

 万水计算机技术实用大全系列

Visual C # .NET

精彩编程实例集锦

罗斌 等编著



包含所有实例的源代码



中国水利水电出版社
www.waterpub.com.cn

万水计算机技术实用大全系列

Visual C#.NET 精彩编程实例集锦

罗斌 等编著

中国水利水电出版社

内 容 提 要

本书用 150 个独立的实例程序介绍了 Microsoft Visual C#.NET 重点编程技术。书中所有实例程序摒弃语法说教, 侧重实际应用, 主要内容包括: 使用 GDI+ 技术开发图形文字特效程序技巧、使用 ADO.NET 编写 Access 和 SQL Server 数据库程序技巧、使用 ASP.NET 开发 Web 应用程序技巧、使用 Crystal Report 水晶报表向导开发报表程序技巧以及 Web Service 程序设计技巧。本书还介绍了文件加解密、文件切分组合、XML 转换等大量的系统和文件管理及界面设计的实用技术。本书所有实例在配套光盘中均有源代码, 让您所见即所得、所学即所用。

本书适合于使用 Microsoft Visual C#.NET 进行程序开发的初、中级用户参考, 也可作为高等院校及计算机培训学校在校学生的学习案例。

图书在版编目 (CIP) 数据

Visual C#.NET 精彩编程实例集锦 / 罗斌等编著. —北京: 中国水利水电出版社, 2005

(万水计算机技术实用大全系列)

ISBN 7-5084-3031-X

I. V… II. 罗… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 057073 号

书 名	Visual C#.NET 精彩编程实例集锦
作 者	罗斌 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 32 印张 726 千字
版 次	2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷
印 数	0001—5000 册
定 价	58.00 元 (含 1CD)

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

Microsoft.NET 平台的推出，掀起了计算机领域的一场革命；经过比尔·盖茨先生几年的摇旗呐喊，如今，.NET 的概念已渐渐深入人心；与之同生共长的 Visual C#语言也逐渐成为有志于在未来的计算机领域有所建树的编程人员追求的一种明星语言。

在 Microsoft Visual Studio.NET 开发工具中，微软提供了 Visual Basic、Visual C#、Visual J# 和 Visual C++四种编程语言，毫无疑问，Visual C#语言是.NET 技术的核心开发语言，是.NET 技术最忠实的实践者和实现者。Visual C#语言具有 C++语言的强大功能、Java 语言的跨平台特性和 Delphi 语言的方便快捷等众多的优秀品质。最引人注目的是，它是出自于 Borland 公司的前首席软件架构师——安德森·希斯伯格之手，回忆这位软件大师是曾经怎样让 Pascal 摇身一变成为 Delphi，使 Borland 公司起死回生，再移师微软使 Visual J++从 1.0 一跃跳到 6.0，搞得 Sun 措手不及，最后神话般地推出 Visual C#语言的历程，就没有理由怀疑 Visual C#语言的未来。

Visual C#语言是一种简单、现代、面向对象和类型安全的编程语言，它实现了快速应用程序开发、跨平台部署，能够访问平台固有的资源，支持 COM 和 .NET 技术，提供了 Java 风格的垃圾收集，Visual C#语言的一个重要特点就是与 Web 紧密结合，展示了 Runtime 和 .NET 类库的高性能。在.NET 的技术体系中，可以实现：在 Visual C#.NET 中创建一个基类，再在 Visual C++.NET 中创建一个继承这个基类特性的派生类，然后在 Visual Basic.NET 中创建这个派生类的一个应用实例，这样的跨语言一条龙生产作业。需要说明的是，Visual C#.NET 并不生成可直接执行的 X86 机器码，而是生成 MSIL，也就是.NET 平台的中间语言，它由 .NET 定义，并由 JIT（实时编译器）编译解释执行；也就是说，Visual C#.NET 生成的可执行程序并不是真正的可直接执行的程序，而是一种被别人解释执行的程序，因此它在执行时的响应速度比 Visual C++.NET 生成的可执行程序慢。不过，这种差别不是特别明显，其慢的速度现在我们基本可以忽略。

本书不是一本细数 Visual C#语言的特点和概念的编程指导书。用鲜活的实例诠释 Visual C#.NET 的精华，释放 .NET 的威力是本书编写时追求的目标。全书由简到繁、从易到难，共分为控件操作、文件管理、程序设置、系统维护、图像处理、数据库开发、网络应用和 Web 应用八个部分共 150 个编程实例，每一个编程实例都给出了简要的编写说明、具体的实现步骤和完整的源代码（见光盘），每一个编程实例都至少介绍了一种方法，解决了一个问题，这些方法和问题可能与语言有关、可能与系统有关、可能与应用有关，总之，它们都是您在使用 Visual C#.NET 解决实际问题时可能遇到的问题。

本书适用于具有 Visual C#语言基础知识的初、中级读者学习，书中的所有内容和思想并非一人之力所能及，而是凝聚了众多 Visual C#.NET 编程高手多年的心得并经过充分的提炼和总结而成，在此对他们的智慧表示崇高的敬意和衷心的感谢！参加本书编写的同志

包括罗兴禄、罗斌、罗帅、雷治英和罗顺文，终稿由罗斌统筹完成。书中的所有实例程序均在 Microsoft Visual Studio.NET/Microsoft Office 2003 /Microsoft Windows XP 环境中编译执行成功，因此，请在以上环境中进行实例程序的学习，当然，由于作者水平有限，其中也可能存在一些疏漏和不当之处，敬请读者批评指正，电子邮箱 binluobin@163.cm。

罗斌

2005年于重庆长寿

目 录

前言

第 1 章 控件操作	1
实例 1 如何使用错误提醒控件	1
实例 2 如何使用信息提示控件	4
实例 3 如何使用菜单控件	5
实例 4 如何使用工具栏控件	7
实例 5 如何使用状态栏控件	13
实例 6 如何使用托盘控件	17
实例 7 如何使用标签页控件	20
实例 8 如何使用进度条控件	22
实例 9 如何使用树视图控件	24
实例 10 如何使用列表视图控件	30
实例 11 如何使用多选列表框控件	35
实例 12 如何使用图像控件	37
实例 13 如何使用窗体分隔控件	38
第 2 章 文件管理	42
实例 14 如何获取程序文件信息	42
实例 15 如何获取指定文件的根信息	46
实例 16 如何获取当前程序所在的文件夹	48
实例 17 如何获取指定文件夹下的所有文件	50
实例 18 如何获取指定文件夹下的所有文件夹	52
实例 19 如何获取和设置文件属性	54
实例 20 如何获取和设置文件夹属性	60
实例 21 如何判断文件或文件夹是否存在	64
实例 22 如何创建和删除多层文件夹	67
实例 23 如何监视文件和文件夹系统变化情况	69
实例 24 如何加解密文件	72
实例 25 如何切分组合文件	78
实例 26 如何以流式方式读写文本文件	82
实例 27 如何直接打印文件	86
实例 28 如何操作帮助文件	90
实例 29 如何操作 Word 文件	93

实例 30	如何操作 Excel 文件	96
实例 31	如何播放 MP3 文件	99
实例 32	如何播放 VCD 文件	104
实例 33	如何播放 DVD 文件	108
实例 34	如何播放 Flash 文件	110
实例 35	如何播放 RealPlayer 文件	114
第 3 章	程序设置	118
实例 36	如何创建椭圆形窗体	118
实例 37	如何创建文字形窗体	120
实例 38	如何创建不规则窗体	122
实例 39	如何创建移动的窗体	123
实例 40	如何创建无标题栏窗体	125
实例 41	如何拖动无标题栏窗体	126
实例 42	如何制作自绘式菜单	128
实例 43	如何制作图形组合框	132
实例 44	如何设置窗体的透明度	135
实例 45	如何设置窗体启动位置	137
实例 46	如何设置窗体最前端显示	139
实例 47	如何实现精灵提示	141
实例 48	如何实现打印预览	144
实例 49	如何实现剪贴板循环	150
实例 50	如何实现菜单历史列表	154
实例 51	如何实现程序互斥运行	159
实例 52	如何屏蔽鼠标消息	161
实例 53	如何模拟键盘输入	162
实例 54	如何使用消息提示框	166
实例 55	如何使用系统 API 函数	170
实例 56	如何获取和设置输入法	173
实例 57	如何在程序中启动其他程序	176
实例 58	如何在程序中关闭其他程序	179
实例 59	如何在开机时自动启动程序	183
实例 60	如何启动电话拨号程序	186
实例 61	如何启动屏幕保护程序	188
实例 62	如何启动系统控制面板程序	190
实例 63	如何编写多线程程序	191
实例 64	如何编写 DLL 服务端程序	195
实例 65	如何编写 DLL 客户端程序	197

实例 66	如何编写用户控件程序	199
实例 67	如何给程序打包	201
第 4 章	系统维护	206
实例 68	如何获取系统基本信息	206
实例 69	如何获取系统服务信息	211
实例 70	如何获取系统进程信息	215
实例 71	如何获取系统日期时间	219
实例 72	如何获取系统所有驱动器	223
实例 73	如何获取系统驱动器信息	224
实例 74	如何获取系统计数器信息	227
实例 75	如何获取操作系统版本信息	229
实例 76	如何获取系统特殊文件夹信息	231
实例 77	如何获取系统文化和国家信息	234
实例 78	如何获取系统已经安装的字体	237
实例 79	如何获取系统安装日期和启动时间	242
实例 80	如何获取 CPU 信息	244
实例 81	如何浏览注册表信息	247
实例 82	如何隐藏系统驱动器	251
实例 83	如何清理系统回收站	256
实例 84	如何制作日志查看器	263
实例 85	如何禁止关闭操作系统	268
实例 86	如何禁止屏幕保护程序	270
实例 87	如何禁止系统菜单按钮命令	272
第 5 章	图像处理	274
实例 88	如何实现大图像浏览	274
实例 89	如何实现多图像浏览	275
实例 90	如何以反色方式显示图像	280
实例 91	如何以浮雕方式显示图像	282
实例 92	如何以黑白方式显示图像	285
实例 93	如何以柔化方式显示图像	287
实例 94	如何以锐化方式显示图像	290
实例 95	如何以雾化方式显示图像	293
实例 96	如何以百叶窗方式显示图像	295
实例 97	如何以马赛克方式显示图像	298
实例 98	如何以非矩形方式显示图像	302
实例 99	如何以任意比例缩放显示图像	306
实例 100	如何以任意角度旋转显示图像	308

实例 101	如何实现动态显示图像	311
实例 102	如何以旋转方式显示文字	315
实例 103	如何在图像中添加文字	318
实例 104	如何以图像填充文字线条	320
实例 105	如何实现图形绘制拉伸效果	325
实例 106	如何控制图像的显示方式	328
实例 107	如何控制图像的显示质量	331
实例 108	如何绘制渐变色背景	334
实例 109	如何绘制颜色散射图形	338
实例 110	如何复制粘贴图像	341
实例 111	如何转换图像文件格式	345
实例 112	如何以任意比例转换图像大小	349
第 6 章	数据库开发	354
实例 113	如何读写 Access 数据库	354
实例 114	如何查询 Access 数据库	364
实例 115	如何创建 SQL Server 数据库	368
实例 116	如何读写 SQL Server 数据库	373
实例 117	如何更新 SQL Server 数据库	382
实例 118	如何读写 XML 文件	388
实例 119	如何读取 EXCEL 文件	394
实例 120	如何取得聚合函数返回值	396
实例 121	如何直接操作主从关系表	400
实例 122	如何以 Web 方式查询 Access 数据库	405
实例 123	如何使用列表视图显示数据库信息	409
实例 124	如何显示和修改数据库中的图像信息	413
实例 125	如何使用水晶报表向导创建报表	416
第 7 章	网络应用	422
实例 126	如何下载网页	422
实例 127	如何上下载文件	424
实例 128	如何自制浏览器	427
实例 129	如何检查邮件数量	430
实例 130	如何发送电子邮件	434
实例 131	如何实现点对点的聊天	438
实例 132	如何在线收听网络电台节目	441
实例 133	如何获取主机名和 IP 地址	443
实例 134	如何实现连续的 PING	445
实例 135	如何扫描端口	450

第 8 章 Web 应用	453
实例 136 如何在 Web 页中使用广告控件.....	453
实例 137 如何在 Web 页中使用表格控件.....	455
实例 138 如何在 Web 页中使用日历控件.....	462
实例 139 如何在 Web 页中使用数据网格控件.....	464
实例 140 如何在 Web 页中使用比较校验控件.....	471
实例 141 如何在 Web 页中使用单选列表控件.....	473
实例 142 如何在 Web 页中使用多选列表框控件.....	475
实例 143 如何在 Web 页中使用正则表达式控件.....	478
实例 144 如何在 Web 页中获取浏览器信息.....	480
实例 145 如何在 Web 页中浏览图像.....	484
实例 146 如何在 Web 页中绘制图形.....	486
实例 147 如何在 Web 页间传递数据.....	488
实例 148 如何创建 Web Services 程序.....	491
实例 149 如何打包 Web Services 程序.....	494
实例 150 如何访问 Web Services 程序.....	496

第 1 章 控件操作

实例 1 如何使用错误提醒控件

实例目的

此实例主要介绍了如何使用错误提醒控件（ErrorProvider）指示窗体上有关错误信息的编程技术。实例程序执行后，当用户两次密码的输入不一致时，会弹出一个错误提示框，并在密码输入控件的右边显示一个红色的警告图标，当鼠标停留在该图标时，会出现一个浮动框显示相关提示信息，效果如图 1-1 所示。

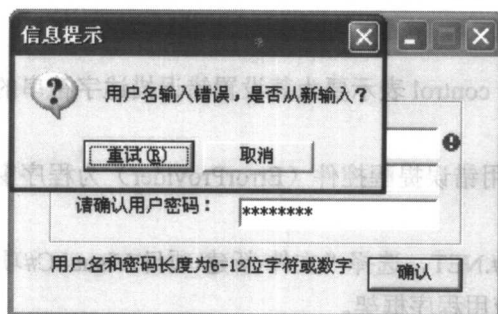


图 1-1

实现方法

Visual C#.NET 工具箱中的错误提醒控件（ErrorProvider）是能够指示窗体上有关错误的控件。它使用一个简单的界面，向最终用户指出窗体上的控件具有与它关联的错误。如果为控件指定了错误描述，控件旁边将会出现一个图标。此图标按照指定的方式、以指定的速率闪烁。当鼠标悬停在此图标上时，会出现显示错误描述字符串的工具提示。错误提醒控件（ErrorProvider）的常用属性包括：

- (1) BlinkRate 属性，该属性用于获取或设置错误图标的闪烁速率。
- (2) Icon 属性，该属性用于获取或设置错误图标，当为控件设置了错误描述字符串时，该图标显示在有错误的控件旁边。
- (3) BlinkStyle 属性，该属性用于获取或设置一个值，通过该值指示错误图标的闪烁时间。BlinkStyle 属性是一个 ErrorBlinkStyle 枚举类型，其成员说明如表 1-1 所示。

表 1-1

成员名称	说明
AlwaysBlink	当错误图标第一次显示时, 或者当为控件设置了错误描述字符串并且错误图标已经显示时, 总是闪烁
BlinkIfDifferentError	当图标已经显示并且为控件设置了新的错误字符串时闪烁
NeverBlink	错误图标从不闪烁

错误提醒控件 (ErrorProvider) 常用的公共方法包括:

- (1) SetError()方法, 该方法用于设置指定控件的错误描述字符串。
- (2) SetIconAlignment()方法, 该方法用于设置错误图标相对于控件的放置位置。
- (3) SetIconPadding()方法, 该方法用于设置指定控件和错误图标之间应保留的额外空间量。

错误提醒控件 (ErrorProvider) 最重要的公共方法是 SetError()方法, 该方法的声明如下:

```
public void SetError(
    Control control,
    string value
);
```

其中, 参数 Control control 表示要为其设置错误描述字符串的控件; 参数 string value 是错误描述字符串。

下面是实例程序使用错误提醒控件 (ErrorProvider) 为程序窗体中的控件添加错误信息提示的具体操作步骤。

(1) 启动 Visual C#.NET, 选择“文件\新建\项目\Visual C#项目\Windows 应用程序”, 创建一个 C# Windows 应用程序框架。

(2) 选择“视图\工具箱”菜单, 指向“工具箱”的“Windows 窗体”页, 在程序设计窗体上拖放一个 GroupBox 控件 (groupBox1), 在属性对话框中设置其 Text 属性为“用户注册信息”; 再拖放四个 Label 控件, 在属性对话框中设置其 Text 属性分别为“请输入用户名:”、“请输入用户密码:”、“请确认用户密码:”和“用户名和密码长度为 6-12 位字符或数字”; 再拖放三个 TextBox 控件, 在属性对话框中设置其中的两个密码 TextBox 控件的 PasswordChar 属性为“*”。调整其大小适应程序设计窗体的大小。

(3) 选择“视图\工具箱”菜单, 指向“工具箱”的“Windows 窗体”页, 在程序设计窗体上拖放一个 ErrorProvider 控件 (errorProvider1), 在属性对话框中设置其 BlinkRate 属性为 250、设置其 BlinkStyle 属性为 BlinkIfDifferentError。

(4) 选择“视图\工具箱”菜单, 指向“工具箱”的“Windows 窗体”页, 在程序设计窗体上拖放一个 Button 控件 (button1), 在属性对话框中设置其 Text 属性为“确认”, 然后为其鼠标单击事件 (Click) 添加如下所示的实现代码:

```
private void button1_Click(object sender, System.EventArgs e)
{ //检验用户数据输入结果
    if (this.textBox1.Text.Length > 12 || this.textBox1.Text.Length < 6)
```

```
{
    this.errorProvider1.SetError(this.textBox1, "用户名输入错误");
    DialogResult ReturnDlg=MessageBox.Show(this, "用户名输入错误, 是否从
新输入? ", "信息提示", MessageBoxButtons.RetryCancel, MessageBoxIcon.Question);
    switch (ReturnDlg)
    {
        case DialogResult.Retry:
            this.textBox1.Text="";
            break;
        case DialogResult.Cancel:
            break;
    }
}
if(this.textBox2.Text.Length>12||this.textBox2.Text.Length<6)
{
    this.errorProvider1.SetError(this.textBox2, "用户密码输入错误");
    DialogResult ReturnDlg=MessageBox.Show(this, "用户密码输入错误, 是否从新
输入? ", "信息提示", MessageBoxButtons.RetryCancel, MessageBoxIcon.Question);
    switch (ReturnDlg)
    {
        case DialogResult.Retry:
            this.textBox2.Text="";
            break;
        case DialogResult.Cancel:
            break;
    }
}
if(!(this.textBox2.Text==this.textBox3.Text))
{
    this.errorProvider1.SetError(this.textBox3, "用户密码两次输入不一致");
    DialogResult ReturnDlg=MessageBox.Show(this, "用户密码两次输入不一致, 是
否从新输入? ", "信息提示", MessageBoxButtons.RetryCancel, MessageBoxIcon.Question);
    switch (ReturnDlg)
    {
        case DialogResult.Retry:
            this.textBox3.Text="";
            break;
        case DialogResult.Cancel:
            break;
    }
}
}
```

最后, 编译执行程序即可。

实例源代码

有关程序项目文件全部源代码及执行程序,请参考配套光盘“源代码\第1章控件操作\实例1如何使用错误提醒控件”。

实例2 如何使用信息提示控件

实例目的

此实例主要介绍了如何使用信息提示控件(ToolTip)为程序窗体控件添加提示信息的编程技术。实例程序执行后,效果如图2-1所示。

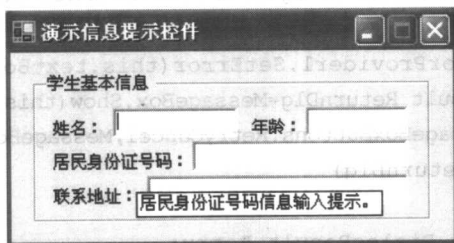


图 2-1

实现方法

Visual C#.NET 提供了大量的控件以实现应用程序的快速开发,这些控件充分体现了C#语言面向对象的特点,其中的大部分控件都具有一些共同属性,这些共同属性包括:①Name属性,表示控件的名称;②Enabled属性,表示控件是否可以对用户交互作出响应;③Font属性,表示控件中显示文字的字体格式,包括字体名称、字号以及是否以斜体、加粗和下划线显示;④BackColor属性,表示控件的背景色;⑤ForeColor属性,表示控件的前景色;⑥Image属性,表示控件上显示的图像;⑦Location属性,这是一个复合属性,表示控件的左上角相对于其容器的左上角的坐标;⑧Size属性,这是一个复合属性,表示控件的高度和宽度;⑨Visible属性,表示是否显示该控件。

如果在程序设计窗体中添加了信息提示控件(ToolTip),则程序设计窗体中的所有控件的属性对话框中将自动增加一个ToolTip杂项属性,用于为程序设计窗体中的控件添加提示信息。ToolTip控件是一个不可视控件,该控件的用途是当鼠标位于某个控件上并停留一段时间后,显示该控件功能的提示信息。其主要属性包括:

- (1) Active属性,该属性用于指示该控件当前是否处于激活状态。
- (2) AutomaticDelay属性,该属性用于设置经过多长时间显示提示信息,默认值为500ms。
- (3) AutoPopDelay属性,该属性用于设置鼠标指针停留多长时间后提示信息消失。

(4) ReshowDelay 属性, 该属性用于表示鼠标指针从一个控件移到另一个控件时, 后面控件的工具提示窗口出现前必须经过多长时间。

(5) ShowAlways 属性, 该属性用于指示信息提示窗口在其父控件不活动时是否显示。

下面是实例程序使用信息提示控件 (ToolTip) 为程序窗体中的控件添加提示信息的具体操作步骤。

(1) 启动 Visual C#.NET, 选择“文件\新建\项目\Visual C#项目\Windows 应用程序”, 创建一个 C# Windows 应用程序框架。

(2) 选择“视图\工具箱”菜单, 指向“工具箱”的“Windows 窗体”页, 在程序设计窗体上拖放一个 GroupBox 控件 (groupBox1), 在属性对话框中设置其 Text 属性为“学生基本信息”; 再拖放四个 Label 控件, 在属性对话框中设置其 Text 属性分别为“姓名:”、“年龄:”、“居民身份证号码:”和“联系地址:”; 再这些 Label 控件右边拖放四个 TextBox 控件。并调整这些控件大小适应程序设计窗体大小。

(3) 选择“视图\工具箱”菜单, 指向“工具箱”的“Windows 窗体”页, 在程序设计窗体上拖放一个 ToolTip 控件, 在属性对话框中设置其 Active 属性为 True、设置其 AutomaticDelay 属性为 500、设置其 AutoPopDelay 属性为 5000、设置其 ReshowDelay 属性为 100、设置其 ShowAlways 属性为 False。

(4) 用鼠标双击程序设计窗体 Form1, 为程序设计窗体的装载事件 (Load) 添加如下所示的实现代码:

```
private void Form1_Load(object sender, System.EventArgs e)
{
    this.toolTip1.SetToolTip(this.textBox1, "请输入学生姓名。");
    this.toolTip1.SetToolTip(this.textBox2, "请输入学生年龄。");
    this.toolTip1.SetToolTip(this.textBox3, "请输入身份证号码。");
    this.toolTip1.SetToolTip(this.textBox4, "请输入联系地址。");
    this.toolTip1.SetToolTip(this.label3, "居民身份证号码信息输入提示。");
    this.toolTip1.SetToolTip(this.label4, "联系地址信息输入提示。");
}
```

最后, 编译执行程序即可。

实例源代码

有关程序项目文件全部源代码及执行程序, 请参考配套光盘“源代码\第1章控件操作\实例2 如何使用信息提示控件”。

实例3 如何使用菜单控件

实例目的

此实例主要介绍了如何使用主菜单控件 (MainMenu) 在程序窗体中定制主菜单的编程技术。实例程序执行后, 效果如图 3-1 所示。

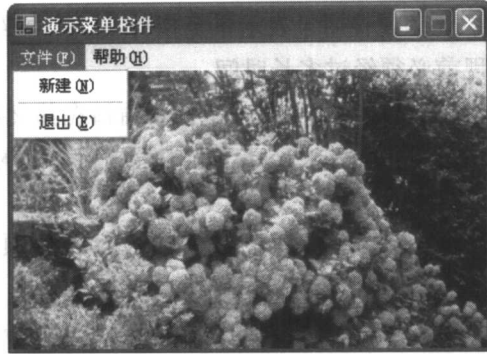


图 3-1

实现方法

在 Visual C#.NET 中, 创建应用程序窗体主菜单使用主菜单控件 (MainMenu)。利用主菜单控件 (MainMenu) 的菜单设计器可以非常快速地建立应用程序主菜单的结构。在设计主菜单的菜单项时, 可以用&符号代表快捷键 ALT, 如&F 代表 Alt+F; 用“-”代表分隔线。下面是实例程序使用 MainMenu 控件为应用程序创建主菜单的具体操作步骤。

(1) 启动 Visual C#.NET, 选择“文件\新建\项目\Visual C#项目\Windows 应用程序”, 创建一个 C# Windows 应用程序框架。

(2) 选择“视图\工具箱”菜单, 指向“工具箱”的“Windows 窗体”页, 在程序设计窗体上拖放一个 MainMenu 控件 (mainMenu1)。默认情况下, 该控件自动停留在程序设计窗体的上方, 并出现一个“请在此处输入”编辑框, 如图 3-2 所示。“请在此处输入”代表了主菜单中第一个菜单组的标题。单击“请在此处输入”编辑框, 然后输入“文件(&F)”, 此时在该文本框的下边和右边同时出现了另外两个“请在此处输入”编辑框。其中右边的“请在此处输入”编辑框代表的是下一个菜单组的标题, 如果不在这个文本编辑框中输入内容, 则不会有第二个菜单组; 下边的编辑框代表“文件”菜单组的第一个菜单项 (当然它也可能是一个菜单组), 如果不在其中输入内容, 则“文件”菜单组中不会有命令。请在“请在此处输入”位置处输入菜单项的名称, 如“文件”、“新建”、“退出”、“帮助”和“关于”等。如果需要移动或改变菜单项的位置, 直接拖动菜单项到目标位置即可。



图 3-2

(3) 在程序设计窗体中, 直接双击 **MainMenu** 控件 (**mainMenu1**) 中的“新建”、“退出”和“关于”菜单项, 为这些菜单项的鼠标单击事件 (**Click**) 添加如下所示的实现代码:

```
private void menuItem2_Click(object sender, System.EventArgs e)
{
    //响应“新建”菜单单击事件
    MessageBox.Show(this, "您选择了新建菜单", "信息提示", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
}
private void menuItem3_Click(object sender, System.EventArgs e)
{
    //响应“退出”菜单单击事件
    Application.Exit();
}
private void menuItem6_Click(object sender, System.EventArgs e)
{
    //响应“关于”菜单单击事件
    MessageBox.Show(sender.ToString());
}
```

最后, 编译执行程序即可。

说明: (1) 在菜单名称中, “&”符号后面的字母代表了该菜单命令的快捷键; 不过, 也可以通过菜单项的 **Shortcut** 属性设置菜单快捷键。菜单项的 **ShowShortcut** 属性决定是否隐藏所定义的菜单快捷键, 在默认状态下, 菜单快捷键是显示出来的;

(2) 如果需要在菜单中添加分隔栏, 只需要将菜单项的文字改为“-”符号即可。

(3) 为了指明一个菜单是否已经被选择, Windows 菜单系统支持将选择标记放在菜单标题文字旁边的功能, 菜单项的 **Checked** 属性指示命令是否已经被选定, 如果选定, 菜单标题文字旁边会打一个勾。

(4) **MenuItem** 类支持一种能够选择相互排斥的命令, 即组菜单; 当有了一组菜单命令但是每次只能选择一个时, 可以使用这一特性; 创建这种组菜单时, 首先要为每一个属于这个相互排斥的菜单组中的菜单项设置 **RadioChecked** 属性为 **true**, 接着设置 **Checked** 属性, 此时它看上去会是一个图形化的单选按钮, 而不是一个标记。

实例源代码

有关程序项目文件全部源代码及执行程序, 请参考配套光盘“源代码\第1章控件操作\实例3如何使用菜单控件”。

实例4 如何使用工具栏控件

实例目的

此实例主要介绍了如何使用工具栏控件 (**ToolBar**) 在程序窗体上制作带图标按钮的工具栏的编程技术。实例程序执行后, 工具栏上的按钮将同时显示图标和文字, 这些按钮分别可以执行新建、打开和保存文本文件命令, 并可以对文本文件的内容进行剪切、复制和粘贴操作, 效果如图 4-1 所示。