示：

```
<%@ Page Language="C#" %>
```

2) Import 指令用于给 Web 应用程序引入另外的命名空间。例如，如果你想从 ASP.NET 页面访问数据库，那么需要显式导入 System.Data 名称空间，你需要使用如下 Import 指令：

```
<%@ Import Namespace="System.Data" %>
```

3. 代码声明块

可以在 Web 应用程序的代码后置文件或 aspx 文件中添加应用程序逻辑。如果你想在 aspx 文件中添加应用程序逻辑，你需要在代码声明块中编写应用程序逻辑。代码声明块包含 ASP.NET 页面的应用程序逻辑、全局变量声明和函数。在 aspx 文件指定一个代码声明块如下：

```
<script Language="C#" runat="server"> //代码声明块开始
```

```
void ShowMsg() //名为 ShowMsg 的用户定义方法
```

```
Response.Write("ASP.NET 教程");
```

```
</script> //代码声明块结束
```

其中 Language="C#" 指定 <Script> 标记中使用的是 C# 语言。

4. 代码实施块

如果要执行 ASP.NET 页面的 HTML 内容里的代码，就需要使用代码实施块。有两种类型的代码实施块：Inline code 和 Inline expression。Inline code 以“<%”开始，以“%>”结束，用于执行一条语句或一系列语句。Inline expressions，以“<%=”开始，以“%>”结束，用于显示变量或方法的值。

【例 1-4】下面程序演示了 Inline expression 和 Inline code 两种代码实施块的使用，运行结果如图 1-12 所示。

Inline.aspx