

实用百例

ASP.NET

应用开发百例



Programmer to Programmer

资 深 程 序 员 的 力 作



俞雷 周松建 编著

清华大学出版社

ASP.NET 应用开发百例

俞雷 周松建 编著

清华大学出版社

北 京

内 容 简 介

本书通过 100 个具体的 ASP.NET 实例详细讲解了如何使用 ASP.NET 进行 Web 开发。本书中的实例涉及有关 Web 开发的各个方面,包括基础知识、各种控件的使用、数据库、图形图像、Web Service、移动开发等。针对每一方面都选择了有代表性的实例进行分步骤的详细讲解,每个实例的知识点也都重点提出,读者能够迅速从中学习到相关方面的核心知识。

本书语言简练、实例丰富、可操作性强,主要面向 ASP.NET 的初、中级用户以及各 IT 企业的程序员。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

ASP.NET 应用开发百例/俞雷,周松建编著. —北京:清华大学出版社,2003

ISBN 7-302-07511-5

I. A… II. ①俞… ②周… 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 097557 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:胡伟卷

文稿编辑:刘金喜

封面设计:张海滨

版式设计:康博

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:18 字数:426 千字

版 次:2003 年 12 月第 1 版 2003 年 12 月第 1 次印刷

书 号:ISBN 7-302-07511-5/TP·5534

印 数:1~5000

定 价:28.00 元

前 言

ASP.NET 简介

ASP.NET 是 ASP 的新一代版本,然而 ASP.NET 并非从 ASP 3.0 自然演化而来,在很多方面,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,数据处理方面也引入了许多具有开创性的新技术。这些具有革新意义的特性让 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 Server 2003 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。

本书面向的读者

本书面向 ASP.NET 的初、中级用户,全面系统地介绍了 ASP.NET 的特点、基础知识和具体应用。即使用户是一名高手,这本书也有很好的参考价值。

本书由浅入深、循序渐进地讲解了 ASP.NET 技术,并通过 100 个具体的实例真实地演示了这些技术是如何运用在实际开发中的。本书中的例子都使用了 Visual Basic.NET 的语法。所以,如果对 Visual Basic 的语法或对 ASP 很熟悉的话,很快就可以上手。如果用户精通别的编程语言,通过阅读本书,也会跟上编程技术发展的前沿。

本书的组织结构

本书共分为 10 章,从 ASP.NET 的基础运用实例到高级的 Web Service 开发和移动 Web 开发实例,都有详细的讲述和具体的操作步骤。

第 1 章通过 20 个具体的实例介绍了 ASP.NET 开发的基础知识。

第 2 章通过 9 个实例详细介绍了在 ASP.NET 中如何使用 HTML 服务器控件。

第 3 章通过 10 个实例详细介绍了最常用的 Web 服务器控件的使用方法。

第 4 章通过 9 个实例详细介绍了如何在 ASP.NET 中对文件和目录进行操作。

第 5 章通过 12 个实例介绍了在 ASP.NET 中对图形图像进行处理的各个方面。

第 6 章通过 12 个实例介绍了 ADO.NET 在各个方面的应用。

第 7 章通过 12 个实例介绍了在 ASP.NET 中如何对 XML 文件进行分析和操作。

第 8 章通过 5 个实例介绍了在 ASP.NET 中如何使用 Web Service。

第 9 章通过 7 个实例介绍了如何在 ASP.NET 中进行移动开发。

第 10 章通过 4 个实例介绍了如何在 ASP.NET 中完成一个论坛和一个留言板类型的程序。

学习本书需要使用的工具

本书中所有的实例均基于 .NET Framework SDK v1.1 开发, 因此用户需要配置一个运行环境, 最简单的就是安装这个 SDK 开发包, 可以从微软公司的网站上下载该开发包, 也可以安装 Visual Studio.NET 开发环境。本书建议安装 Visual Studio.NET 开发环境, 因为本书中的实例全部基于此开发环境。

本书主要由俞雷、周松建编写, 此外, 非常感谢金哲和刘秋红在本书写作过程中给予的无私帮助与支持, 同时也感谢下列人员在录入排版中付出的辛勤劳动: 张勇、刘兵、马强、吴军、陆新、郭青、董迎、牟江、何栋、顾影、王修、杨伟、夏厅、巩萍、余利、江华、宋刚、隋炎、杜贺、赵耕、童鸣、冯海、李芬、戈昂、钱静、贺顿、丁力等。

由于时间仓促, 书中难免存在不足之处, 敬请广大专家、读者提出宝贵意见(作者的 E-mail: yulei@yeah.net)。本书中的实例代码, 读者可到 <http://www.tupwk.com.cn/downpage/index.asp> 处下载。

编 者

目 录

第 1 章 基本应用	1
实例 1 文字显示	1
实例 2 动态生成随机数	3
实例 3 页面跳转	5
实例 4 发送邮件	7
实例 5 检索浏览器信息	9
实例 6 检索服务器环境变量	12
实例 7 使用程序生成页面内容	14
实例 8 动态改变文字显示样式	16
实例 9 IP 地址验证	19
实例 10 IP 地址和主机名相互查找	22
实例 11 显示 Access 数据库的数据	24
实例 12 显示 SQL Server 数据库的数据	27
实例 13 创建 Web 用户控件	29
实例 14 调用 COM 组件	31
实例 15 调用 C#类库	32
实例 16 调用 WebService	34
实例 17 捕捉异常	37
实例 18 区域性演示	39
实例 19 页面跟踪	41
实例 20 应用程序级跟踪	43
第 2 章 HTML 服务器控件	46
实例 21 添加 HTML 服务器控件	46
实例 22 上传文件	48
实例 23 设置背景颜色	50
实例 24 动态创建表格	53
实例 25 照片预览	55
实例 26 用户身份验证	57
实例 27 保存用户信息	59
实例 28 获取 HTML 服务器控件的属性	61

实例 29 Panel 控件示例.....	64
第 3 章 Web 服务器控件	67
实例 30 动态公告条.....	67
实例 31 日历.....	70
实例 32 比较变量.....	72
实例 33 验证控件.....	73
实例 34 DataGrid 控件显示数据	75
实例 35 图像按钮.....	78
实例 36 占位符示例.....	80
实例 37 面板示例.....	82
实例 38 Repeater 控件示例	84
实例 39 编辑 DataList 数据项.....	88
第 4 章 文件和目录操作.....	94
实例 40 显示文本文件信息	94
实例 41 文件信息保存	97
实例 42 向文件中追加文本	100
实例 43 删除、复制和移动文件	102
实例 44 浏览文件属性信息	105
实例 45 浏览目录属性信息	107
实例 46 浏览目录包含的文件和子目录	109
实例 47 遍历目录.....	112
实例 48 获取系统特殊文件夹的路径	115
第 5 章 图形图像处理	119
实例 49 绘一个字符串	119
实例 50 画一个钟表.....	121
实例 51 绘制用纹理填充的线条	125
实例 52 不透明和半透明的线条	127
实例 53 缩略图.....	128
实例 54 用颜色渐变填充形状	133
实例 55 变换图像颜色	135
实例 56 绘制饼状图.....	137
实例 57 绘制柱状图.....	139
实例 58 折线图.....	142
实例 59 五角星.....	144
实例 60 在图像上绘制图标	146

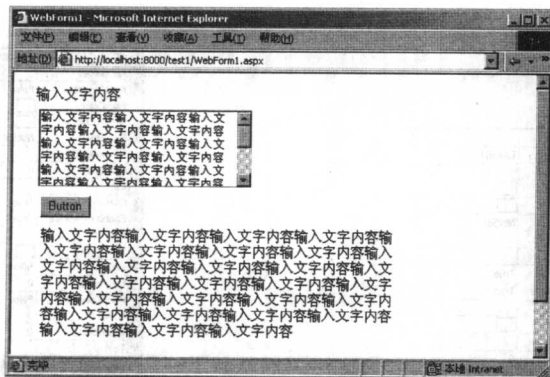
第 6 章 ADO.NET 及相关控件	149
实例 61 使用 DataAdapter 填充 DataSet	149
实例 62 使用 Repeater 显示数据	153
实例 63 使用 DataList 显示数据	156
实例 64 使用 DataGrid 显示数据	162
实例 65 分页显示	164
实例 66 DataGrid 控件中的数据排序	167
实例 67 使用 DataView 自定义显示的数据	169
实例 68 向 DataTable 中增加数据	172
实例 69 更新 DataTable 中的数据	175
实例 70 从 DataTable 中删除数据	178
实例 71 从 SQL 中以 XML 格式返回数据	181
实例 72 问卷调查	183
第 7 章 XML 应用	187
实例 73 读 XML 文件	187
实例 74 写 XML 文档	190
实例 75 使用 XSL 转换 XML 文件	193
实例 76 创建新属性	195
实例 77 插入子节点	197
实例 78 检索 XML 节点	199
实例 79 删除 XML 节点	202
实例 80 用户注册	205
实例 81 用户登录	208
实例 82 更改密码	211
实例 83 把 DataSet 数据转换成 XML 文件	214
实例 84 用二维表格显示 XML 文件	217
第 8 章 Web Service	221
实例 85 创建 Web Service	221
实例 86 Web Service 计算器	225
实例 87 返回 XML 文件	227
实例 88 返回 DataSet	229
实例 89 Web Service 中的状态管理	231
第 9 章 移动开发	233
实例 90 移动日历	234
实例 91 在移动设备上显示文本	237

实例 92	在移动设备上显示广告条	238
实例 93	在两个 Form 间切换	241
实例 94	PhoneCall 控件	242
实例 95	将数据库内容显示在手机上	243
实例 96	从移动终端调用 Web Service	245
第 10 章	论坛和留言板	248
实例 97	论坛登录和注册	248
实例 98	显示和增加论坛标题	257
实例 99	浏览和回复文章	263
实例 100	用户留言	273

第1章 基本应用

本章包括 20 个实例：文字显示、动态生成随机数、页面跳转、发送邮件、检索浏览器信息、检索服务器环境变量、使用程序生成页面内容、动态改变文字显示样式、IP 地址验证、IP 地址和主机名相互查找、显示 Access 数据库的数据、显示 SQL Server 数据库的数据、创建 Web 用户控件、调用 COM 组件、调用 C#类库、调用 WebService、捕捉异常、区域性演示、页面跟踪和应用程序级跟踪。本章通过这 20 个基础的实例，演示了如何在 ASP.NET 中执行在 Web 开发中常见的任务。通过本章实例的学习，读者可以迅速对 ASP.NET 的开发过程有一个基本的认识，为后续几章的学习打好基础。

实例1 文字显示



【实例说明】

本实例演示如何将输入的值显示出来。文字显示是基础中的基础，所有 Web 应用都离不开信息的输入和显示。通过本实例，读者将学习到如何通过属性赋值的方式显示用户输入的数据。

【开发要点】

Web 控件中的 Label 控件主要用来显示信息。本实例即通过给 Label 控件的 Text 属性赋值来达到显示信息的目的。

【步骤】

(1) 启动 Visual Studio.NET，选择“新建项目”命令，在“新建项目”对话框中选择“Visual Basic 项目”，在“模板”列表框中选择“ASP.NET Web 应用程序”。默认的项目名称是 Web

Application1, 可根据需要改成实际项目的名称, 然后单击“确定”按钮, 如图 1-1 所示。

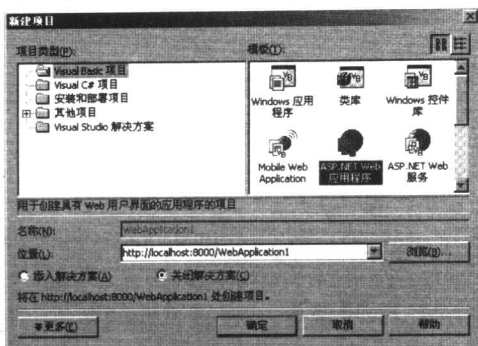


图 1-1 创建新项目

(2) 从左边“工具箱”中的“Web 窗体”中选择两个 Label 控件、一个 TextBox 控件和一个 Button 控件加入到页面中。

(3) 单击选中 Label1 控件, 在右面的“属性”面板中设置 Label1 控件的 Text 属性值为“输入文字内容”, 如图 1-2 所示。

(4) 单击选中 TextBox1, 拖动 TextBox1 控件周围的小方块以调整其大小, 再切换到右边的“属性”面板, 设置 TextBox1 的 Text 属性值为“(空串)”, TextMode 属性值为 MultiLine, 如图 1-3 所示。

(ID)	Label1
AccessKey	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
Enabled	True
EnableViewState	True
Font	
ForeColor	
Height	
TabIndex	0
Text	输入文字内容
ToolTip	
Visible	True
Width	

图 1-2 设置 Label 1 控件属性

(ID)	TextBox1
AccessKey	
AutoPostBack	False
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
Columns	0
CssClass	
Enabled	True
EnableViewState	True
Font	
ForeColor	
Height	90px
MaxLength	0
ReadOnly	False
Rows	0
TabIndex	0
Text	
TextMode	MultiLine
ToolTip	
Visible	True
Width	244px
Wrap	True

图 1-3 设置 TextBox 1 属性

(5) 单击选中 Button1, 切换到右边的“属性”面板, 设置其 Text 属性为“显示文字内容”。设置完成后的页面如图 1-4 所示。

(6) 添加程序代码。

- 双击 Button1, 此时 Visual Studio.NET 会自动创建一个 CodeBehind 的 Visual Basic 文件, 名称为 WebForm1.aspx.vb。如果更改了 Form 名称, 则此文件将会相应地改变成“Form 名称.aspx.vb”。
- 向 WebForm1.aspx.vb 文件中的 Button1_Click 方法中添加如下代码。

```
Label2.Text = TextBox1.Text
```

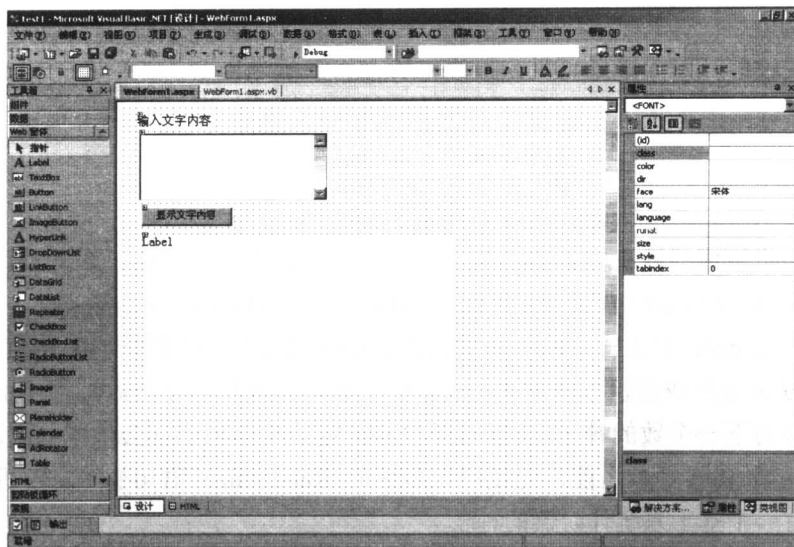


图 1-4 添加控件

然后保存文件。

(7) 按快捷键 F5 或者单击工具栏上的  按钮, Visual Studio.NET 会自动编译程序并运行。

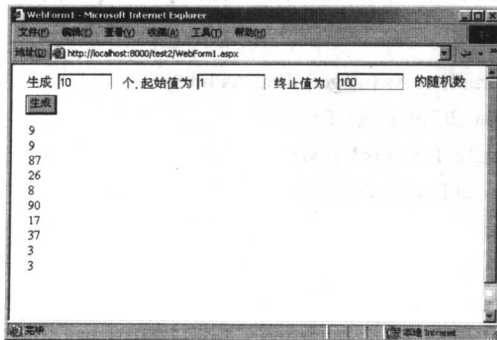
【举一反三】

本实例利用几个 Web 控件演示了如何通过属性赋值来显示输入的信息, 学完本实例后, 可以自行写出获取或显示其他 Web 控件值的方法。

【应用】

本实例全部通过 Web 控件来做演示, 实际应用中还有其他一些方法来实现, 比如利用 HTML 控件, 或者直接在页面中使用传统的 `<%=Request.Form("control")%>` 绑定方法显示 Form 的值。具体应用时要根据实际需要来决定采用哪种方法。

实例 2 动态生成随机数



【实例说明】

随机数在程序开发中随处可见，很多地方都会遇到随机数，比如抽奖程序、猜数字游戏、在线答题等。本实例演示如何使用 ASP.NET 来生成指定数量、指定范围的随机数。

【开发要点】

本例主要利用公式 $\text{Int}((\text{upperbound} - \text{lowerbound} + 1) \times \text{Rnd} + \text{lowerbound})$ 来生成随机数，lowerbound 是指生成的随机数的最小值，upperbound 是指生成的随机数的最大值。Rnd 函数是 Visual Basic.NET 的内置函数，语法是：Rnd([number])。其中 number 是一个 Single 值或任何有效的 Single 表达式。Rnd 函数将返回小于 1 但大于或等于 0 的值，number 的值将决定 Rnd 函数生成随机数的方式。由于每一次后续调用 Rnd 函数时都使用序列中前一个生成的数作为下一个数的种子，所以对于任何给定的初始种子都会生成相同的数字序列。因此应该在调用 Rnd 函数之前，先使用无参数的 Randomize 语句初始化随机数生成器，该生成器具有一个基于系统计时器的种子。

【步骤】

(1) 启动 Visual Studio.NET，选择“新建项目”命令，在“新建项目”对话框中选择“Visual Basic 项目”，在“模板”列表框中选择“ASP.NET Web 应用程序”。默认的项目名称是 WebApplication1，根据需要改成实际项目的名称，然后单击“确定”按钮，参见图 1-1 所示。

(2) 从左边“工具箱”中的“Web 窗体”中选择 5 个 Label 控件、3 个 TextBox 控件和 1 个 Button 控件加入到页面中。

(3) 设置 Label1 的 Text 属性为“生成”，设置 Label2 的 Text 属性为“个，起始值为”，设置 Label3 的 Text 属性为“终止值为”，设置 Label4 的 Text 属性为“的随机数”，设置 TextBox1 的 Text 属性为 10，设置 TextBox2 的 Text 属性为 1，设置 TextBox3 的 Text 属性为 100，设置 Button1 的 Text 属性为“生成”。设置完成后的页面如图 1-5 所示。

(4) 添加生成随机数的程序代码。

- 双击 Button1，此时 Visual Studio.NET 会自动创建一个 CodeBehind 的 Visual Basic 文件，名称为 WebForm1.aspx.vb。
- 向 WebForm1.aspx.vb 文件的 Button1_Click 方法中添加如下代码。

```
Dim i As Integer
Dim iTotal As Integer, iStart As Integer, iEnd As Integer
iTotal = Convert.ToInt32(TextBox1.Text)
iStart = Convert.ToInt32(TextBox2.Text)
iEnd = Convert.ToInt32(TextBox3.Text)
Label5.Text = ""
For i = 1 To iTotal
    Randomize()
    Label5.Text = Label5.Text & CInt(Int(((iEnd - iStart + 1) * Rnd(Timer)) + iStart)) & "<BR>"
Next
```

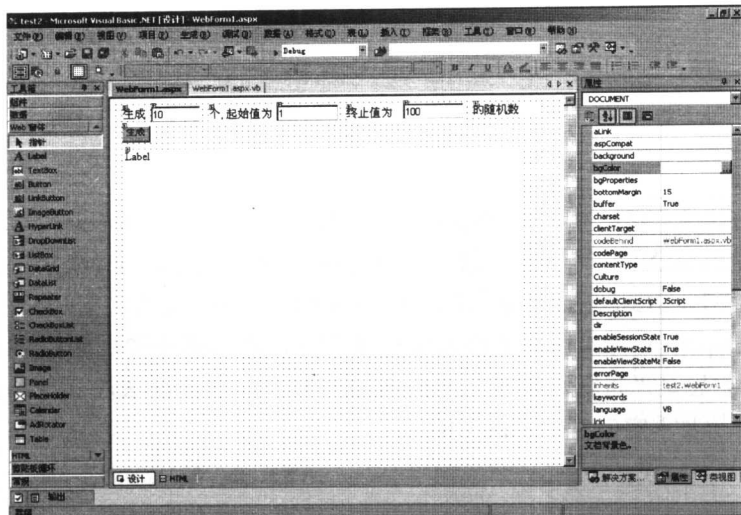


图 1-5 添加控件

然后保存文件。

提示:

要特别注意不要忘记使用 `Randomize()` 语句, 否则每次生成的随机数将会是相同的。

(5) 按快捷键 F5 或者单击工具栏上的  按钮, Visual Studio.NET 会自动编译程序并运行。

【举一反三】

学会本实例以后, 稍做修改就可以将其应用到各种需要随机数的 ASP.NET 应用程序中。

【应用】

实际应用中, 还会遇到各种需求, 比如要求随机数不能相同, 或者要求生成的随机数要排好序等。要完成这些功能, 只需增加或修改本例的相应程序即可实现。

实例3 页面跳转



【实例说明】

在 Web 应用开发中有很多地方会用到跳转功能，比如用户验证时、控制程序流程时、做友情链接时等。本例即通过一个简单的示例演示在 ASP.NET 中如何实现这种效果。

【开发要点】

本例使用与传统 ASP 相同的 Response 对象的内置方法 Redirect 来完成跳转功能。Response 对象是 ASP.NET 的内置对象，熟悉传统 ASP 的读者应该对这个内置对象非常熟悉。在 ASP.NET 中，Response 对象实际上就是 HttpResponse 类的实例，只不过用户可以直接使用罢了。Response 对象包含了很多属性和方法，本实例主要用到了 Redirect 方法。Redirect 方法的意思是“将客户端重定向到新的 URL”，它的语法是：Redirect(String)，其中参数 String 就是要重定向到的新的 URL。

【步骤】

(1) 启动 Visual Studio.NET，选择“新建项目”命令，在“新建项目”对话框中选择“Visual Basic 项目”，在“模板”对话框中选择“ASP.NET Web 应用程序”。默认的项目名称是 WebApplication1，根据需要改成实际项目的名称，然后单击“确定”按钮，参见图 1-1 所示。

(2) 从左边“工具箱”中的“Web 窗体”中选择一个 Label 控件、一个 DropDownList 控件和一个 Button 控件加入到页面中。

(3) 设置 Label1 的 Text 值为“选择链接”，设置 Button1 的 Text 值为“访问选中的网站”，然后向 DropDownList1 添加一些网址：首先单击选中该控件，切换到“属性”面板，单击 Items 选项，再单击右边的  按钮，此时 Visual Studio.NET 将会弹出一个对话框(如图 1-6 所示)用来添加 DropDownList1 里的各个值，单击“添加”按钮，然后在对话框右面的属性列表框里设置 Text 属性和 Value 属性为相应的网址名称及 URL。完成设置后的页面效果如图 1-7 所示。

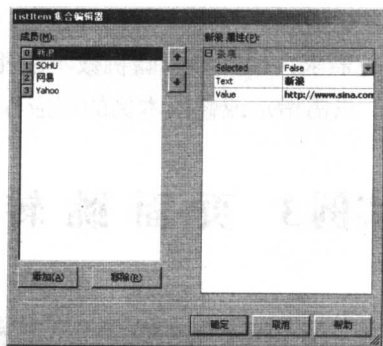


图 1-6 向 DropDownList 控件添加值

(4) 添加程序代码。

- 双击 Button1，此时 Visual Studio.NET 会自动创建一个 CodeBehind 的 Visual Basic 文件，名称为 WebForm1.aspx.vb。
- 向 WebForm1.aspx.vb 文件中的 Button1_Click 方法中添加如下代码。

```
Response.Redirect(Me.DropDownList1.SelectedItem.Value)
```

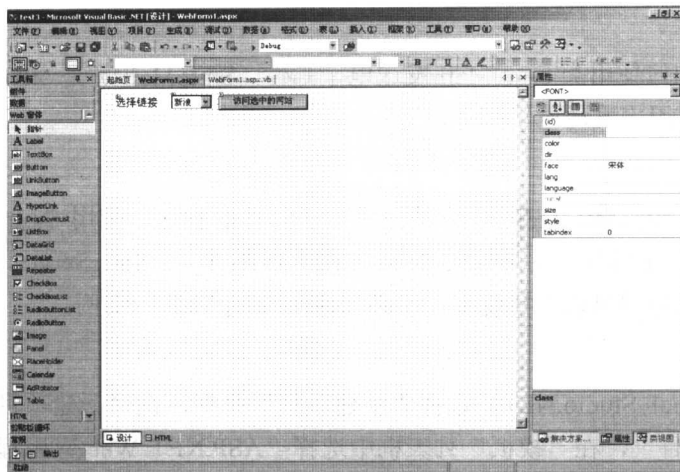


图 1-7 添加控件

然后保存文件。

(5) 按快捷键 F5 或者单击工具栏上的  按钮, Visual Studio.NET 会自动编译程序并运行。

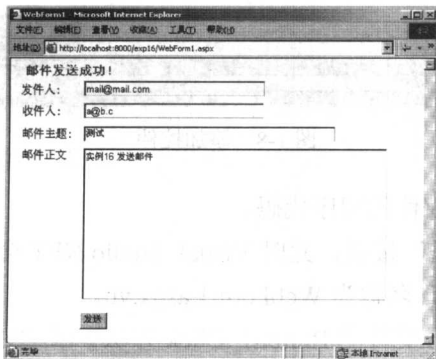
【举一反三】

学会本实例以后, 把 Redirect 方法的参数修改为实际的 URL, 即可以应用在很多需要跳转功能的地方。

【应用】

ASP.NET 中还有一个内置的 Server 对象, 它的 Transfer 方法也可以达到跳转的目的。实际应用中, 有的时候会遇到使用 Server.Transfer 的情况, 它与 Response.Redirect 是不同的两个方法: Server.Transfer 属于服务器端跳转, Response.Redirect 属于客户端跳转, 从另一个角度可以解释成 Server.Transfer 是一次请求执行跳转, 而 Response.Redirect 是通过两次请求实现的跳转。读者应特别注意两个方法之间的区别, 并且在应用中选择适当的方法。

实例 4 发送邮件



【实例说明】

本实例演示如何使用类 `SmtpMail` 发送电子邮件。`SmtpMail` 类是 .NET Framework 类库中提供的类，主要用来发送电子邮件。

【开发要点】

本例使用类 `SmtpMail` 的 `Send` 方法完成电子邮件的发送。使用 `SmtpMail` 类时，只要正确设置 `SmtpServer` 属性，就可以用 `Send` 方法发送邮件了。`SmtpServer` 属性和 `Send` 方法都是静态的，也就是说可以直接使用，不必创建类 `SmtpMail` 的实例。

【步骤】

(1) 启动 Visual Studio.NET，选择“新建项目”命令，在“新建项目”对话框中选择“Visual Basic 项目”，在“模板”列表框中选择“ASP.NET Web 应用程序”。默认的项目名称是 `WebApplication1`，根据需要改成实际项目的名称，然后单击“确定”按钮，参见图 1-1 所示。

(2) 从左边“工具箱”中的“Web 窗体”中选择 5 个 `Label` 控件、4 个 `TextBox` 控件和一个按钮控件加入到页面中。分别设置前 4 个 `Label` 控件的 `Text` 属性为“发件人”、“收件人”、“邮件主题”、“邮件正文”，第 5 个 `Label` 控件设其 `Text` 属性为空、前景色为红色、字体为 `Large`，这个控件将用在正确发送时显示信息，所以初始时其 `Text` 属性应为空。4 个 `TextBox` 分别命名为 `txtFrom`、`txtTo`、`txtSubject`、`txtMailBody`，按钮控件的 `Text` 属性设为“发送”，如图 1-8 所示。

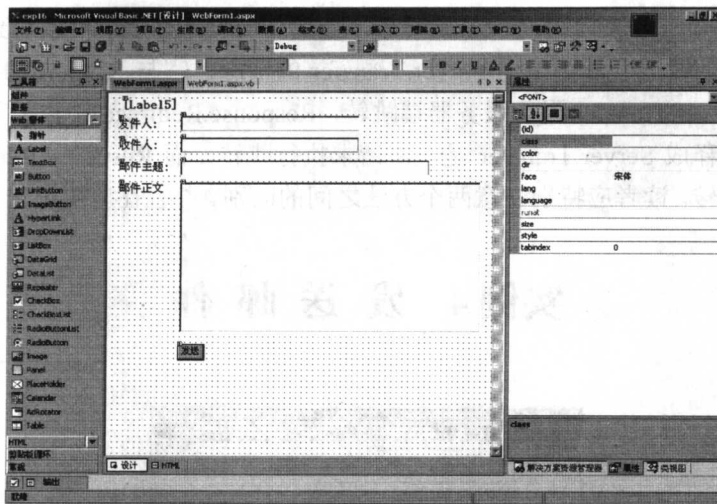


图 1-8 添加控件

(3) 添加处理发送电子邮件的程序代码。

- 双击页面上的“发送”按钮，此时 Visual Studio.NET 会自动创建一个 CodeBehind 的 Visual Basic 文件，名称为 `WebForm1.aspx.vb`。
- 向 `WebForm1.aspx.vb` 文件的 `Button1_Click` 方法中添加如下代码。