



随书光盘附带
本书所需源代码

ASP.NET

技术详解与应用实例

李万宝 主编



机械工业出版社
CHINA MACHINE PRESS

ASP.NET 技术详解与应用实例

李万宝 主编



机械工业出版社

本书详细介绍了利用 ASP.NET 开发网页的基本技术。全书共分 15 章, 其中前 12 章介绍了 Web 应用程序开发知识, 不仅包含 ASP.NET Web 运行原理、基本控件和验证控件的使用, 还包含 XML、数据访问、应用程序配置、跟踪调试、网络安全和系统优化等内容。第 13 章介绍了简单 ASP.NET Web 服务的开发。第 14 章通过对实例 OnNetBookSell 的分析, 介绍了如何综合使用 ASP.NET 的各个基本知识点并结合 3 层体系设计思想构建 B/S 电子商务。第 15 章通过对 DotNetForum 的分析, 介绍了如何充分利用 .NET 技术, 并结合 N 层结构设计思想开发较为复杂的网络论坛。

本书结构清晰、内容丰富, 无论是对于 ASP.NET 初学者, 还是具有一定经验的 ASP.NET 使用者, 都具有一定的指导参考价值。

图书在版编目 (CIP) 数据

ASP.NET 技术详解与应用实例/李万宝主编.

-北京: 机械工业出版社, 2005.9

ISBN 7-111-17427-5

I. A… II. 李… III. 主页制作-程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2005) 第 108094 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: 吴宏伟 责任编辑: 李虎斌 版式设计: 崔俊利

北京中兴印刷有限公司印刷

2005 年 9 月第 1 版第 1 次印刷

787mm×1092mm 1/16·32.5 印张·800 千字

0001-5000 册

定价: 46.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话: (010) 68326294

封面无防伪标均为盗版

前 言

ASP.NET 技术是 Microsoft 公司推出的新一代基于 B/S 的动态 Web 开发工具，是今后电子商务、电子政务以及远程教育等 Web 应用开发的主流技术之一。

虽然它继承了原 ASP 的特点，但是在语法和应用上都和 ASP 有了很大的区别。鉴于此，笔者根据自己的工作经验，在借鉴了许多同类书籍和网站资料的基础上，编写了本书。

在撰写本书时，笔者试图表述以下 3 个观点：

观点一：与其一门心思钻入枯燥的 ASP.NET 基本语法和 HTML 语言，不如借助一个可视化创作工具来提高效率；而与其依赖可视化创作工具并不成熟和并不完美的功能，还不如在可视化创作工具的指引下，深入学习 ASP.NET 语法和 HTML 语言。

Visual Studio.NET 正好是这种学习思路的最好体现，它的可视化工具让我们拥有完全对代码进行直接控制的能力，同时通过代码又可以控制页面运行。

当然，考虑到 ASP.NET 的复杂性，也可以通过利用基本的字处理软件（记事本）介绍 ASP.NET 的基本知识。

观点二：在掌握了 ASP.NET 基础知识的应用上，毫无疑问有一个例子是最好的。这个例子不能过于简单，太简单将不能体现 ASP.NET 的特点；同时也不能太复杂，太复杂又不易于介绍；同时还要求实用、典型。

也许微软在开发 ASP.NET 时，就已经考虑到了这个问题，它在发布 ASP.NET 时就提供了 5 个例子。这 5 个例子包括了 ASP.NET 在 Windows 开发的各个方面，其中以 IBuySpy.com 最为典型。因此，本书以对 IBuySpy.com 的二次开发作为例子。在二次开发中仍然保持原例的特点，同时使其更为实用。

通过该例子介绍了 ASP.NET 应用程序开发的 3 层结构的概念：

- 用户层：该层负责处理用户的输入和向用户输出。
- 商业逻辑层：它是连接上下两层的纽带，封装了用户对数据库的请求。
- 数据库层：该层负责实际数据的存储。

该例子几乎囊括了 ASP.NET 在 Web 开发应用的各个方面，因此，它需要前面作为基础，只有更好地理解 ASP.NET，才能更为灵活地使用该例子。

观点三：在以后的网站开发中，随着各种技术的不断涌现，只有充分掌握各种技术并综合利用，才能开发出较好的网站，因此本书最后通过 DotNetForum 论坛的分析介绍了如何在一个解决方案里加入更多的项目。这些项目并不仅仅应用于 Web 开发，对于 Windows 的设计也适用。

本书通过大量的代码实例，循序渐进、层次分明地讲述了 ASP.NET 的基本知识点与使用技巧，不仅适合 ASP.NET 初、中级读者，对于有一定 ASP.NET 基础的朋友也有益处。

参加本书编写的还有肖军辉、徐磊、郭春艳、周梅、宋素芳、朱琳、薛洁、田加玉、王磊、段坤军、李洪跃、乔保忠、李付江、陈永锋、裴伟伟、吴云、王星、索睿。

由于编者水平有限，书中不足和错误在所难免，恳求读者批评指正。来信请寄 cbooks@163.com，如果读者使用中有什么不明之处，也可以到 bbs.aspcool.com 论坛的 ASP.NET 板块中提问。

目 录

前言

第 1 章 认识 ASP.NET 1

1.1 IIS 的安装与基本配置 1

1.1.1 IIS 的功能和安装 1

1.1.2 打开 IIS 服务器 1

1.1.3 配置 IIS 服务器 2

1.1.4 虚拟目录的功能与配置 4

1.1.5 配置本书实例 7

1.2 安装配置 Visual Studio.NET 7

1.2.1 Visual Studio.NET 的 安装需求 7

1.2.2 安装配置 Visual Studio .NET 8

1.2.3 检测.NET 安装框架 9

1.2.4 Visual Studio.NET 运行 过程常见问题解析 10

1.3 VS.NET 开发环境介绍 11

1.3.1 认识新建项目对话框 12

1.3.2 打开 Web 项目 12

1.3.3 客户区 13

1.3.4 控制浮动面板 13

1.3.5 解决方案资源管理器 窗口 13

1.3.6 工具箱 15

1.4 安装配置数据库 15

1.4.1 Access 数据库 15

1.4.2 SQL Server 数据库 15

1.4.3 检测数据库是否正确 配置 16

1.5 使用帮助系统 17

第 2 章 C#知识概述 19

2.1 C#语言语法结构 19

2.1.1 字符集、关键字与数据 类型 19

2.1.2 变量定义与变量类型 21

2.1.3 C#运算符与语句 22

2.1.4 基本语句 24

2.2 C#面向对象知识 27

2.2.1 类、对象、继承与派生 27

2.2.2 接口 (Interface) 29

2.2.3 命名空间 (Namespace) 29

2.2.4 修饰符 30

2.2.5 数组 32

第 3 章 ASP.NET Web Forms 35

3.1 .NET 框架运行基本原理 35

3.1.1 HTTP 请求与往返过程 35

3.1.2 静态页面和动态页面 36

3.1.3 ASP.NET 框架运行过程 39

3.2 ASP.NET 具体运行过程 40

3.2.1 建立、运行一个项目 40

3.2.2 ASP.NET Web 请求和 实现 41

3.2.3 进程的处理过程与事件 的触发顺序 43

3.2.4 ASP.NET Web Forms 代码模型 44

3.3 用户界面 (UI) 文件 MyWebForm.aspx 45

3.3.1 用户界面 (UI) 文件 MyWebForm.aspx 概述 45

3.3.2 HTML 代码的使用 45

3.3.3 窗体表单 Form 48

3.4 Web Form 页面处理过程 49

3.4.1 隐藏文件概述	49	4.4.4 使用客户端脚本	80
3.4.2 ASP.NET Web Forms 的事件模型	51	4.4.5 ImageButton 控件	81
3.4.3 页面生存周期	54	4.4.6 LinkButton 控件	83
3.5 ASPX 和 ASPX.CS 的关联	57	4.5 文本的显示	83
3.5.1 布局设计和逻辑设计 的关联	57	4.5.1 静态文本的显示	83
3.5.2 Page 指令	57	4.5.2 HTML 标记的显示	83
3.6 Page 类	59	4.5.3 用 Label 和 Panel 控件 显示动态文本	85
3.6.1 Page 属性列表	59	4.6 TextBox 控件	87
3.6.2 IsPostBack 属性	60	4.6.1 TextBox 控件的功能 与使用	87
3.6.3 Request 属性	61	4.6.2 TextBox 的事件	88
3.6.4 Response 属性	63	4.7 DropDownList 控件	90
3.7 应用程序组织方式	64	4.7.1 DropDownList 控件 的使用	90
3.7.1 文件概述	64	4.7.2 DropDownList 数据绑定 ..	91
3.7.2 解决方案、项目和文件	64	4.7.3 利用 AutoPostBack 关联 选项	95
3.7.3 标记、代码以及资源	65	4.7.4 OnSelectedIndexChanged 事件	97
3.7.4 AssemblyInfo.cs 文件 以及 bin 文件夹	65	4.8 ListBox 控件	97
第 4 章 服务器控件的使用	66	4.8.1 ListBox 控件功能介绍	97
4.1 Visual Studio.NET 控件概述	66	4.8.2 多项的添加和删除	100
4.1.1 HTML 控件与 Web 服 务器控件	66	4.9 CheckBoxList 和 CheckBox 控件	101
4.1.2 服务器控件与 HTML 标记	68	4.9.1 CheckBox 和 CheckBoxList 控件的功能与区别	101
4.1.3 服务器控件的区分与 使用	70	4.9.2 CheckBox 和 CheckBoxList 的使用	103
4.2 服务器控件概述	71	4.10 RadioButtonList 控件和 RadioButton 控件	104
4.2.1 System.Web.UI.HtmlControls	71	4.11 Table 标记和 Table 控件	107
4.2.2 System.Web.UI.WebControls	74	4.11.1 表格的组成	107
4.3 Web 控件的运行原理	76	4.11.2 Table、TableRow 和 TableCell	111
4.4 Button 控件	77	4.11.3 数据库中表的基本 概念	113
4.4.1 按钮控件的功能和类型	77	4.11.4 DataTable、DataRow 和 DataColumn 类	114
4.4.2 Button 控件的使用	77		
4.4.3 鼠标指针移过 Button 的效果	79		

4.11.5 数据筛选与排序	121	5.6.1 客户端验证和服务器 端验证	158
4.11.6 数据访问	122	5.6.2 客户端验证	159
4.11.7 建立父子(主次) 关系表	125	5.7 ValidationSummary 控件的 功能和使用	160
4.12 AdRotator 控件	125	5.8 带有验证码的注册页面	161
4.12.1 AdRotator 控件的功能 ..	125	5.8.1 生成验证码	162
4.12.2 AdRotator 事件	127	5.8.2 注册页面	165
4.13 Calendar 控件	128	第 6 章 用户控件	170
4.13.1 Calendar 控件的功能 和外观介绍	128	6.1 用户控件概述	170
4.13.2 样式对象	128	6.1.1 用户控件的作用	170
4.13.3 Calendar 属性的基本 使用	132	6.1.2 用户控件和 Web 应用 程序的区别和联系	170
4.13.4 Calendar 事件	133	6.1.3 Visual Studio.NET 中用户 控件的添加与引用	170
4.13.5 Calendar 应用	134	6.2 用户的控件属性	171
4.14 图片上传	141	6.2.1 公开用户控件的属性	171
4.14.1 图片上传概述	141	6.2.2 在用户控件中封装事件 ...	175
4.14.2 上传文件到文件夹 并显示	142	6.3 以编程方式创建用户控件	176
4.14.3 上传文件到数据库	144	第 7 章 数据库	179
第 5 章 验证控件	148	7.1 SQL 语句与基本概念	179
5.1 验证控件概述	148	7.1.1 Select 语句	179
5.1.1 验证控件的共性	148	7.1.2 Insert 语句	183
5.1.2 关于脚本库与验证 的触发	149	7.1.3 Update 语句	184
5.2 RequiredFieldValidator 控件 的功能与使用	149	7.1.4 Delete 语句	184
5.3 CompareValidator 比较验证 控件	151	7.1.5 SQL 语句变量	184
5.4 RangeValidator 控件	153	7.1.6 事务、存储过程与 触发器	184
5.5 RegularExpressionValidator 控件	154	7.2 数据绑定	186
5.5.1 正则表达式模式语法	154	7.2.1 绑定属性: <%# Prop %>	186
5.5.2 RegularExpressionValidator 功能介绍	157	7.2.2 绑定到控件属性	186
5.6 CustomValidator 自定义验证 控件	158	7.2.3 DataBinder.Eval()函数	188
		7.3 使用 Connection 对象连接到 数据库	189
		7.3.1 Connection 对象概述	189

7.3.2	OleDbConnection 的使用	190
7.3.3	OLE 连接 ConnectionString 属性介绍	191
7.3.4	查看数据库中的内容	192
7.3.5	SqlConnection 的使用	192
7.3.6	SqlConnection 类	195
7.4	执行数据库命令	196
7.4.1	生成 SQL 语句	196
7.4.2	执行 SQL 命令	197
7.4.3	SqlCommand 类	199
7.4.4	存储过程功能概述	202
7.4.5	建立存储过程	202
7.4.6	使用存储过程	203
7.5	DataReader 对象	206
7.5.1	DataReader 概述	206
7.5.2	SqlDataReader 的使用	206
7.5.3	用 DataReader 返回多个结果集	209
7.6	DataAdapter 和 DataSet	209
7.6.1	DataAdapter 和 DataSet 基本机制	209
7.6.2	DataSet 类和 DataAdapter 类	211
7.6.3	数据访问	215
7.7	DataGrid 控件	218
7.7.1	DataGrid 控件的功能和使用	218
7.7.2	列绑定	220
7.7.3	数据排序	234
7.7.4	数据分页	235
7.8	DataList 控件	241
7.9	Repeater 控件	244
7.9.1	Repeater 控件的功能和特点	244
7.9.2	嵌套 Repeater 控件显示数据	246
7.9.3	Repeater 事件	247

第 8 章 .NET 对 XML 的支持 249

8.1	XML 概述	249
8.1.1	XML 定义及和 HTML 的比较	249
8.1.2	XML 的格式	253
8.2	XML 数据的显示	259
8.2.1	使用 HTML 显示 XML 中的数据	259
8.2.2	使用 CSS 格式化数据	261
8.2.3	使用 XSL 显示数据	262
8.3	XML 的读取和写入之 SAX	267
8.3.1	SAX 的基本概念	267
8.3.2	XmlTextReader 对象	268
8.3.3	XmlTextWriter 对象	273
8.4	XML 文档读取/写入之 DOM	276
8.4.1	DOM 的历史和概述	276
8.4.2	DOM 的处理机制	277
8.5	.NET 对 XML 支持之 DataSet	281

第 9 章 应用程序配置 284

9.1	全局应用文件 Global.asax	284
9.1.1	Global.asax 功能概述	284
9.1.2	阅读和编写 Global.asax	284
9.1.3	Global 事件	285
9.2	bin 文件夹	286
9.2.1	bin 文件夹的功能和作用	286
9.2.2	编译使用 cs 文件	286
9.3	web.config	288
9.3.1	ASP.NET 配置简介	288
9.3.2	配置文件访问原则	288
9.3.3	配置文件的语法规则	290
9.3.4	标准节和自定义节的配置	291
9.3.5	ASP.NET 配置的访问	292
9.3.6	使用位置和路径进行局部配置与锁定配置	293
9.4	Application 对象	294

9.5	Cookie	295	10.5	异常处理	332
9.6	Session 对象	297	10.5.1	异常的概念	332
9.6.1	Session 模型与作用	297	10.5.2	异常的实现	333
9.6.2	Session 意义与基本 使用	298	第 11 章	网络安全	335
9.6.3	Session 失效	299	11.1	网络安全概述	335
9.6.4	Session 的配置	305	11.1.1	安全性概述	335
9.6.5	HttpSessionState 类	307	11.1.2	身份验证与授权的 配置	335
9.7	ViewState	308	11.1.3	身份验证与授权的 基本流程	336
9.7.1	ViewState 的功能和作用	308	11.2	基于 Windows 的身份验证 与文件授权	338
9.7.2	ViewState 的配置及 使用	308	11.2.1	Windows 身份验证	338
第 10 章	页面跟踪、调试与异常处理	313	11.2.2	Windows 权限管理 基本思想	340
10.1	ASP.NET 的调试需求与配置	313	11.2.3	IIdentity 接口和 IPrincipal 接口	341
10.1.1	调试基本要求与 解决方案	313	11.2.4	身份标识与模拟	346
10.1.2	调试错误信息的显示 配置	314	11.3	基于窗体的身份验证	348
10.1.3	按条件显示自定义 错误页面	317	11.3.1	基于窗体的身份验证 概述	348
10.1.4	通知系统管理员	318	11.3.2	基于窗体验证的分析 与配置	352
10.2	ASP.NET 跟踪技术	321	11.3.3	基于窗体验证的授权 配置	354
10.2.1	ASP.NET 跟踪概述	321	11.3.4	身份验证票与操作	355
10.2.2	页级跟踪	321	11.3.5	身份验证总结与 Forms 验证的角色解决方案	358
10.2.3	ASP.NET 跟踪模式的 输出	322	第 12 章	系统缓存与优化	366
10.2.4	使用 IsEnabled 属性 跟踪输出信息	324	12.1	缓存原理	366
10.3	应用程序级跟踪	326	12.2	页面输出缓存	367
10.4	Visual Studio.NET 调试技术	328	12.2.1	页面输出缓存概述	367
10.4.1	ASP.NET 调试概述 与配置	328	12.2.2	HttpCachePolicy 类	370
10.4.2	调试 ASP.NET 应用 程序	328	12.3	页片断缓存	371
10.4.3	使用 Visual Studio.NET 调试 ASP.NET 应用程序	330	12.3.1	缓存参数	371
			12.3.2	缓存控件	372
			12.4	数据缓存	374

12.4.1 数据缓存的使用	374	14.4.3 图书模块	423
12.4.2 Cache 类	376	14.4.4 购物模块	423
12.5 系统优化	382	14.4.5 订单模块	424
12.5.1 系统优化概述	382	14.4.6 服务模块	425
12.5.2 ASP.NET 系统优化提示	383	14.5 用户控件与页面的实现	426
第 13 章 Web Service	385	14.5.1 用户控件的实现	428
13.1 Web Service 的应用	385	14.5.2 页面设计与实现	431
13.1.1 Web Service 介绍	385	14.6 应用程序的配置与发布	449
13.1.2 Web Service 的应用	388	第 15 章 网络论坛	450
13.2 Web 服务的数据类型与内部对象	394	15.1 系统设计	450
13.2.1 Web 服务的数据类型	394	15.1.1 设计思想	450
13.2.2 内部对象	395	15.1.2 数据库设计	450
第 14 章 网络图书购物系统	397	15.1.3 存储过程	456
14.1 整体设计思想与规划	397	15.2 程序预览	460
14.2 数据设计	397	15.3 业务逻辑整体设计	467
14.2.1 数据库设计	397	15.3.1 业务逻辑总体设计思想	467
14.2.2 数据关系图	400	15.3.2 业务逻辑设计实现(下层)	469
14.3 存储过程与数据访问	400	15.3.3 业务逻辑设计实现(中层)	477
14.3.1 存储过程的实现	401	15.3.4 业务逻辑设计实现(上层)	478
14.3.2 数据访问层	403	15.4 应用程序的配置和发布	505
14.4 程序预览	422		
14.4.1 主页模块	422		
14.4.2 注册和登录模块	422		

第 1 章

认识 ASP.NET

1.1 IIS 的安装与基本配置

1.1.1 IIS 的功能和安装

IIS 是 Internet Information Server (网络信息服务) 的简称, 简单地讲就是在 Internet 上发布信息的 Web 服务器。IIS 通过使用超文本传输协议 (HTTP) 传输信息, 并将信息加载到网站上, 当然还可以配置 IIS 提供文件传输协议 (FTP) 服务, 以便于信息的发布和资源的共享。

IIS 是在网站上构建和部署电子商务解决方案以及任务关键的应用程序的一种安全平台。要想让自己的计算机能够作为网络服务器, 首先需要安装 IIS。IIS 的安装步骤如下(注: 这里以 Windows XP 为例, 其他版本的 Windows 与此类似)。

(1) 在“开始”菜单中选择“控制面板”选项。

(2) 双击“添加或删除程序”图标。

(3) 单击对话框左边的“添加/删除 Windows 组件”图标, 在弹出的“Windows 组件向导”对话框中选中“Internet 信息服务 (IIS)”复选框, 如图 1-1 所示。单击右下部的“详细信息”按钮, 可以查看子选项, 确认至少选中了“FrontPage 2000 服务器扩展”、“World Wide Web 服务器”、“公用文件”和“Internet 信息服务单元”复选框。

(4) 单击“下一步”按钮, 系统将复制、安装和配置组件, 直到完成安装。

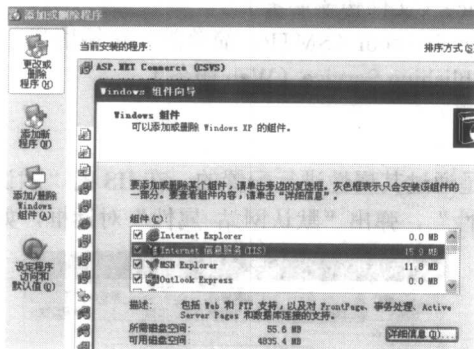


图 1-1 安装 IIS 的 Windows 组件向导

1.1.2 打开 IIS 服务器

安装完了 IIS 服务器以后, 接下来就是要启动 IIS 网络信息服务器。选择“开始→控制面板→管理工具→Internet 服务管理器”, 打开 IIS 服务器。一种更快捷的方法是选择“开始→运行”, 通过输入“inetmgr.exe”命令直接打开 IIS 服务器。

打开 IIS 后将看到如图 1-2 所示的窗口，在左边目录树上依次展开各个节点，就可以看到 IIS 服务器提供的 Web 服务——“默认网站”。图 1-2 所示为笔者机器上默认 Web 站点上的部分内容，实际目录随机器不同略有差异。

当选中“默认网站”时，注意此时工具栏最后 3 个图标工具，它们分别代表启动、停止和暂停项目。需要注意的是，IIS 服务器的工具栏随着选择文件夹的不同而有着不同含义，因此，此处应确认选择的是“默认网站”。

当在上面停止默认网站时，它所提供服务仍然存在。如果安装了 IIS，但是并不想进行网站开发，可以停止该 IIS 服务，以便节省系统资源。

默认情况下，IIS 服务是随系统启动而自动启动的。如果想停止 IIS 服务，可以按照下面的操作步骤设置。

(1) 选择“开始→控制面板”命令，双击“管理工具”图标。

(2) 双击“服务”图标，找到名称为 IIS Admin 的服务选项。

(3) 双击 IIS Admin 服务，弹出“IIS Admin 的属性”对话框。在“常规”选项卡的“服务状态”选项区域中，单击“停止”按钮，这样就停止了 IIS 服务（如图 1-3 所示）。如果在“启动类型”下拉列表框中选择启动类型为“手动”，以后就可以手动启动 IIS 服务。

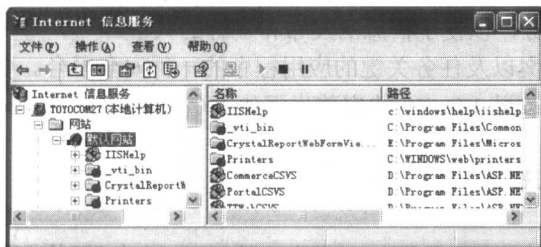


图 1-2 IIS 信息服务管理器

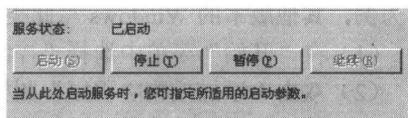


图 1-3 “IIS Admin 的属性”对话框

停止 IIS 服务时，一些依赖于 IIS 服务器的程序将停止运行。“IIS Admin 的属性”对话框中，在“依赖关系”选项卡中可以看到有如下应用程序依赖于该服务（具体内容随安装 IIS 组件的不同而略有不同）：

- FTP Publishing Service（文件传输服务）
- Simple Mail Transport Protocol（SMTP，简单邮件传输协议）
- World Wide Web Publishing Service（Web 服务）

1.1.3 配置 IIS 服务器

IIS Web 站点服务主要是通过其属性进行配置的。在 IIS 的“默认网站”上单击鼠标右键，在弹出的快捷菜单中选择“属性”，弹出“默认网站 属性”对话框，如图 1-4 所示。

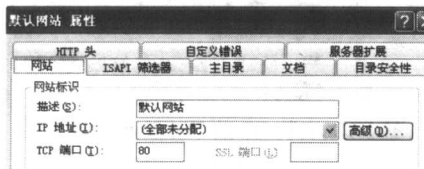


图 1-4 “默认网站 属性”对话框

1. “网站”选项卡

“网站”选项卡如图 1-4 所示。

在“IP 地址”下拉列表框中输入网络的 IP 地址。IP 地址是网络上计算机通信的基础，网络上的计算机通过 IP 地址来寻找另一台计算机。

假设现在在 C:\inetpub\wwwroot\ 目录下有一个 page.htm 页面，而本机的 IP 地址是 192.168.200.27，那么就可以在 IP 地址栏中输入 http://192.168.200.27/page.htm 访问该页面。

通常情况下，使用系统默认的设置“（全部未分配）”。其实，这个全部未分配的 IP 地址为 127.0.0.1，它映射本机 localhost 这个域名。域名可以简单理解为 IP 地址的别名。换句话说，在浏览器中输入 http://localhost 的效果和输入 http://127.0.0.1 的效果相同，只是容易记忆。

在“TCP 端口”文本框中，输入 TCP 使用的端口。端口是进程之间进行通信的基础，它和地址 IP 分别从微观上和宏观上决定了一个应用程序的执行。

可以把 IP 地址比喻为一个公司的总机，把端口比作分机。试想如果要想找公司里的一个人，则不但需要知道总机号，还要知道这个人的分机号，而这个“分机号”就可以理解为端口。

正如公司里不同的人通常有不同的分机号一样，在网络中不同的协议也使用不同的端口进行通信。对于 Windows 操作系统来说，可以在 C:\Windows\system32\drivers\etc 目录下找到 services 这个文件。用“记事本”打开后，可以看到部分内容如下：

```
.....
Qotd          17/tcp      quote          #Quote of the day
Qotd          17/udp      quote          #Quote of the day
Chargen       19/tcp      ttytst source  #Character generator
Chargen       19/udp      ttytst source  #Character generator
ftp-data      20/tcp                        #FTP, data
ftp           21/tcp      ftp            #FTP. control
telnet        23/tcp
smtp          25/tcp      mail           #Simple Mail Transfer Protocol
http          80/tcp      www www-http   #World Wide Web
.....
```

该列表显示了协议和端口的基本关系。例如，ftp 使用 21 端口、telnet 使用 23 端口、smtp 使用 25 端口等，TCP 使用的是 80 端口。这些端口都是不同应用程序直接默认使用的端口，不要更改。

2. “主目录”选项卡

在“主目录”选项卡中主要设置本地文件存放的位置。系统默认存放在 c:\inetpub\wwwroot 位置。可以单击“浏览...”按钮选择其他位置。将在 1.1.4 节详细介绍目录的配置。

3. “文档”选项卡

“文档”选项卡如图 1-5 所示。

“文档”选项卡主要用于设置作为本站主页的默认文档。那么什么叫主页的默认文档呢？

对于网上冲浪者来说，通常在地址栏里输入一个网址，然后浏览器就自动打开一个页面，这个页面就为默认文档。例如，在浏览器中输入 SOHU 主页地址 http://www.sohu.com 时就会出现 SOHU 主页，事实上完整的网址是 http://www.sohu.com/index.html:80，这个 index.html 就定义了它的默认文档（80 表示的是 HTTP 默认使用的端口，可以省略不写）。

启用默认文档可以减少浏览者的网址输入，也便于浏览者记忆网址。用户可以设置自己的默认文档。单击“添加”按钮，输入想作为主页的默认文档的文档名后单击“确定”按钮即可。提倡使

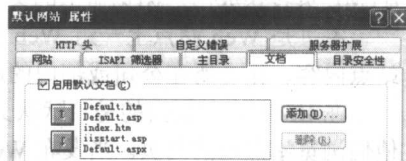


图 1-5 “文档”选项卡

用如下的默认名称: index.htm、index.html、index.asp、index.aspx 或者 default.htm、default.html、default.asp 和 default.aspx。

当在“默认文档”列表中输入多个文档名称时,系统将会依次从上到下在网站文档中寻找最先匹配的页面文件。如果没有,系统将发出错误信息。

当安装 IIS 时,系统自动添加了 iisstart.asp 到“默认文档”列表中。可以单击“添加”按钮,添加多个默认文档。现在假设在主目录下(默认为 c:\inetpub\wwwroot)有一个 main 文件夹,该文件夹下有一个页面文件 text.htm。而在“默认文档”列表中,设置了 3 个默认文档,顺序如下:

- iisstart.asp
- default.htm
- index.htm

那么当输入 http://localhost/main 时, IIS 将先试用 http://localhost/main/iisstart.asp 访问,如果没有,就用 http://localhost/main/default.htm 访问,如果还没有就用 http://localhost/main/index.htm 访问。最后如果还没有就报错,提示所请求的页面不存在。虽然可以通过 http://localhost/main/test.htm 访问 test.htm 页面,但是通过 http://localhost/main 并不能够访问 test.html 页面。

现在已经配置好了 IIS。启动浏览器,在地址栏中输入 http://localhost/, 如果可以看到图 1-6 所示页面,那么配置成功。

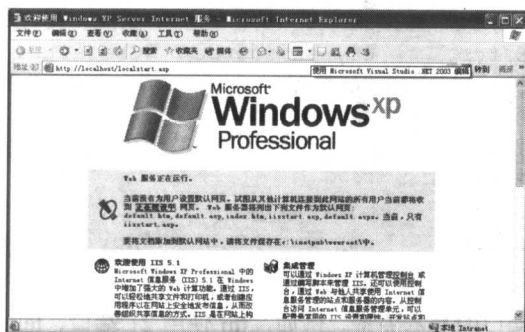


图 1-6 运行页面 http://localhost 页面

1.1.4 虚拟目录的功能与配置

1. 虚拟目录功能

Web 站点是计算机网络上的一个位置,它以网页或文档形式提供信息,访问者只需通过 Web 浏览器连接到站点。计算机网络可以是 Internet,也可以是连接办公室中所有计算机的本地网络 Intranet。信息可以按 HTML 页格式发布,也可以按其他文档格式发布。要查看 Web 站点上的可用信息,访问者需使用 Web 浏览器软件程序,如 Microsoft Internet Explorer,它可将 Web 站点上的 HTML 页转换成用户浏览器上的文本和图形。

主页是 Web 站点上一组 Web 页或其他文件的起始页。它是 HTML 格式的文档,可用来说明站点内容、作者所在公司或部门的新闻、指向其他相关文档的链接或站点作者的个人信息等。

在 IIS 中,主页指的是“默认文档”。主页的首要功能就是向访问者介绍 Web 站点,并帮助他们浏览站点上的网页。当发布 Web 站点时,需要指定包含要发布文档的目录。Web 服务器无法发布未包含在指定目录中的文档。要发布 Web 站点,必须首先确定如何组织发布目录中的文件。本节将详细介绍这方面的问题。首先选择 IIS 的默认站点,右击选择“属性”命令,在弹出的对话框中切换到“主目录”选项卡,如图 1-7 所示。

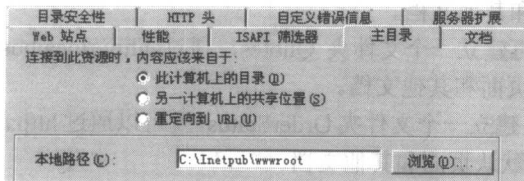


图 1-7 主目录路径

2. 主目录

每个 Web 站点都必须有一个主目录。“主目录”是站点访问者的起始点，也是 Web 发布树的顶端，其中包含主页或索引文件，用来欢迎访问者并包含指向 Web 站点中其他页的链接。主目录映射到站点的域名。例如，如果站点的 Internet 域名是 `www.mywebsite.com`，主目录是 `C:\inetpub\wwwroot`，则 Web 浏览器使用 URL `http://www.mywebsite.com/` 就可以来访问 `C:\inetpub\wwwroot` 目录中的文件。

在 Windows 环境中，默认主目录是 `C:\inetpub\wwwroot\`。正是这个原因，所以如果使用 Visual Studio.NET 建立项目时，默认文档就存放在 `C:\inetpub\wwwroot\` 目录下。

3. 更改主目录

除了使用默认主目录外，还可以更改主目录。在“主目录”选项卡中单击“浏览...”按钮，选择需要的文件夹路径作为主目录。例如，在 `D:\webserver` 目录下存放有 Web 文档，则可以将本地路径设置为 `D:\webserver`，这样就可以通过 `http://localhost/` 访问该文档下的文件。

除此以外，IIS 还支持将网络上另一台计算机上的文件作为主目录以及支持重定向到 URL。

4. 虚拟目录

要从主目录以外的其他目录中进行发布，就必须创建虚拟目录。虚拟目录不包含在主目录中，但在显示给客户浏览器时就像位于主目录中一样。

虚拟目录有一个别名，供 Web 浏览器访问此目录。别名通常要比目录的路径名短，更便于用户输入。使用别名更安全，因为用户不知道文件是否真的存在于服务器上，所以便无法使用这些信息来修改文件。使用别名可以更方便地移动站点中的目录。一旦要更改目录的 URL，只需更改别名与目录实际位置的映射即可。

例如，假设在公司的内部网上为销售部门建立了一个 Web 站点，如图 1-8 所示。表 1-1 所示为文件的实际位置与访问文件的 URL 之间的映射关系。



图 1-8 文件的虚拟位置

表 1-1 文件实际位置、别名与 URL 的关系

实际位置	别名	URL 路径
<code>C:\inetpub\wwwroot</code>	主目录（没有）	<code>http://localhost/Sales</code>
<code>C:\inetpub\wwwroot\Sales\Quotes</code>	无	<code>http://localhost/Sales/Quotes</code>
<code>C:\inetpub\wwwroot\Sales\OrderStatus</code>	无	<code>http://localhost/Sales/OrderStatus</code>
<code>D:\Sales\PR</code>	PR	<code>http://localhost/PR</code>
<code>\\Server2\SalesData\ProdCustomers</code>	Customers	<code>http://localhost/Customers</code>

上面的例子中将 `C:\inetpub\wwwroot` 作为主目录。

假设主目录下有一个文件夹 `Sales`，其中包含 `Sales` 的各种文档，这样用 `http://localhost/Sales`

就能够访问它的默认页面和其他文档。

同样在 Sales 目录下再建立一个文件夹 Quotes, 通过 <http://localhost/Sales/Quotes> 也能够访问 Quotes 文件夹下面的默认页面和其他文档。

如果在 Sales 目录下又建立一个文件夹 OrderStatus, 就可以通过 <http://localhost/Sales/OrderStatus> 访问 OrderStatus 目录下的默认页面和其他文档。

比较上面的网址不难看出, 文件夹的顺序和实际目录下文件夹的顺序是一样的。

现在在 D 盘建立一个文件夹 D:\Sales\PR, 设置它的别名为 PR, 这样就可以通过 <http://localhost/PR> 访问 PR 文件夹下的默认页面和其他文档。此时通过网址并不能判断文件的实际位置, 这就是使用虚拟目录的优点。

利用虚拟路径, 不但可以将路径映射到本地文件, 还可以映射到网络上其他计算机上, 同时在映射时也可以给文件起别名。

假设在一网络中有一名称为 Server 的服务器, 该服务器下有一个文件夹 SalesData, 其下还有一个文件 ProdCustomers。给这个路径起一个别名 Customer, 通过 <http://localhost/Customers> 就能够访问 \\Server2\SalesData\ProdCustomers 目录下的默认页面和文档, 所以利用虚拟路径除屏蔽页面的实际位置、提高了安全性外, 还可以简化用户的网址输入。

5. 识别虚拟目录

打开 IIS 服务器, 可以看到 IIS 左右两边的文件夹有 4 种图标, 如图 1-9 所示。

- : Windows 标准文件夹图标, 其意义是文件夹的实际目录存在于 Web 服务器的主目录中。
- : 表示它们是虚拟目录, 通常包含脚本和 _vti_bin 文件等。
- : Web 应用图标。它们也是虚拟目录, 但可以转换为物理目录 (后面介绍怎样转换)。
- : 表示它们是具体的文件, 只出现在 IIS 服务器右边文件中。

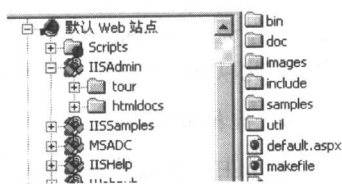


图 1-9 IIS 中文件夹图标

6. 创建虚拟目录

有两种方法可以创建虚拟目录。

方法一:

(1) 在站点上单击鼠标右键, 选择“新建→虚拟目录”命令, 弹出新建“虚拟路径创建向导”对话框, 并单击“下一步”按钮。

(2) 设置虚拟路径的别名, 然后单击“下一步”按钮。

(3) 在“目录”文本框中输入文件的实际位置, 也可以单击“浏览...”按钮进行文件的选择, 单击“下一步”按钮。

(4) 使用默认读写设置, 完成创建。

方法二:

在“默认网站属性”对话框中, 切换到“主目录”选项卡, 在其下半部可以设置整个站点的虚拟路径, 如图 1-10 所示。单击“创建”按钮, 为应用程序配置虚拟路径。

在创建虚拟目录的同时, “创建”按钮变为“删

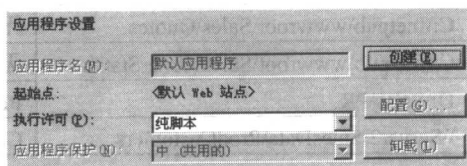


图 1-10 文件属性

除”按钮，单击“删除”按钮，可以删除刚才创建的虚拟目录。

除了给整个应用程序配置虚拟路径外，还可以为 Web 站点列表中的每一个文件单独配置虚拟路径。在默认网站的站点列表中，选择具体的一个文件，单击鼠标右键，选择“属性”命令，在弹出的对话框中的“主目录”选项卡中也有类似图 1-10 所示的创建虚拟目录选项。单击“创建”按钮，将创建别名是应用程序实际名的虚拟路径。

1.1.5 配置本书实例

要运行本书实例，需要配置虚拟目录，这里假设将配套光盘的文件复制到 E:\目录下。以 OnNetBookSell 配置为例，按照上面的讲述，使用向导创建虚拟目录时，使用 OnNetBookSell 作为别名，使用 E:\OnNetBookSell 作为实际文档位置，完成后即可使用 <http://localhost/OnNetBookSell> 进行访问。

提示：如果将源代码复制到 E:\Code 文件夹下，将主目录设置为 E:\Code，可以使用方法二快速配置虚拟路径，在配置时不要使用中文字符。本书配套光盘中的“配置虚拟目录.doc”介绍了怎样配置本书实例。

1.2 安装配置 Visual Studio.NET

要想运行 ASP.NET 页面，必须先安装 .NET Framework 框架，微软目前也在其官方网站 <http://www.asp.net> 提供了免费的最新 .NET 框架，读者可以到那里下载安装。

本书中将使用 Visual Studio.NET 2003 开发 ASP.NET 页面，由于 Visual Studio.NET 自身已经配带了 .NET Framework，因此在安装 Visual Studio.NET 的同时，也安装了 .NET Framework 框架，但是需要读者明白的是，ASP.NET 页面的运行需要的是 .NET Framework 框架，并不一定需要 Visual Studio.NET 的支持。

虽然理论上可以用任何字处理软件编写 ASP.NET 应用程序（保存时以 .aspx 为后缀），但是面对越来越复杂和庞大的应用系统，这实际上是极不可取的做法。而 Visual Studio.NET 提供了良好的开发环境，使用户可以专注于实际的业务逻辑。如果把 ASP.NET 应用程序制作比作画画，那么 Visual Studio.NET 就像一支笔，不同的画家将使用这支“笔”根据不同的业务需求画出不同的画。而本书的目的就是教会读者怎样使用这支“笔”。

1.2.1 Visual Studio.NET 的安装需求

相对于一般的应用软件，Visual Studio.NET 对硬件和软件的安装要求较高，安装时间也较长。但是整个过程没有什么特别配置，利用安装向导，可以很容易地完成安装。下面列出了系统必备的硬件和软件要求。

1. 硬件要求

- (1) CPU: 450 MHz Pentium II 处理器，推荐 600 MHz Pentium III 以上的处理器。
- (2) 内存: Visual Studio.NET 2003 的最低 RAM 要求如下。

- Microsoft Windows Server 2003——160MB
- Windows XP Professional——160MB
- Windows XP Home Edition——96MB
- Windows 2000 Professional——96MB
- Windows 2000 Server——192MB

推荐内存存在 256MB 以上，否则运行速度可能很慢。