



ASP.NET 技术与技巧

ASP.NET Tips & Techniques

(美) Greg Buczek 著

程永敬 韩平 董启雄 等译

开发人员专业技术丛书

ASP.NET技术与技巧

(美) Greg Buczek 著

程永敬 韩平 董启雄 等译



机械工业出版社
China Machine Press

本书中包含了数以百计的技术与技巧，展示了如何使用ASP.NET及其相关技术。这些技术与技巧都是独立的，每一则技术和技巧都是以相同的格式来提供的。在相应的技巧或技术的标题之后，会提供一段关于该主题的简单描述，其后是完成相应任务所需要的代码或步骤。相信这种精心的编排格式定能助你快速发现渴望已久的解决方案，并在创建Web站点时游刃有余。

Greg Buczek: ASP.NET Tips & Techniques (ISBN: 0-07-222514-9) .

Copyright © 2002 by The McGraw-Hill Companies.

Original language published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳－希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书版权登记号：图字：01-2002-4184

图书在版编目（CIP）数据

ASP.NET技术与技巧 / (美) 布克泽克 (Buczek, G.) 著；程永敬等译. -北京：机械工业出版社，2003.2

(开发人员专业技术丛书)

书名原文：ASP.NET Tips & Techniques

ISBN 7-111-11580-5

I. A… II. ①布… ②程… 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004399号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李云静

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2003年2月第1版第1次印刷

787mm × 1092mm 1/16 · 29.5印张

印数：0 001-4 000册

定价：48.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

前 言

不断发展的ASP.NET方法使得创建动态Web应用程序的过程成为一种全新的技术。从某些方面来讲,这种技术变得更加容易了;而从另外某些方面讲,它又变得更加复杂了。无论怎样,你都将发现它能使所创建的Web应用程序更加符合商务需求,这便降低了其学习难度。无论是通过在Web页面中使用控件,发送E-mail,还是通过创建一个Web Service (Web服务),ASP.NET都能将通过Web所做的事进行拓展。

本书可以作为一本工具书使用。通过本书,可以理解ASP.NET是如何帮助你来开发站点的。当你试着使用ASP.NET来解决一个问题的时候,请浏览一下本书中所提供的一些技术和技巧,如果这些技术与技巧不能直接解决你的问题的话,那么就要思考一下:你怎样对本书中给出的这些代码做些调整以使之适用于自己的需求呢?别忘了,出色完成任务的开发人员都善于利用周围的资源,很少有人会不从其他页面拷贝/粘贴而只是自己来创建整个页面。

本书的结构

本书中包含了几百条技术和技巧,向你展示了如何使用ASP.NET及其相关技术。这些技术与技巧都是分别作为独立的单元提供的。换句话说就是,要使用并弄清楚一条技术或技巧,并不需要依赖之前的技术或技巧。

每一则技术和技巧都是以相同的格式来提供的。在技术或技巧的标题之后,会提供一段关于该主题的简单描述。其后是完成相应的任务所需要的代码或步骤。大部分的技术和技巧都有一个相应的ASP.NET页面,其中包含了该主题所涉及的代码。因此,你往往可以直接将所需要的代码拷贝并粘贴到应用程序中。

相似的技术和技巧组合在一起而形成一章。下面的内容对各章中所提供的技术和技巧进行了简要的说明。

第1章: ASP.NET页面的基本问题 在开篇的第1章,提供了与页面结构相关的一些技术和技巧。这些内容将向你展示一个页面的结构,以及如何在ASP.NET页面上定义编译器选项等内容。

第2章: HTML控件 由ASP.NET提供的最基本的控件库之一是通过HTMLControls命名空间来定义的。该库中的控件可以用来定义一些与基本的HTML元素密切对应的元素。该章提供的技术和技巧将向你展示如何使用这些控件。

第3章：基本的Web控件 ASP.NET所带的库中可能是最重要的，而也可能是最常用到的库就是WebControls命名空间。该章提供的技术和技巧展示了这个库中包含的基本控件的用法。

第4章：用于列表的Web控件 该章继续讲解使用WebControls命名空间的技术和技巧。该章提供了用来列出重复数据的控件如DataGrid控件和Repeater控件的技术和技巧。

第5章：用于验证的Web控件 ASP.NET中包含了许多可以方便地用于对访问者的输入进行验证的控件。这些控件可以用来确认访问者是否在某个字段中输入了数据，检查所输入数据的类型，或者检查所输入的数据是否与相应的模式相匹配。该章中所提供的技术和技巧将向你展示这些控件的用法。

第6章：其他Web控件 该章所提供的技术和技巧是用于那些WebControls命名空间中较为高级或专用的控件的。这里所讨论的控件包括Calendar（日历）控件、AdRotator（旋转广告）控件、Panel（面板）控件以及Table（表格）控件。

第7章：Internet Explorer Web控件 Internet Explorer（即IE）Web Controls命名空间是一个附加的也是非常有用的控件库，可以从微软的站点上下载。这个命名空间所包含的控件有：MultiPage（多页面）控件、TabStrip控件、ToolBar（工具条）控件以及TreeView（树状视图）控件等。该章中所提供的技术和技巧将向你展示这些控件的用法。

第8章：创建自定义控件 ASP.NET所提供的一种非常好的功能就是可以创建自定义的控件。通过创建自定义的控件，你可以将想重复使用的功能封装到一个源中。该章所提供的技术和技巧将向你展示如何创建和使用自定义控件。

第9章：E-mail 在你觉得需要通过ASP.NET页面发送E-mail之前，你所创建的ASP.NET应用程序还不会复杂到哪里去。你可以通过ASP.NET来实现这些相对复杂的内容。该章中所提供的技术和技巧将向你展示如何在ASP.NET页面中处理E-mail。

第10章：应用程序的问题 在掌握了使用ASP.NET进行单页面的开发工作之后，你就可以将所创建的页面组合到一个ASP.NET应用程序中了。该章中所提供的技术和技巧将向你展示如何在ASP.NET中处理与应用程序有关的问题。

第11章：后台代码文件 ASP.NET提供了一种技术，这种技术使得我们可以将代码与设计界面分离开来。这样就可以由一个开发人员来做编写代码的工作，而另一个人来负责页面的设计工作。每一个开发人员都分别处理其自己的文件。该章所提供的技术和技巧将向你展示如何将代码与相应的页面设计分离开来。

第12章：使用IIS 如果要创建ASP.NET页面或应用程序，那么使用IIS将使你的工作更加容易。该章所提供的技术和技巧将向你展示如何在IIS中执行各种任务。

第13章：数据库代码 随着Web站点复杂性的不断增长，你会想提供来自数据库的动态内容。该章所提供的技术和技巧将向你展示通过ASP.NET连接到数据库、从数据库中检索数据、向数据库中插入数据等有关内容。

第14章：Web服务 Web服务提供了一种方式，利用这种方式，可以将可用于同一服务器上其他页面或站点、或者可由其他服务器上其他开发人员所使用的功能进行封装。该章中所提供的技术和技巧将向你展示Web服务的创建、编译及使用。

第15章：Mobile Internet Toolkit Internet访问方式的变化之一就是，访问者逐渐开始使用像移动电话这样的移动设备造访你的网站了。如果你希望为那些通过移动设备来连接的访问者创建专门的页面和站点，那么，现在就可以使用移动Internet工具箱（Mobile Internet Toolkit）来完成这项工作。该章所提供的技术和技巧将向你展示如何创建Mobile（移动）ASP.NET页面。

第16章：页面示例 在本书的最后一章中，我们提供了一些页面示例。这些例子将会把全书中用到的控件组合起来，以创建更为复杂的页面。

下载本书示例代码

本书中所提供的绝大多数技术和技巧都有一份ASP.NET页面或文件，其中包含了与主题相关的代码。你可以到<http://www.osborne.com>这个站点上，在本书主题下点击“Free Downloadable Code”有关的链接来下载这些代码。

将下载的压缩包解压到你计算机上的某个指定位置。你将发现这些文件是按照章的序号来组织的。在每一章的文件夹中，可找到相应的文件。

目 录

前言

第1章 ASP.NET页面的基本问题	1
1.1 编写基本的ASP.NET页面框架	1
1.2 使用Page对象的Load事件	3
1.3 使用Page对象的Unload事件	4
1.4 当发生不能解决的问题时将访问者重定向	5
1.5 使用Page对象的IsPostBack属性	5
1.6 创建自定义的过程	7
1.7 使用Page指令	8
1.8 使用Import指令	10
1.9 使用OutputCache指令	10
第2章 HTML控件	12
2.1 使用HTMLAnchor控件	12
2.2 创建一个按钮形式的HTMLAnchor控件	14
2.3 使用HTMLButton控件	14
2.4 在HTMLButton控件上显示图形	15
2.5 定义一个基本的Span控件	16
2.6 使用HTMLForm控件	18
2.7 通过HTMLImage控件显示图像	19
2.8 使用HTMLInputText控件	21
2.9 使用HTMLInputCheckBox控件	22
2.10 使用HTMLInputFile控件上传文件	23
2.11 通过HTMLInputHidden控件将值放置在表 单上	25
2.12 使用HTMLInputImage控件	26
2.13 通过HTMLInputImage控件得到被 点击的坐标	27
2.14 允许通过HTMLInputRadioButton控件 进行输入	28
2.15 使用HTMLSelect控件	29
2.16 使用HTMLTextArea控件	31
2.17 通过HTMLTable控件显示信息	32
2.18 用代码在HTMLTable控件中添加行和 单元格	34
第3章 基本的Web控件	37
3.1 创建基本的TextBox控件	37
3.2 使用TextBox控件用于密码的输入	42
3.3 创建多行TextBox控件	44
3.4 用代码动态添加TextBox控件	47
3.5 编写当TextBox控件中的文本更改时触 发运行的代码	50
3.6 创建一个基本的Label控件	51
3.7 在代码中操纵一个Label控件	54
3.8 显示一个基本的RadioButton控件	54
3.9 在RadioButton控件组中设置并获取选 中的条目	57
3.10 编写当RadioButton控件选中状态发 生改变时触发运行的代码	59
3.11 显示一个基本的CheckBox控件	61
3.12 编写当某个CheckBox控件被选中时 运行的代码	64
3.13 创建一个基本的HyperLink控件	65
3.14 在HyperLink控件上显示图片	66
3.15 创建一个基本的LinkButton控件	67
3.16 将LinkButton控件用做一个Command 按钮	69
3.17 创建一个基本的Image控件	71
3.18 创建一个基本的ImageButton控件	72
3.19 确定在ImageButton控件的图像上所 点击的坐标	74
3.20 在ASP.NET页面上添加一个简单的 Button控件	76
3.21 将Button控件用做Command按钮	77
第4章 用于列表的Web控件	80

4.1 创建一个基本的DropDownList控件	81	4.31 将RadioButtonList控件绑定到一个数据源	146
4.2 处理DropDownList控件的外观	82	4.32 创建一个基本的CheckBoxList控件	148
4.3 对DropDownList控件中选中的条目进行读和写操作	85	第5章 用于验证的Web控件	150
4.4 编写当选中DropDownList控件中的一个条目时所触发运行的代码	87	5.1 使用RequiredFieldValidator控件	150
4.5 将数据库中的数据绑定到一个DropDownList控件	88	5.2 利用一个RequiredFieldValidator控件来对初始值进行检查	152
4.6 在代码中创建DropDownList控件	91	5.3 通过RangeValidator控件检查数值的范围	153
4.7 创建一个基本的DataList控件	94	5.4 通过RangeValidator控件检查日期的范围	155
4.8 在DataList控件中使用模板	97	5.5 通过CompareValidator控件根据一个值对用户的输入进行验证	155
4.9 在DataList控件中进行行的删除	101	5.6 使用CompareValidator控件依据另一个控件中的值对用户的输入进行验证	156
4.10 对DataList控件中的行进行编辑	104	5.7 使用CompareValidator控件将一个字段与日期数据类型进行对比	157
4.11 在DataList控件中添加行	108	5.8 使用CompareValidator控件将一个字段与数值数据类型进行对比	158
4.12 创建一个基本的Repeater控件	110	5.9 在CompareValidator控件中使用一个比较操作符	159
4.13 在Repeater控件中使用模板	112	5.10 使用一个RegularExpressionValidator控件来检查一个ZIP码	159
4.14 创建一个基本的DataGrid控件	114	5.11 使用RegularExpressionValidator控件来阻止输入指定的特殊字符	161
4.15 在DataGrid控件中创建BoundColumn控件	117	5.12 利用RegularExpressionValidator控件来验证电话号码	161
4.16 在DataGrid控件中使用HyperLinkColumn控件	120	5.13 利用RegularExpressionValidator控件对E-mail地址进行验证	163
4.17 在DataGrid控件中使用ButtonColumn控件	123	5.14 使用CustomValidator控件依据一个可能值的范围进行验证	164
4.18 在DataGrid控件中使用模板	125	5.15 使用CustomValidator控件依据一个正偶数进行验证	165
4.19 在DataGrid控件中进行排序	128	5.16 使用CustomValidator控件验证一个百分数	166
4.20 通过DataGrid控件来显示数据页面	130	5.17 使用CustomValidator对当年的日期进行验证	168
4.21 在DataGrid控件中进行行的删除	131	5.18 创建一个基本的ValidationSummary控件	169
4.22 从DataGrid控件中对行进行编辑	134		
4.23 向DataGrid控件中添加行	137		
4.24 创建一个基本的ListBox控件	138		
4.25 检查ListBox控件中呈选中状态的值	139		
4.26 允许在ListBox控件中进行多条目选择	140		
4.27 编写当ListBox控件中条目的选中状态发生改变时触发运行的代码	141		
4.28 将一个ListBox控件绑定到一个数据源	142		
4.29 创建一个基本的RadioButtonList控件	143		
4.30 编写所需代码, 当在RadioButtonList控件中选中一个条目时触发运行它	145		

5.19 使用ValidationSummary控件创建一个错误列表	171	6.23 使用Table控件创建一个HTML表格	199
5.20 使用ValidationSummary控件在一个消息框中显示错误消息	172	6.24 在代码中创建Table控件	201
第6章 其他Web控件	174	6.25 利用代码在Table控件中添加一行	202
6.1 创建一个基本的Calendar控件	175	第7章 Internet Explorer Web控件	204
6.2 格式化Calendar控件中的标题样式	177	7.1 创建基本的MultiPage控件	205
6.3 对Calendar控件中的日期部分进行格式化	178	7.2 在MultiPage控件中添加一个PageView控件	206
6.4 显示或隐藏Calendar控件的各个部分	180	7.3 将MultiPage控件用做向导的界面	208
6.5 在Calendar控件中设置自定义的前一月份和后一月份的显示文本	181	7.4 使用DropDownList在MultiPage控件中提供导航	209
6.6 编写当在Calendar控件中选中一个日期时触发运行的代码	182	7.5 利用代码在PageView控件中添加控件	210
6.7 显示Calendar控件中的一个日期	182	7.6 创建一个基本的TabStrip控件	212
6.8 允许在Calendar控件中选中多个日期	183	7.7 设置TabStrip控件的方向	212
6.9 通过Calendar控件读取一个日期范围中选中的那些日期	185	7.8 对TabStrip控件中的Tab控件进行设置	213
6.10 显示Calendar控件中的一个选中日期范围	186	7.9 在TabStrip控件中使用TabSeparator控件	214
6.11 编写当Calendar控件中的月份发生更改时触发运行的代码	187	7.10 设置TabStrip控件中的样式	215
6.12 对Calendar控件中日期单元格的外观进行控制	188	7.11 在TabStrip控件中的Tab控件上使用图像	216
6.13 为AdRotator控件创建一个Ad文件	188	7.12 联合使用TabStrip控件和MultiPage控件	216
6.14 使用基本的AdRotator控件	190	7.13 创建一个基本的ToolBar控件	218
6.15 为AdRotator控件创建一个使用Keyword标记的Ad文件	190	7.14 在ToolBar控件上添加一个ToolBarButton控件	219
6.16 使用定义了关键字的AdRotator控件	191	7.15 在ToolBar控件中添加ToolBarCheckButton控件	221
6.17 编写当使用AdRotator控件创建一条广告时触发运行的代码	192	7.16 在ToolBar控件中添加ToolBarCheckGroup控件	222
6.18 为AdRotator控件创建一个使用自定义标记的Ad文件	193	7.17 在ToolBar控件中添加ToolBarDropDownList控件	223
6.19 在AdRotator控件中使用自定义标记	193	7.18 在ToolBar控件中添加ToolBarLabel控件	224
6.20 创建基本的Panel控件	194	7.19 在ToolBar控件中添加ToolBarTextBox控件	225
6.21 使用Panel控件来控制代码流	195	7.20 在ToolBar控件中添加ToolBarSeparator控件	227
6.22 设置Panel控件的样式	198	7.21 使用ToolBar控件的样式属性	227
		7.22 通过代码在一个ToolBar控件中添加其他控件	228
		7.23 创建一个基本的TreeView控件	229
		7.24 在TreeView控件中使用TreeNode控件	231

7.25 在TreeView控件中添加TreeNode CheckBox 控件	233	10.1 创建Global.asax配置文件	269
7.26 在TreeView控件中将TreeNode控件作为 链接使用	233	10.2 编写Application_OnStart事件 处理代码	270
7.27 在TreeView控件的TreeNode控件中使用 图像	234	10.3 编写Application_OnEnd事件代码	272
7.28 在TreeView控件中设置样式属性	236	10.4 编写Session_OnStart事件代码	273
7.29 编写当在TreeView控件中选中某个 TreeNode控件时触发运行的代码	236	10.5 编写Session_OnEnd事件代码	273
7.30 编写当展开一个TreeNode时所触发运行 的代码	237	10.6 在Global.asax文件中使用错误处理程序	274
7.31 编写当TreeNode控件折叠时触发运行 的代码	238	10.7 使用Session变量	274
第8章 创建自定义控件	239	10.8 设置并获取Session属性	276
8.1 在ASP.NET页面上实现用户控件	239	10.9 使用Application变量	278
8.2 向自定义控件中添加控件	241	10.10 Application变量的锁定与解锁	278
8.3 在用户自定义控件中创建只读属性	243	10.11 在Global.asax文件中创建Application 对象	279
8.4 在用户自定义控件中创建只写的属性	245	10.12 在Global.asax文件中创建Session对象	279
8.5 在用户自定义控件中创建可读写的属性	246	10.13 使用Web.Config文件	280
8.6 在用户自定义控件中添加Sub类型的 方法	248	10.14 使用Web.Config文件的Errors部分	281
8.7 向用户自定义控件中添加Function类型的 方法	249	10.15 在Web.Config文件中添加Application 设置	282
8.8 在用户自定义控件中使用事件处理过程	250	10.16 在代码中获取Web.Config应用程序 设置	283
8.9 创建一个无界面的用户自定义控件	251	10.17 通过Web.Config文件来设置Session 属性	284
第9章 E-mail	253	10.18 使用HttpBrowserCapabilities类来确定 访问者的浏览器类型	285
9.1 在代码中发送一则简单的E-mail消息	253	10.19 使用HttpBrowserCapabilities类来确定 访问者的操作系统	285
9.2 直接使用Send方法	255	10.20 使用HttpBrowserCapabilities类来检查 访问者浏览器的脚本与代码能力	286
9.3 设置一封邮件的优先级	256	10.21 检测访问者是否正在使用一个测试版的 浏览器	287
9.4 向其他接收者发送邮件	257	10.22 使用HttpBrowserCapabilities类来确定 访问者的浏览器是否支持cookie	288
9.5 将文件附加到一则邮件消息中	258	10.23 根据访问者浏览器类型将其重定向 到相应的页面	289
9.6 在一则邮件消息中发送多个文件	260	10.24 确定访问者的IP地址和主机名称	289
9.7 发送一则HTML格式的邮件消息	262	10.25 检查访问者来自何方	290
9.8 在一则邮件消息中嵌入图片	263	10.26 通过ServerVariables集合来获取客户端	
9.9 在邮件消息中添加一个表单	265		
9.10 邮件群发	266		
第10章 应用程序的问题	268		

与服务器端信息	291	映射	324
第11章 后台代码文件	293	12.26 在ASP.NET应用程序中启用会话状态	325
11.1 创建并链接至一个后台代码文件	293	12.27 为ASP.NET应用程序设置脚本超时值	326
11.2 在后台代码文件中创建过程	294	12.28 删除一个ASP.NET应用程序	326
11.3 继承后台代码文件中的控件	297	12.29 重定向文件请求	327
11.4 继承后台代码文件中的事件	298	12.30 备份与恢复IIS配置	327
11.5 创建多个ASP.NET页面中使用的后台 代码文件	301	第13章 数据库代码	329
11.6 创建一个Include代码库	302	13.1 连接到Access数据库	330
11.7 链接到一个Include代码库	303	13.2 检索Access数据	331
第12章 使用IIS	304	13.3 在DropDownList控件中显示数据	333
12.1 设置应用程序保护	305	13.4 在DataGrid控件中显示数据	334
12.2 创建Web站点	306	13.5 添加Access数据	336
12.3 多网站共享一个IP地址	307	13.6 更新Access数据	337
12.4 限制Web站点的连接数量	308	13.7 删除Access数据	339
12.5 记录页面活动日志	309	13.8 从Access数据库中检索单个值	341
12.6 将页面活动日志记录到数据库	309	13.9 对一个Access数据库表中的记录进行 迭代	342
12.7 从需要记录日志的页面中排除一些文件	311	13.10 使用Access数据库的事务处理程序	343
12.8 限制使用网站资源	311	13.11 连接到SQL Server数据库	345
12.9 设置网站位置	312	13.12 检索SQL Server数据	346
12.10 允许浏览目录	313	13.13 在DropDownList控件中显示SQL Server 数据	347
12.11 允许文件上载	314	13.14 在DataGrid控件中显示SQL Server 数据	348
12.12 设置执行权限	314	13.15 添加SQL Server数据	349
12.13 设置默认文档	315	13.16 更新SQL Server数据	351
12.14 使用文档页脚	316	13.17 删除SQL Server数据	352
12.15 通过IP地址限制Web访问	316	13.18 从SQL Server数据库中检索单个值	353
12.16 取消对站点的匿名访问	317	13.19 对SQL Server数据库表中的记录进行 迭代	355
12.17 创建自定义标头	318	13.20 使用SQL Server数据库的事务	356
12.18 编辑错误页	319	13.21 调用一个存储过程	357
12.19 使用纯文本错误信息	320	13.22 通过存储过程来检索数据	358
12.20 将属性继承到新站点和文件夹	320	13.23 通过输入参数来给存储过程传递数据	360
12.21 压缩Web站点文件	321	13.24 使用存储过程的输出参数	361
12.22 创建虚拟目录	322	第14章 Web服务	363
12.23 创建ASP.NET应用程序	323	14.1 理解为什么需要创建Web服务	363
12.24 为ASP.NET应用程序设置执行与运行 模式属性	323		
12.25 设置ASP.NET应用程序中的应用程序			

14.2 创建一个WebService文件	364	验证数据类型	395
14.3 在WebService中包含命名空间引用和 其他的指令	365	15.20 创建一个基本的Mobile Image控件	396
14.4 在WebService中添加Sub方法	365	15.21 创建一个基本的Mobile Label	397
14.5 在WebService中添加Function	367	15.22 使用Mobile Label控件的StyleReference 属性	398
14.6 测试WebService	372	15.23 创建一个基本的Mobile Link控件	399
14.7 创建WebService的类文件	374	15.24 创建一个基本的Mobile List控件	400
14.8 编译WebService	374	15.25 在代码中给Mobile List控件添加一个 选项	401
14.9 在ASP.NET页面中调用WebService	375	15.26 从Mobile List控件中删除选项	401
第15章 Mobile Internet Toolkit	376	15.27 使用Mobile List控件来显示超链接 列表	402
15.1 获取Mobile Internet Toolkit	377	15.28 将Mobile List控件绑定到一个数据源	403
15.2 获取Microsoft Mobile Explorer	378	15.29 使用MobileCapabilities类来确定 浏览器的类型	404
15.3 使用Microsoft Mobile Explorer	378	15.30 使用MobileCapabilities类来确定屏幕的 大小	405
15.4 创建Mobile ASP.NET页面	380	15.31 使用MobileCapabilities类来确定访问者的 设备是否支持颜色	405
15.5 使用Mobile表单	381	15.32 使用MobileCapabilities对象来确定设备 类型特性	406
15.6 在代码中引用Mobile表单	382	15.33 创建一个基本的Mobile ObjectList 控件	407
15.7 为Mobile AdRotator控件创建一个 AdRotator Schedule文件	383	15.34 在Mobile ObjectList控件中创建自己的 字段	409
15.8 使用Mobile AdRotator控件	385	15.35 在Mobile表单中添加一个Mobile Panel 控件	410
15.9 编写创建一个Mobile Ad所触发的事件 代码	386	15.36 通过代码来给Mobile Panel控件添加 子控件	411
15.10 创建一个基本的Mobile Calendar控件	387	15.37 使用Mobile Panel控件作为一个向导 界面	412
15.11 在Mobile Calendar控件中设置或者检索 选中的日期	388	15.38 使用Mobile RangeValidator控件	413
15.12 允许通过Mobile Calendar控件来选择 多个日期	389	15.39 使用Mobile RequiredFieldValidator控件	415
15.13 检索在Mobile Calendar控件中选择的 多个日期	389	15.40 利用Mobile RequiredFieldValidator控件 针对初始值来进行检验	416
15.14 创建一个基本的PhoneCall控件	390	15.41 创建一个DropDownList Mobile SelectionList控件	417
15.15 在PhoneCall控件内提供一个替换的 超链接	391		
15.16 创建一个基本的Mobile Command控件	391		
15.17 创建一个基本的Mobile Compare Validator控件	393		
15.18 使用Mobile CompareValidator控件来比较 两个Mobile TextBox控件中的值	394		
15.19 使用Mobile CompareValidator控件来			

15.42 创建一个ListBox Mobile SelectionList 控件.....	417	16.3 创建一个用于定位商店位置的页面	430
15.43 创建一个Radio Mobile SelectionList 控件.....	418	16.4 生成一个Contact Us页面	433
15.44 将Mobile SelectionList控件绑定到一个 数据源.....	418	16.5 显示一个Survey页面	435
15.45 创建一个基本的Mobile TextBox控件	420	16.6 生成一个Employee of the Month (本月雇员) 页面.....	438
15.46 创建一个只允许数字输入的Mobile TextBox控件	420	16.7 创建一个What's New页面	439
15.47 创建用于密码输入的Mobile TextBox 控件.....	421	16.8 显示一个New Products页面	441
15.48 创建一个基本的Mobile TextView控件.....	421	16.9 创建一个Tip of the Day (每日提示) 页面	442
15.49 使用Mobile ValidationSummary控件	422	16.10 在ASP.NET页面上显示查询结果	444
第16章 页面示例	425	16.11 创建一个Help Desk页面	447
16.1 创建一个登录页面	425	16.12 显示一个Links页面	450
16.2 显示一个电话名册	428	16.13 给Links页面添加一个超链接	451
		16.14 记录页面的活动.....	453
		16.15 显示页面活动.....	454
		16.16 创建一个测验页面.....	456

第1章 ASP.NET页面的基本问题

本章提供的技巧

- 编写基本的ASP.NET页面框架
- 使用Page对象的Load事件
- 使用Page对象的Unload事件
- 当发生不能解决的问题时将访问者重定向
- 使用Page对象的IsPostBack属性
- 创建自定义的过程
- 使用Page指令
- 使用Import指令
- 使用OutputCache指令

当你创建一个ASP.NET页面时，其中会包含一些特定的项，这些项需要以一种所要求的方式进行定义。为了使得你所做的绝大多数开发工作得以正常进行，你需要熟悉ASP.NET页面的结构。本章中提供的技巧和方法将向你展示如何高效地定义一个ASP.NET页面。此外，本章中还提供了有关Page对象的一些方法和技巧，其中的Page对象是一种底层的类，所有的ASP.NET页面都是基于它建立起来的。

1.1 编写基本的ASP.NET页面框架

当你创建一个ASP.NET页面时，这个页面需要包含一些特定的标记，同时，这些标记要具有预期的结构。本技巧定义了一个ASP.NET页面的基本结构。

 为本条技巧所创建的页面包含了一个ASP.NET页面的基本结构。在这个页面的顶部，你将发现这些指令：

```
<%@ Page Language=VB Debug=true %>  
<%@ Import Namespace="System.Web.Mail" %>
```

指令是一些命令代码，它被发送给编译器，并将页面应该怎样来运行通知编译器。在本例中，Page指令用来指出在本页面中所用的编码语言。Import指令指出应该编译到本页面中的命名空间。

之后是你的页面代码，这些代码放在了Script标记之中。

```
<script runat=server>  
Private A as Integer  
Sub Page_Load(ByVal Sender as Object, ByVal E as EventArgs)  
  
End Sub
```

```
Sub SubmitBtn_Click(Sender As Object, E As EventArgs)

End Sub

</script>
```

注意, **Script**标记包含了一个参数**RunAt**, 这个参数的值这里设置为**Server**。这个意思是说这段代码是服务器端代码。此外, 当本页面载入或者回送到服务器进行处理时, 编译器将用到这个参数。

在**Script**标记中, 定义了本页面所需的过程代码。这些过程可由一些事件过程组成, 当在本页面上发生特定的情况——比如当点击一个按钮或者是页面载入时, 触发相应的事件过程。这些过程也可以由你自己(自定义)的过程所组成, 而在其他的一些事件驱动的过程中调用这些自定义过程。

在为页面定义了相应代码之后, 就可以开始为这个页面结构中加入**HTML**标记了:

```
<HTML>
<HEAD>
<TITLE>Coding the Basic ASP.NET Page Structure</TITLE>
</HEAD>
<Body LEFTMARGIN="40">
```

通常其后将跟随一个**ASP.NET Form**(表单)的开始标记:

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

在这个**Form**标记中, 可以与标准的 **HTML**元素混合在一起定义你的**ASP.NET**控件:

```
<asp:label
    id="lblDataEntered"
    runat="server"
/>
<BR><BR>
Enter your Name:<BR>
<asp:textbox
    id="txtName"
    columns="25"
    maxlength="30"
    runat=server
/>
<BR><BR>
<asp:button
    id="butOK"
    text=" OK "
    onclick="SubmitBtn_Click"
    runat="server"
/>
```

在这里, 定义了一个**Label**(标签)控件、一个**TextBox**(文本框)控件和一个**Button**(按钮)

控件。注意，在这些控件的定义之间，还定义了一些HTML元素。

在完成了页面上控件和其他内容的定义之后，就需要关闭ASP.NET的Form控件：

```
</Form>
</BODY>
</HTML>
```

注意同时也关闭了相应的HTML标记。

1.2 使用Page对象的Load事件

通常来讲，你需要在页面中包含一些在页面载入的时候触发执行的代码。这种类型的代码用来对页面中包含的各项内容进行设置。例如，如果你想要显示一个产品列表，那么就需要在访问者浏览这个页面时，从数据库中检索这些产品。你可以创建一个用于Page对象的Load事件过程来实现这种功能。这里给出的方法向你展示了如何创建这样的过程。

 为本方法所创建的页面，通过两个Label控件向访问者显示相应的文本信息。这两个控件中的文本内容在页面加载时通过代码来设定。

这些Label控件在这个ASP.NET页面上的Form中进行定义。第一个Label控件的定义如下：

```
<asp:label
    id="lblFixed"
    runat="server"
/>
```

另一个Label控件如下定义：

```
<asp:label
    id="lblStatic"
    runat="server"
/>
```

当这个页面载入的时候，运行下列代码：

```
Sub Page_Load(ByVal Sender as Object, ByVal E as EventArgs)
    lblFixed.Text = "Welcome to the page!"
    lblStatic.Text = "The current time on the server is ' _
        & TimeOfDay()
End Sub
```

注意，这个过程的名字叫Page_Load：

```
Sub Page_Load(ByVal Sender as Object, ByVal E as EventArgs)
```

如果想无论何时加载这个页面都运行这段代码，那么就必须为这个过程取Page_Load这个名字。此外，还要注意的是给这个过程传递了两个参数。如果不提供这两个参数，那么这个页面将不会运行，而返回一个错误。

在这个过程中，定义了一段每当本页面加载时都去运行的代码：

```
lblFixed.Text = "Welcome to the page!"  
lblStatic.Text = "The current time on the server is " _  
    & TimeOfDay()
```

在本例中，第一个Label控件中显示了一段固定的文本内容。而当前系统时间则显示在另一个Label控件中。

之后，需要关闭这个过程的定义：

```
End Sub
```

1.3 使用Page对象的Unload事件

你可以为其提供代码的另一个Page对象事件过程是Unload事件。当页面过程处理结束并从内存中删除的时候，触发此事件。你可以使用这个过程来编写任意的最后清除代码，例如关闭到数据库的连接或保存一些正在关闭的静态值。注意，即使是这样，你也不能在此事件中更改页面上的控件，因为这些控件这时已经要释放了。这则方法向你展示了Unload事件的使用法。

 为本文本所创建的页面在页面加载时向访问者显示一条文本信息。然后当页面完成其处理时，将页面结束的时间记录在一个永久变量中。

在这个ASP.NET页面中定义了下列三个Label控件：

```
<asp:label  
    id="lblFixed"  
    runat="server"  
>  
<BR><BR>  
<asp:label  
    id="lblStatic"  
    runat="server"  
>  
<BR><BR>  
<asp:label  
    id="lblDone"  
    runat="server"  
>
```

当页面加载时触发下列代码：

```
Sub Page_Load(ByVal Sender as Object, ByVal E as EventArgs)  
    lblFixed.Text = "Welcome to the page!"  
    lblStatic.Text = "The current time on the server is " _  
        & TimeOfDay()  
    lblDone.Text = Session("FinalText")  
End Sub
```

注意，第三个Label控件中的文本在一个Session（会话）变量中进行了赋值。这个变量在Unload事件中获取其值，因此当页面首次加载时，这个Label将不会显示任何文本。但当下次调