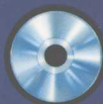


光盘内容
书中所有范例源代码



指南

王有礼 编著

ASP.NET 2.0

完全开发指南——基于VB

- | | | |
|--|---|--|
| <ul style="list-style-type: none"> ● VB.NET简介 ● 标准控件 ● 验证控件和数据的验证 ● SQL Server 2005及SQL语句 ● SqlDataSource、GridView和AccessDataSource ● 数据控件 | <ul style="list-style-type: none"> ● 数据库高级技术 ● 文件和常用对象的使用方法 ● HTML控件 ● ASP.NET对XML的支持 ● 标准控件高级技术 ● 数据控件高级技术 ● 母版页和导航 | <ul style="list-style-type: none"> ● 用户控件和自定义控件 ● 组件和Web Services, Global.asax和Web.config文件 ● 网页安全和Login控件 ● 层叠样式表 ● 主题、个性化配置和Web部件 |
|--|---|--|

服务支持网站

<http://graph.cs.ntust.edu.tw/asp.net/>

TP393.092/944D

2008

ASP.NET 2.0 完全开发指南

——基于 VB

王有礼 编著

科学出版社

北京科海电子出版社

内 容 提 要

本书引自台湾畅销 ASP.NET 图书,作者基于多年的科研和教学经验,依据初学者的实际学习需求编写而成,全书辅以大量范例程序,对 ASP.NET 2.0 中的各种技术作了具体而细致的讲解。

作者以 VB 为开发语言,分 21 章介绍了 ASP.NET 2.0 Web 应用开发的技术和方法过程,内容包括:VB.NET 简介,标准控件,验证控件和数据的验证,SQL Server 2005,SQL 语句和表的规范化,SqlDataSource、GridView 和 AccessDataSource,数据控件,数据库高级技术,文件和常用对象的使用方法,HTML 控件,XML、XML Schema 和 XSL,ASP.NET 对 XML 的支持,标准控件高级技术,数据控件高级技术,母版页和导航,用户控件和自定义控件,组件和 Web Services 简介,Global.asax 和 Web.config 文件,网页安全和 Login 控件,层叠样式表,主题、个性化配置和 Web 部件等。配书光盘提供了书中所有范例的源代码。

全书范例极为丰富,讲解细致,并配有大量习题。同时,作者从相关 ASP.NET 论坛收集了众多读者在实际开发中遇到的问题,并予以详细解答。所以,本书既适合 ASP.NET 2.0 的初学者,对已有一定经验的开发人员也有很大的技术参考价值。

图书在版编目(CIP)数据

ASP.NET 2.0 完全开发指南——基于 VB /王有礼编著.

—北京:科学出版社,2007

ISBN 978-7-03-020310-6

I. A… II. 王… III. ①主页制作—程序设计②BASIC 语言—程序设计 IV. TP393.092 TP312

中国版本图书馆 CIP 数据核字(2007)第 150825 号

责任编辑:何 武 王程程 / 责任校对:科 海

责任印刷:科 海 / 封面设计:林 陶

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京市鑫山源印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2008 年 1 月第一版

开本:16 开

2008 年 1 月第一次印刷

印张:47

印数:1-4000

字数:1144 千字

定价:79.00 元(含 1CD 价格)

(如有印装质量问题,我社负责调换)

前言

或许用下面这首诗来形容 ASP.NET 最是恰当不过:

横看成岭侧成峰,
远近高低各不同。
不识庐山真面目,
只缘身在此山中。

宋·苏轼——《题西林壁》

ASP.NET 涵盖的知识范围相当浩瀚, 穷我们一生之力都有可能看不完。但是, 只要从其中学到了三招两式, 也就能设计出不错的网页系统。通常, 学习 ASP.NET 或用 ASP.NET 来开发系统时, 都要借助 MSDN。有人说: “学习 ASP.NET 是不需要看书的, 只要有 MSDN, 再加上 Google 就可以了”。此话诚然有其道理, 只是 MSDN 就像大海一样, 如果只借助 Google, 还是只能在大海之中摸索, 很快就会耗尽体力, 迷失方向。其实, 大多数的作者开始着手写书时, 主要借助的工具就是 MSDN 和 Google。虽然有了这两种工具, 就可以找到许许多多的资料, 但是有些资料像断简残篇一般, 学完之后脑海中仍然是雾里看花, 不知其所以然。一本好的 ASP.NET 书籍应该像指南针一样, 能够帮助读者很快地找到大多数问题的解决方案, 不用再花大量的摸索时间。所以, 要学好 ASP.NET, 除了要借助 MSDN 和 Google 之外, 还要有几本好的 ASP.NET 参考书, 我想这也是大多数作者在写 ASP.NET 相关书籍时的一种动力。

写前两本 ASP.NET 的书籍时, 是在闹市中的校园。推出后受到读者的喜爱, 不及半载, 就已销售一空。这次新版的写作, 转到了位于小城镇边的大学, 此处人烟稀少、青山绿水、鸟语花香、风景如画, 称得上是世外桃源。除了有置身画中写书的感觉, 我还从写书之中获得了三种乐趣, 愿与读者共享。专心写书时, 时间是静止的, 无恩恩怨怨, 也无人间冷暖, 应该很像老僧入定的感觉, 此乃宁静之乐也。写书时, 也会碰到重重的困难, 每一章都会遇到一些问题, 使人迟迟不能动笔, 常会有跳过这些问题的逃避想法。或许是多年来做研究的训练, 遇到问题不会轻易放弃, 所以差不多所有的问题最终都能迎刃而解, 此乃毅力之乐也。前两本关于 ASP.NET 的作品推出时, 获得许多读者的好评, 这是所有作者都很在意的成就感。只是写书时并不能感受成就感的乐趣, 而要等到书籍推出一年半载之后。此乃成就之乐也, 希望半年后我仍能享受此乐趣。

本书分成 21 章, 包含了 ASP.NET 2.0 中的大部分技术。在 ASP.NET 2.0 中包含了 8 种控件: 标准控件、验证控件、数据控件、HTML 控件、导航控件、Login 控件、Web 部件控件和 Crystal Reports 控件, 前 7 种控件在本书中都做了详细介绍。除了这些控件技术之外, 本书还介绍了 VB 语言、数据库技术、文件及常用对象的用法、XML 技术、组件技术、Web Services 技术、网页程序的信息安全、层叠样式表、个性化配置技术等, 相信能使读者对 ASP.NET 有较完整的了解。另外, 除了各种控件的介绍, 笔者还从知名的 ASP.NET

论坛中收集了许多读者在开发实践中遇到的问题，并将这些问题在书中逐一解答，相信能使读者减少许多开发时间。

本书未包含的部分主要是 Crystal Reports 控件。Crystal Reports 控件，以笔者的经验，宜以专书介绍较妥。由于本书篇幅已经相当大，应对某些技术略作取舍，因此并未对此技术加以介绍。

另外，上学期，我开了“.NET 程序设计”这门课，采用的教科书当然就是拙作《ASP.NET 2.0 完全开发指南——基于 VB》，才发现 ASP.NET 2.0 的内容实在太多，一学期下来，还教不到 10 章。所以建议以拙作当教科书的老师，可以教第 1 章到第 7 章、第 9 章、第 15 章和第 19 章。其余的章节就留给学生自修，或将这门课改成一年的课程，剩下的章节就可以在下一学期教授。

读者阅读本书时若有不解之处，欢迎来函告之，我很乐意为各位解答问题。我的 E-mail 是 yuelwang@ncnu.edu.tw，另外，也可以在我的网站（我学生为我设计的网站）留言，网址是 <http://graph.cs.ntust.edu.tw/asp.net/>。

王有礼

2007 年 10 月

目 录

第 1 章 简介.....	1	3.3 Label、HyperLink、Image 和 ImageMap.....	64
1.1 网页设计回顾.....	2	3.4 TextBox.....	67
1.2 代码内联和代码隐藏.....	13	3.5 CheckBox 和 CheckBoxList.....	68
1.3 章节架构.....	17	3.6 RadioButton 和 RadioButtonList.....	71
习题.....	18	3.7 ListBox 和 DropDownList.....	73
第 2 章 VB.NET 简介.....	19	3.8 Table、TableRow 和 TableCell.....	75
2.1 变量.....	20	3.9 Literal、BulletedList 和 HiddenField.....	77
2.2 表达式的运算符.....	22	3.10 Calendar.....	80
2.3 语句.....	24	3.10.1 DayRender 事件.....	83
2.3.1 赋值语句.....	25	3.10.2 SelectionChanged 事件.....	86
2.3.2 If 语句.....	25	3.10.3 VisibleMonthChanged 事件.....	87
2.3.3 Select 语句.....	26	3.10.4 日历文字的变化.....	87
2.3.4 Do 语句.....	27	3.11 AdRotator.....	90
2.3.5 For 语句.....	28	3.12 FileUpload.....	93
2.3.6 While 语句.....	29	3.13 Wizard.....	96
2.3.7 try 语句.....	29	3.14 Panel 和 Placeholder.....	102
2.3.8 其他的语句.....	31	3.15 MultiView 和 View.....	105
2.4 子过程及函数.....	31	3.16 Substitution 控件和 OutputCache 指示符.....	109
2.4.1 子过程.....	31	3.17 Localize 和 Xml.....	113
2.4.2 函数.....	33	习题.....	113
2.5 内置函数.....	34	第 4 章 验证控件和数据的验证.....	115
2.5.1 Math 对象中的数学函数.....	34	4.1 RequiredFieldValidator.....	116
2.5.2 Strings 对象中的字符串函数 及 String 对象.....	35	4.2 CompareValidator.....	117
2.5.3 日期函数.....	41	4.3 RangeValidator.....	118
2.5.4 数据类型转换函数.....	51	4.4 RegularExpressionValidator.....	119
习题.....	52	4.5 CustomValidator.....	123
第 3 章 标准控件.....	53	4.6 ValidationSummary.....	124
3.1 Button、LinkButton 和 ImageButton.....	54	4.7 ValidationGroup 属性.....	127
3.2 命名空间和 Import 指示符.....	62		

4.8 System.Text.RegularExpressions 的 正则表达式	128
4.8.1 Regex 对象、Match 对象和 MatchCollection 对象	128
4.8.2 Group 对象和 GroupCollection 对象	136
4.8.3 Capture 对象和 CaptureCollection 对象	138
4.8.4 System.Text.RegularExpressions 命名空间中的元字符	140
习题	143

第 5 章 SQL Server 2005、SQL 语句 和表的规范化 144

5.1 创建数据库和表	144
5.1.1 SQL Server 2005 数据库和表的 创建方法	144
5.1.2 SQL Server 2000 数据库和表的 创建方法	149
5.1.3 Microsoft Access 数据库和表的 创建方法	152
5.1.4 在 SQL Server 中用程序来创建 数据库和表	155
5.1.5 在 Microsoft Access 中创建表	164
5.2 新建数据	166
5.3 查询数据	170
5.3.1 简单的数据查询	171
5.3.2 Where 条件式	173
5.3.3 连接表的查询	177
5.3.4 查询统计数据	180
5.4 修改数据	181
5.5 删除数据	182
5.6 表的规范化	183
5.6.1 第一范式	183
5.6.2 表的分割	186
5.6.3 第二范式	188
5.6.4 第三范式	191
5.6.5 Boyce/Codd 范式	193

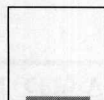
5.6.6 第四范式	195
5.6.7 第五范式	197
习题	201

第 6 章 SqlDataSource、GridView 和 AccessDataSource 203

6.1 GridView 的 BoundField 及 CheckBoxField	205
6.2 GridView 字段的格式	208
6.3 GridView 的排序	210
6.4 GridView 的分页	211
6.5 GridView 的编辑及 CommandField ..	217
6.5.1 默认的编辑模式	217
6.5.2 用 CommandField 来设计 编辑字段	219
6.5.3 调整编辑框的大小	220
6.5.4 确认消息框	222
6.6 参数查询	224
6.7 插入数据	226
6.8 GridView 的 HyperLinkField	228
6.9 GridView 的 ButtonField	231
6.10 GridView 的 TemplateField	233
6.11 数据绑定表达式	237
6.11.1 一般表达式	238
6.11.2 再谈 GridView 字段的格式	240
6.12 AccessDataSource	243
习题	244

第 7 章 数据控件 245

7.1 DetailsView 控件	245
7.1.1 DetailsView 显示数据的方式	245
7.1.2 DetailsView 和 GridView 的 结合使用	246
7.1.3 自定义 DetailsView 的格式	248
7.1.4 DetailsView 字段的格式	249
7.1.5 DetailsView 数据的修改和 删除	251
7.1.6 DetailsView 的新建数据	253



7.1.7 DetailsView 的 HyperLinkField ..255	
7.1.8 DetailsView 的 CommandField ..257	
7.1.9 DetailsView 的 ButtonField ..259	
7.1.10 DetailsView 编辑框大小的 控制 ..264	
7.1.11 DetailsView 删除数据时的 提示消息框 ..271	
7.2 Repeater 控件 ..275	
7.3 DataList 控件 ..278	
7.3.1 DataList 的重复显示功能 ..278	
7.3.2 SelectedItemTemplate 的变换 显示功能 ..281	
7.3.3 EditItemTemplate 的编辑功能 ..283	
7.4 FormView Control ..288	
7.4.1 PagerSettings ..289	
7.4.2 PagerTemplate ..290	
7.4.3 FormView 控件的编辑 ..293	
7.4.4 FormView 控件的新建功能 ..296	
习题 ..302	

第 8 章 数据库高级技术 ..303

8.1 DataSet 中数据表的新建、修改和 删除 ..303	
8.2 节省内存的 SqlDataReader 对象 ..311	
8.3 数据库的存储过程、参数查询及 事务 ..313	
8.3.1 存储过程 ..313	
8.3.2 参数查询 ..318	
8.3.3 事务 ..320	
8.3.4 事务的存储过程 ..325	
8.4 不同数据库的集成 ..329	
8.5 在运行时创建 SQL 语句 ..332	
习题 ..335	

第 9 章 文件和常用对象的使用方法 ..337

9.1 有关文件的对象 ..337	
9.1.1 文件和目录的新建和删除 ..337	
9.1.2 文件的打开 ..342	

9.1.3 将数据写入文件 ..345	
9.1.4 将数据读出文件 ..346	
9.1.5 文件和目录的基本属性信息 ..349	
9.2 Email 的 Smtplib 对象和 MailMessage 对象 ..353	
9.3 Response 对象 ..355	
9.3.1 BufferOutput、Expires、 ExpiresAbsolute 等属性 ..357	
9.3.2 流的属性和方法 ..357	
9.3.3 状态码的种类 ..358	
9.3.4 ClearContent、Flush、End 等 方法 ..358	
9.3.5 Redirect 和 Write 方法 ..359	
9.4 Request 对象 ..360	
9.4.1 浏览器属性 ..361	
9.4.2 ClientCertificate 属性 ..362	
9.4.3 TotalBytes 属性 ..362	
9.4.4 Headers 属性 ..363	
9.4.5 ServerVariables 属性 ..363	
9.4.6 Form 和 Querystring 方法 ..365	
9.5 Page 对象 ..366	
9.5.1 Page 对象中的事件 ..366	
9.5.2 Page 对象中常用的属性和 方法 ..368	
9.6 Application 对象 ..369	
9.7 Session 对象和 Cookie 对象 ..370	
9.7.1 Session 对象 ..371	
9.7.2 Cookie 对象 ..374	
9.8 StateBag 对象的视图状态变量 ..376	
9.9 Server 对象 ..378	
9.9.1 HtmlDecode 和 HtmlEncode 方法 ..378	
9.9.2 MapPath 方法 ..379	
习题 ..379	

第 10 章 HTML 控件 ..381

10.1 HtmlButton、HtmlInputButton 和 HtmlInputImage ..381	
---	--

10.2 HTML 控件共有的属性	385
10.3 HtmlForm 和 HtmlInputText	386
10.4 HtmlAnchor 和 HtmlImage	388
10.5 HtmlInputCheckBox 和 HtmlInputRadioButton	389
10.6 HtmlInputHidden	391
10.7 HtmlSelect	391
10.8 HtmlTextArea	393
10.9 HtmlTable、HtmlTableRow 和 HtmlTableCell	394
10.10 HtmlInputFile	395
10.11 HtmlHead、HtmlLink 和 HtmlTitle	398
10.12 HtmlInputPassword、HtmlSubmit 和 HtmlReset	401
习题	402

第 11 章 XML、XML Schema 和 XSL 403

11.1 XML Schema	406
11.2 XML 文件的验证	413
11.3 XSL 简介	416
11.3.1 XSL 程序和 XML 文件的关系： xml-stylesheet 和 xsl:stylesheet	420
11.3.2 获得标签值的语句： xsl:value-of	421
11.3.3 循环控制语句 xsl:for-each	422
11.3.4 数据排序语句 xsl:sort	423
11.3.5 条件语句：xsl:if	424
11.3.6 条件语句：xsl:choose、 xsl:when、xsl:otherwise	425
11.3.7 函数的调用和定义： xsl:apply-templates 和 xsl:template	427
11.3.8 将元素转换成属性： xsl:attribute	428

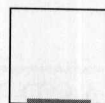
11.4 使用 ASP.NET 程序操作 XSL 程序	429
习题	432

第 12 章 ASP.NET 对 XML 的支持 433

12.1 读取 XML 文件	433
12.2 创建新的 XML 文件	436
12.3 修改 XML 文件	439
12.3.1 修改内容	441
12.3.2 增加新的内容	443
12.3.3 删除某些内容	445
12.4 XML 文件和 DataSet	446
习题	448

第 13 章 标准控件高级技术 449

13.1 文件的编码	449
13.1.1 读取 GB2312 文件的注意 事项	449
13.1.2 文件的默认编码	451
13.1.3 编码代号和字集名称	451
13.2 Label 和 TextBox 中的文字	452
13.2.1 Label 控件中的文字	452
13.2.2 文本框控件中的文字	453
13.3 动态创建 DropDownList 和 ListBox	455
13.3.1 用数组创建 DropDownList	455
13.3.2 用 DataTable 生成 DropDownList	456
13.3.3 用数据库中的 DataTable 创建 DropDownList	458
13.3.4 选项中的“其他”	460
13.3.5 联动式 DropDownList	462
13.4 动态创建 RadioButtonList、 CheckBoxList 和 BulletedList	464
13.4.1 动态创建 RadioButtonList	464
13.4.2 动态创建 BulletedList	465
13.5 动态创建其他控件	466
13.5.1 动态创建多个控件	466



13.5.2 在控件数组中再添加控件.....	467	15.4.2 TreeView 控件的属性	518
13.5.3 动态控件中属性和事件的 设置.....	468	15.4.3 TreeView 控件的 DataBindings 属性	521
13.6 动态设置和清除控件的属性	469	15.4.4 运行时生成 TreeNodes	522
13.6.1 动态设置控件的属性.....	469	15.5 SiteMapPath 控件.....	525
13.6.2 动态清除控件的内容.....	470	15.5.1 编写扩展名为.sitemap 的 XML 文件	525
13.6.3 枚举网页中的控件	472	15.5.2 明确站点地图的提供程序.....	526
13.7 程序的调试和跟踪特性	475	15.5.3 创建含有 SiteMapPath 控件的 网页程序	527
13.7.1 跟踪特性	476	15.5.4 TreeView 控件的 SiteMapPath	529
13.7.2 页面级别的跟踪	477	15.5.5 SiteMapPath 控件中常用的 属性	530
13.7.3 组件的跟踪	479	习题.....	532
13.7.4 应用程序级别的跟踪.....	479		
习题.....	482		
第 14 章 数据控件高级技术	483	第 16 章 用户控件和自定义控件.....	533
14.1 显示确认消息框	483	16.1 用户控件	533
14.2 表字段值 NULL 的处理.....	488	16.1.1 用户控件的创建.....	533
14.3 GridView 中流水编号字段.....	489	16.1.2 用户控件的使用.....	535
14.4 GridView 中行号的格式.....	491	16.1.3 用户控件在模块化中的应用... ..	537
14.5 GridView 和鼠标的交互.....	491	16.2 创建和使用自定义控件.....	540
14.5.1 Attributes 属性	493	16.2.1 创建自定义控件的语法.....	541
14.5.2 Ambient 事件	494	16.2.2 编译用户自定义控件标签 程序	544
14.5.3 Style 属性	496	16.2.3 使用自定义控件.....	546
14.6 GridView 控件加上滚动条.....	499	16.3 属性的设置和 Render 子过程	547
习题.....	500	16.3.1 自定义控件的默认值.....	547
第 15 章 母版页和导航	501	16.3.2 自定义控件中属性值的保存... ..	549
15.1 简单母版页的创建方法	501	16.3.3 枚举类型的属性.....	550
15.1.1 母版页的创建方法	501	16.3.4 层叠式属性.....	552
15.1.2 内容页的创建方法	502	16.3.5 子过程 Render 的写法	555
15.2 嵌套母版页的创建方法	503	16.4 自定义控件的方法	562
15.3 Menu 控件.....	505	16.4.1 简单的方法.....	562
15.3.1 MenuItem 标签.....	507	16.4.2 重载方法	564
15.3.2 菜单控件的属性	509	16.4.3 有返回值的方法.....	566
15.3.3 菜单控件的 DataBindings 属性	513	16.5 自定义控件的事件	568
15.4 TreeView 控件.....	514	16.5.1 不因数据改变而触发的事件... ..	568
15.4.1 TreeNode 标签.....	516		

16.5.2 因数据改变而触发的事件	570
16.6 增加现有控件的功能	573
16.6.1 Web 控件的继承	573
16.6.2 由子控件组成的控件	577
16.6.3 由子控件组成的控件的事件 设置	579
习题	582

第 17 章 组件和 Web Services

简介	583
17.1 组件的创建	583
17.2 组件的编译	584
17.3 组件的引用	585
17.4 一个完整的例子	586
17.5 Web Services 的作用	590
17.6 创建 Web Service 文件	591
17.7 用 HTTP 获取服务	592
17.8 WebMethod 标签	594
17.9 将现有的组件改成 Web Service	595
17.10 在网页应用程序中使用 Web Service	598
习题	606

第 18 章 系统环境: Global.asax 和 Web.config 文件

18.1 Global.asax 文件	607
18.2 Web.config 文件	614
18.2.1 <appSettings>	616
18.2.2 <connectionStrings>	618
18.2.3 <browserCaps>	620
18.2.4 <compilation>	620
18.2.5 <customErrors>	622
18.2.6 <globalization>	622
18.2.7 <httpHandlers>	623
18.2.8 <httpModules>	625
18.2.9 <pages>	625
18.2.10 <processModel>	626
18.2.11 <sessionState>	627

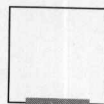
习题	628
----------	-----

第 19 章 网页安全和 Login 控件

19.1 Windows 身份验证	630
19.1.1 Windows 操作系统中的用户 ...	630
19.1.2 Basic Windows 身份验证	632
19.1.3 Digest 和 NTLM Windows 身份验证	634
19.2 Forms 身份验证使用 Login 控件 ...	635
19.2.1 CreateUserWizard 控件	635
19.2.2 将数据存储在自行设计的 数据库中	648
19.2.3 Login 控件和 PasswordRecovery 控件	651
19.2.4 使用 Login 控件的 Forms 身份 验证	656
19.2.5 LoginName 和 LoginStatus 控件	658
19.2.6 ChangePassword 控件	659
19.2.7 LoginView 控件	660
19.2.8 密码格式的设置	663
19.3 Forms 身份验证使用 Web.config 文件中的验证	665
19.3.1 修改“默认的网站”中 “根文件夹”的 Web.config ...	665
19.3.2 创建验证的网页程序	669
19.3.3 Web.config 文件中用户名的 管理	670
19.4 授权	676
19.4.1 File 授权	676
19.4.2 URL 授权	678
19.4.3 角色管理	680
19.5 验证后权限	685
习题	686

第 20 章 层叠样式表

20.1 Style 标签的语法	687
20.2 字体	689



20.3	文本	692	21.2.1	匿名用户的个性化设置	713
20.4	背景	694	21.2.2	程序中有关个性化设置的 写法	715
20.5	页面上项目的位置	698	21.2.3	成员资格和个性化	717
20.6	项目及列表式文字的显示	702	21.3	Web 部件	722
20.7	边框、外补白、内填充和滚动条	706	21.3.1	WebPartManager、WebPartZone 和 CatalogZone 控件	723
20.8	分页打印和鼠标指针外观	707	21.3.2	DeclarativeCatalogPart、 ImportCatalogPart 和 PageCatalogPart 控件	727
20.9	其他	709	21.3.3	EditorZone 及其子控件	728
第 21 章 主题、个性化配置和 Web 部件		711	21.3.4	Web Parts 控件的参数	734
21.1	主题和皮肤	711			
21.2	个性化设置	713			

简介

1

在网页语言的发展史中,最早使用的网页语言是 HTML (Hypertext Markup Language)。HTML 只可以静态地显示各种页面,缺乏与用户交互的界面。针对这种缺点,很快就有动态服务器语言 (ASP, Active Server Page) 的出现。ASP 拥有和用户交互的界面,并且结合数据库的技术,大大增强了网页语言的功能,很快地吸引了网页设计者的注意,并且在很短的时间内,像秋风扫落叶般席卷了网页设计的市场,取代了 HTML。当 ASP 被广泛使用后,出现了一些问题。例如:ASP 的程序代码块常和页面块混杂在一起,使得程序不易维护;又如一个网页在接收数据后,要传至另一个网页中处理,使得页面转换频繁;还有程序设计人员必须自行检验数据的正确性,等等。针对这些缺点,Microsoft 开发出新一代的 ASP 语言,即 ASP.NET,改良并增进了 ASP 的各种功能,使网页语言更臻完美。ASP 在风光了 4、5 年后,又将走入历史,令人不得不感叹信息业变动的迅速,这或许也是信息领域一日千里的写照。对学信息的人来说,一则以忧,一则以喜。忧的是,学一种技术用不了几年。喜的是,只要跟上信息变化的脚步,则永远走在时代的前端。ASP.NET 自从在 2002 年推出,版本的演进从最早的 ASP.NET 1.0,经过 2003 年的 ASP.NET 1.1,到目前的 ASP.NET 2.0,短短的三年之间,就换了三个版本,变换也算频繁。不过,目标始终都是一致的,就是要让 ASP.NET 更容易使用(少写一些代码)及更有效率(执行速度变快)。但是要让一个系统广为流传,还是应该尽早稳定下来,以免造成维护上的困扰。

ASP.NET 结合的技术非常多元,包含标识语言、一般的语言、数据库技术、面向对象技术等等,将传统的程序设计带入了一个崭新的境界。以往要编写很多代码才能设计出来的系统,现在用 ASP.NET 来设计,既快速又方便。也由于结合了许多技术,要学好 ASP.NET,比起以前只要学好一种程序语言,要花上更多的功夫。学习 ASP.NET 要以循序渐进的方法进行,读的时候必须将本书中的例子输入到系统中,一边运行,一边看程序的说明。或许读第一遍时,还很难完全了解,多看几遍,自然而然就会融会贯通。本书所附带的光盘中包含了书中所有的程序,读者可将所需的程序加载到 .NET 工作平台中。

1.1 网页设计回顾

下面我们用一些简单的例子来说明网页语言的演化。首先看程序清单 1-1 的网页程序 WebPage0101.html。

程序清单 1-1 WebPage0101.html

```
1 <html>
2   <body>
3     <center>
4       <font face="宋体">Hi, 你好</font>
5     </center>
6   </body>
7 </html>
```

由程序清单 1-1 可以看出 HTML 的语法是对称性的，如第 1 行及第 7 行的<html>和</html>表示网页语言的开始及结束，其中<html>和</html>称为标签，其他如<body>，</body>，<center>，</center>等等亦称为标签。第 2 行及第 6 行的<body>和</body>表示网页的主体。第 3 行及第 5 行的<center>和</center>表示夹于其间的文件要显示在网页的中央。第 4 行表示“Hi, 你好”要以字体“宋体”显示。运行 WebPage0101.html 后，显示的页面如图 1-1 所示。

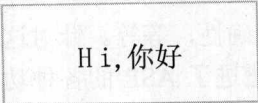


图 1-1 运行 WebPage0101.html

如前所述，HTML 语言只提供显示数据的方法，缺乏和用户交互的功能。为了增加交互的功能，ASP 应运而生，即在 HTML 语言中再添加几个有交互功能的标签。我们用程序清单 1-2 的 WebPage0102.html 和程序清单 1-3 的 WebPage0103.aspx 来说明这种有交互功能的标签。

程序清单 1-2 WebPage0102.html

```
1 <html>
2   <body>
3     <form action="WebPage0103.aspx">
4       请输入姓名:<input type="text" size="8" name="inName"><br />
5       <input type="submit" value="发送">
6     </form>
7   </body>
8 </html>
```

程序清单 1-2 的第 3 行到第 6 行中包含了一些有交互功能的标签，这几个标签是早期的 HTML 语言中所没有的。程序清单 1-2 中第 4 行的<input type="text" size="8"

name="inName">是一种控件标签。此控件标签的意思是定义一个数据输入字段，其变量名称为 inName。控件标签是专门为动态网页设计的，其主要的目的在于接收输入数据。为区别起见，静态网页中使用的标签就称为 HTML 标签。表 1-1 列出常用的 HTML 标签，表 1-2 列出 ASP 常用的控件标签。再看程序清单 1-2 中第 5 行的控件标签，此控件标签在定义一个提交按钮，当此提交按钮被点击后，此网页的输入数据 inName 会被传送到网页程序 WebPage0103.aspx 中处理。WebPage0102.html 怎么知道要将输入的数据传送到 WebPage0103.aspx 中处理呢？那是因为程序清单 1-2 中第 3 行的控件标签 <form action="WebPage0103.aspx">，此控件标签的意思就是，当收到数据后要改至 WebPage0103.aspx 中执行。运行 WebPage0102.html 显示的页面如图 1-2 所示。

图 1-2 中第 1 行右边的空白方块称为文本框，是程序清单 1-2 中第 4 行生成的效果，用来输入姓名。图 1-2 中第 2 行的发送按钮，是程序清单 1-2 中第 5 行生成的效果。点击此按钮，就会执行 WebPage0103.aspx。若在图 1-2 中的文本框内输入“林旺”，如图 1-3，然后点击“发送”按钮，则显示的页面如图 1-4。

请输入姓名:

请输入姓名: 林旺

Hi,林旺你好

图 1-2 运行 WebPage0102.html

图 1-3 运行 WebPage0102.html

图 1-4 图 1-3 输出之结果

图 1-4 是 WebPage0103.aspx 被执行后所输出的结果。下面的程序就是 WebPage0103.aspx。

程序清单 1-3 WebPage0103.aspx

```
1  <%  
2      Response.Write("Hi," + Request("inName") + "你好")  
3  %>  
4  <html>  
5      <body>  
6      </body>  
7  </html>
```

程序清单 1-3 中的第 1 行到第 3 行是以脚本语言编写的程序，必须放在<%及%>之间。本书采用的脚本语言是 VB script，即将 VB.NET 略作修改以便于设计 Web 应用程序。第 2 行是 Response.Write 语句，其中 Response 是 ASP 提供的一个 object(对象)，Write 是 Response 对象中的一个 method(方法)。如果你不懂面向对象程序设计，可以把方法想象成一般程序中的子程序，即 Response 对象中有许多子程序，而 Write 是其中一个。第 2 行中的 Request 也是 ASP 提供的对象，是用来取得输入变量 inName 中的数据。第 2 行整行的意思是，将输入的姓名放在“Hi,”及“你好”之间后输出，输出的页面如图 1-4 所示。程序清单 1-3 中的第 4 行到第 6 行是多余的，因为标签中没有出现任何数据，所以没有这几行执行的结果也是一样。将这几行放在此处，是为了显示 ASP 是结合脚本语言和 HTML 网页语言的。

表 1-1 常用的 HTML (HtmlTextWriterTag) 标签

标签名称	说明
A	定义超级链接, 例: <code>超级链接文字</code> 。
B	黑体字, 例: <code>Hi, 你好</code> , 显示的效果为: Hi, 你好 。
Bgsound	用来说明背景音乐, 其中 <code>src</code> 特性是用来指示背景音乐的网址, <code>loop</code> 可用来说明播放的次数或重播。
Big	较大字体, 例: <code><big>Hi, 你好</big></code> , 显示的效果为: Hi, 你好 。
Body	网页文本须放在 <code><body></code> 及 <code></body></code> 之间。
Br	换行标签, 用法为 <code>
</code> 。
Button	按钮标签, 例: <code><button>Hi, 你好</button></code> 。
Caption	表格标题, 例: <code><caption>表格标题</caption></code> 。
Center	居中, 例: <code><center>Hi, 你好</center></code> , “Hi, 你好”会显示在网页的中央。
Del	用来说明文件中被删除的文字, 例: <code>Hi, 你好</code> 。
Dfn	用来说明定义的项目, 会以斜体字显示。
Dir	用来说明数据项, 通常会以 <code></code> 为子标签来说明各个分项, 例: <code><dir>ABC</dir></code> 。
Div	用来说明一个代码块的内容, 可以同时用几个 <code><div></code> 标签来设置多个代码块。
Font	设置字体、大小、颜色, 例: <code> Hi, 你好</code> , 显示的效果为: Hi, 你好 。
Form	窗体, 可定义输入数据后所要发送的程序。
H1~H6	设置字号, 从 <code><h1></code> 到 <code><h6></code> , 例: <code><h3>Hi, 你好</h3></code> 。
Head	页首文字须夹在 <code><head></code> 及 <code></head></code> 之间。
Hr	画水平线。
Html	网页语言声明, 所有网页的内容皆须夹在 <code><html></code> 及 <code></html></code> 之间。
I	斜体字, 例: <code><i>Hi, 你好</i></code> , 显示的效果为: <i>Hi, 你好</i> 。
Img	显示图形, 例: <code></code> 。
Input	输入窗体, <code>type</code> 有 <code>text</code> 、 <code>password</code> 、 <code>radio</code> 、 <code>checkbox</code> 、 <code>submit</code> 、 <code>reset</code> 等。
Li	用来表示菜单中的一个选项, 例子见 <code><dir></code> 标签。
Link	创建当前网页和外部网页的关系, 必须使用在 <code><head></code> 标签之内, 例: <code><head><link rel="stylesheet" href="styles.css" type="text/css"></head></code> 表示当前的文件要使用 <code>styles.css</code> 所描述的格式。
Map	影像地图, 例: <code><map><area>....</area><area>....</area></map></code> 。
Marquee	走马灯, 例: <code><marquee>Hi, 你好</marquee></code> 。
Menu	用来制作选项, 通常和 <code></code> 标签合用, 例: <code><menu>ab</menu></code> 。
Ol	用来显示项目的编号, 例: <code>AB</code> 。 <code></code> 标签中有 <code>start</code> 属性, 用来设置起始编号, 例: <code><ol start=3></code> 。也有 <code>type</code> 属性, 可以设置文字编号, 例: <code><ol type=a></code> 。
Option	Select 标签中的子标签, 用来定义菜单中的选项。

(续表)

标签名称	说明
P	空一行，用法为<p>。
Pre	编排好的文本须夹在<pre>及</pre>之间，即保留各行的缩进。
Script	用来描述此网页所使用的计算机语言。
Select	定义下拉列表。
Small	较小字体，例：Hi, 你好。
Span	用来当做文本容器，尤其当一整段文字要设置成相同的风格时，可将该段文字写在标签内，例：...。
Style	说明所要使用的样式表。
Sub	下标字体，例：Hi,_{你好}，显示的效果为：Hi _{你好} 。
Sup	上标字体，例：Hi,^{你好}，显示的效果为：Hi ^{你好} 。
Table	定义表格，例：<table><tr><td>字段一</td><td>字段二</td></tr></table>。
Textarea	用来定义输入的文本区域。
Title	标题文字须放在<title>及</title>之间。
Tr	定义表格的行。
Tt	正体字，例：<tt>Hi, 你好</tt>。
U	下划线字，例：<u>Hi, 你好</u>，显示的字体为：Hi _{你好} 。
Ul	枚举型选项，通常和 - 标签合并使用，例： - A - B 。

表 1-2 ASP 中常用的控件标签

名称	说明
form	窗体，可定义输入数据后所要发送的程序。
input	输入窗体，type 有 text、password、radio、checkbox、submit、reset 等。
option	Select 标签中的子标签，用来定义列表中的选项。
select	定义下拉列表。
textarea	用来定义输入的文本区域。

从上面的例子可以看出动态网页 ASP 比静态网页 HTML 多了一些控件标签，因而增加了系统和用户的交互性，使网页设计迈向了新的里程碑。然而处理一件事情却要写好几个程序来完成，例如上例中打印出“Hi,林旺你好”，要用 WebPage0102.html 及 WebPage0103.aspx 两个程序来完成，显然较繁杂。ASP.NET 对此做了大幅的改善，使得网页设计更加灵活方便。将上面的例子用 ASP.NET 来设计，见程序清单 1-4 中的 WebPage0104.aspx。