

Borland Delphi for .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

Borland Delphi for.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 200 个独立的实例程序介绍了 Borland Delphi for Microsoft.NET Framework 最新编程技术, 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 开发数据库程序技巧、使用 ASP.NET 开发 Web 应用程序技巧以及 Web Service 程序设计技巧等。在本书中, 还介绍了托管状态下的文件操作、系统管理、注册表应用、界面设计等难得一见的实用技术, 是 Delphi 程序员进军 Microsoft.NET Framework 的重要参考书。本书所有实例程序在配套光盘中均有源代码, 让您所见即所得、所学即所用。

本书适用于使用 Borland Delphi 2005 进行 Microsoft.NET Framework 程序开发的初、中级用户, 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

Borland Delphi for.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, 2006

(万水计算机技术实用大全系列)

ISBN 7-5084-3536-2

I. B… II. 罗… III. 软件工具—程序设计, IV.TP311.56

中国版本图书馆 CIP 数据核字 (2006) 第 001446 号

书 名	Borland Delphi for.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
经 售	电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 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	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 34 印张 839 千字
版 次	2006 年 1 月第 1 版 2006 年 1 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

为了适应 Internet 的快速发展,微软公司于 2000 年推出了 .NET 战略,以此作为下一代应用程序的开发平台。.NET 战略的推出,为微软的 Visual Studio 开发平台找到了一条新的出路,使之迅速占领了 .NET 开发的制高点。它同时也为在 Win32 战线上搞得红红火火的 Borland Delphi 的继续发展设置了巨大的障碍, Borland 的主力产品之一 C++ BuilderX 明确表示不支持 Microsoft.NET Framework, 作为 Borland 核心产品的 Delphi 又该如何面对呢? 大多数 Delphi 程序员均认为, 这是微软为封杀 Delphi 使出的杀手锏, 是微软 Internet Explorer 颠覆 Navigator 浏览器的故伎重演。

Borland C# Builder 的推出, 使许多处于十字路口的 Delphi 程序员看到了一丝希望的曙光, 他们都急切地盼望着 Delphi 能够绝处逢生, 支持 Microsoft.NET Framework。因为谁都知道, Microsoft.NET Framework 犹如现在的 Windows API, 一旦微软在以后的 Windows 操作系统中以 Microsoft.NET Framework 的形式提供 API, 就无疑会对现在的 Delphi 程序员提出巨大的挑战; 并且 Microsoft.NET Framework 在目前的 B/S 程序开发中的确表现得相当出色。Delphi 2005 的出现, 使人们再度发现 Borland 在语言开发平台方面的非凡实力。Delphi 2005 完美地将现有的 Delphi 技巧、知识及投资和 Microsoft.NET Framework 结合起来, 实现了与 Visual C++.NET 类似的功能: 既支持 Win32 应用程序的开发, 又支持 Microsoft.NET Framework 应用程序的开发, 并且在 ASP.NET 程序开发方面与 Visual Basic.NET 和 Visual C#.NET 语言相比起来也毫不逊色。

本书用立竿见影的实例程序全面展示使用 Borland Delphi 2005 开发各种 Microsoft.NET Framework 应用程序的最新魅力。全书由简到繁、从易到难, 分为程序界面编程实例、程序控制编程实例、文件操作编程实例、系统管理编程实例、GDI+特效编程实例、数据库开发编程实例和 ASP.NET 应用编程实例共七个部分 200 个独立编程实例, 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一种方法、解决一个问题, 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, 它们都可能是您在使用 Borland Delphi 2005 开发 Microsoft.NET Framework 应用程序时遇到的问题。

本书适用于使用 Borland Delphi 2005 进行 Microsoft.NET Framework 程序开发的初、中级用户, 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。书中的所有实例程序均在 Borland Delphi 2005 for the Microsoft.NET Framework/Microsoft Office 2003 /Microsoft Windows XP/ Microsoft Visual Studio.NET 2003 环境中编译执行成功。

书中所有内容和思想并非一人之力所能及, 而是凝聚了众多编程高手多年的心得并经过充分地提炼和总结而成, 在此对他们的智慧表示崇高的敬意和衷心的感谢! 本书编写人

员包括罗斌、罗兴禄、雷治英、罗帅、雷玲、曹勇、雷丽、李平、雷平、邓承会、雷国忠、王年素、雷浩、李佳星、范刚强、杨琴、罗顺文、吴多、王彬、张志红、罗兰、刘显国和刘婷，终稿由罗斌统筹完成。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。联系邮箱 binluobin@163.com 或 binluobin@sina.com。

罗斌

2005年11月于重庆长寿

目 录

前言

第 1 章 程序界面编程实例	1	实例 32 如何设置文件对话框	78
实例 1 如何创建换肤窗体	1	实例 33 如何设置消息提示框	81
实例 2 如何创建托盘窗体	3	实例 34 如何设置应用程序光标	84
实例 3 如何创建渐变色窗体	6	第 2 章 程序控制编程实例	88
实例 4 如何创建标签页窗体	8	实例 35 如何模拟键盘输入	88
实例 5 如何创建文字形窗体	10	实例 36 如何收听网络电台	91
实例 6 如何创建全屏窗体	13	实例 37 如何读取网页内容	93
实例 7 如何创建多文档窗体	14	实例 38 如何自制网页浏览器	95
实例 8 如何创建闪屏启动窗体	19	实例 39 如何实现点对点聊天	99
实例 9 如何创建渐隐渐显窗体	21	实例 40 如何查找替换字符串	102
实例 10 如何最小化最大化窗体	23	实例 41 如何拦截 Esc 按键操作	108
实例 11 如何最顶层显示窗体	24	实例 42 如何填充列表框数据	110
实例 12 如何创建关于对话框	26	实例 43 如何使用正则表达式	111
实例 13 如何在消息框中绘制表格	28	实例 44 如何计算两次时间之差	114
实例 14 如何在窗体上实现超链接	29	实例 45 如何根据年份查询生肖	116
实例 15 如何同步缩放窗体控件	31	实例 46 如何实现程序互斥运行	118
实例 16 如何拖动无标题栏窗体	34	实例 47 如何设置开机自启动程序	119
实例 17 如何显示控件提示信息	36	实例 48 如何设置文字输入法	122
实例 18 如何禁止窗体边框缩放	40	实例 49 如何设置程序功能键	124
实例 19 如何设置窗体背景图像	41	实例 50 如何设置打印页面	131
实例 20 如何设置窗体启动位置	42	实例 51 如何设置打印机	135
实例 21 如何创建历史记录菜单	43	实例 52 如何实现打印预览	139
实例 22 如何创建上下文菜单	50	实例 53 如何直接打印图像	142
实例 23 如何创建中英文菜单	53	实例 54 如何直接打印文本	145
实例 24 如何增加删除菜单	56	实例 55 如何启动电话拨号程序	147
实例 25 如何合并菜单组	58	实例 56 如何启动关闭执行程序	149
实例 26 如何设置菜单状态	60	实例 57 如何使用无窗体定时器	152
实例 27 如何设置菜单功能键	62	实例 58 如何使用有窗体定时器	154
实例 28 如何设置工具栏菜单	66	实例 59 如何获取程序命令行参数	156
实例 29 如何设置工具栏位置	70	实例 60 如何获取程序运行信息	157
实例 30 如何设置窗体状态栏	72	实例 61 如何获取进程模块信息	160
实例 31 如何设置窗体标题栏	75	实例 62 如何获取进程线程信息	163

实例 63	如何创建装配程序	169	第 4 章 系统管理编程实例	264	
实例 64	如何使用装配程序	172	实例 99	如何隐藏显示任务栏	264
实例 65	如何创建带参数多线程程序.....	174	实例 100	如何禁止在任务栏显示.....	266
实例 66	如何创建无参数多线程程序.....	177	实例 101	如何禁止任务栏右键菜单.....	267
实例 67	如何创建 WebService 服务程序 ...	179	实例 102	如何隐藏显示开始按钮.....	269
实例 68	如何创建 WebService 客户程序 ...	183	实例 103	如何新增开始右键菜单.....	270
第 3 章 文件操作编程实例	186	实例 104	如何设置系统桌面墙纸.....	272	
实例 69	如何操作帮助文件	186	实例 105	如何监视系统文件变化.....	274
实例 70	如何操作临时文件	188	实例 106	如何管理系统日志信息.....	278
实例 71	如何转换长文件名	192	实例 107	如何查询系统性能计数器.....	285
实例 72	如何选择多个文件	194	实例 108	如何弹出关闭光盘驱动器.....	288
实例 73	如何复制删除文件	195	实例 109	如何获取操作系统版本号.....	290
实例 74	如何加密解密文件	197	实例 110	如何获取系统特殊文件夹.....	292
实例 75	如何切分组合文件	205	实例 111	如何获取系统已运行时间.....	295
实例 76	如何判断文件是否存在.....	210	实例 112	如何获取系统逻辑驱动器.....	296
实例 77	如何获取文件打开方式.....	211	实例 113	如何获取逻辑驱动器大小.....	298
实例 78	如何获取文件尺寸大小.....	213	实例 114	如何获取光驱标志号码.....	300
实例 79	如何获取文件版本信息.....	214	实例 115	如何获取硬盘标志号码.....	301
实例 80	如何获取设置文件属性.....	217	实例 116	如何获取设置磁盘卷标.....	303
实例 81	如何获取设置文件时间.....	221	实例 117	如何获取用户关联域名.....	306
实例 82	如何分段获取文件名信息.....	224	实例 118	如何获取当前用户名.....	307
实例 83	如何获取设置文件夹时间.....	226	实例 119	如何获取计算机名称.....	308
实例 84	如何获取设置文件夹属性.....	228	实例 120	如何获取系统日志信息.....	309
实例 85	如何获取指定路径根目录.....	231	实例 121	如何获取系统进程信息.....	313
实例 86	如何获取指定目录内容.....	234	实例 122	如何获取系统当前进程.....	318
实例 87	如何获取设置当前目录.....	236	实例 123	如何获取系统执行程序.....	321
实例 88	如何创建多层文件夹	238	实例 124	如何获取系统安装字体.....	322
实例 89	如何删除多层文件夹	239	实例 125	如何获取系统当前日期.....	324
实例 90	如何设置共享文件夹	241	第 5 章 GDI+特效编程实例	328	
实例 91	如何利用 WinRAR 压缩文件	244	实例 126	如何显示油画图像	328
实例 92	如何利用 WinRAR 解压文件	246	实例 127	如何显示锐化图像	331
实例 93	如何以流式方式读写文件.....	249	实例 128	如何显示柔化图像	334
实例 94	如何读写 INI 文件.....	251	实例 129	如何显示木刻图像	337
实例 95	如何播放 AVI 动画文件	253	实例 130	如何显示镜像图像	340
实例 96	如何播放 GIF 动画文件	256	实例 131	如何显示镜框图像	343
实例 97	如何播放 MP3 歌曲文件.....	257	实例 132	如何显示灰度图像	347
实例 98	如何播放 Flash 动画文件.....	260	实例 133	如何显示浮雕图像	351

实例 134	如何显示飞入图像	354	实例 169	如何操作主从关系数据表	458
实例 135	如何显示反色图像	356	实例 170	如何设置数据表格显示风格	460
实例 136	如何显示多帧图像	358	实例 171	如何获取数据表格指定数据	465
实例 137	如何显示拉伸图形	362	实例 172	如何操作数据库事务处理	468
实例 138	如何显示百叶窗图像	369	实例 173	如何操作无连接数据表	471
实例 139	如何显示马赛克图像	371	实例 174	如何读取数据库记录图像	475
实例 140	如何添加图像标记	375	第 7 章 ASP.NET 应用编程实例	479	
实例 141	如何创建图像缩略图	378	实例 175	如何获取客户端主机地址	479
实例 142	如何根据分辨率显示图像	380	实例 176	如何获取客户端浏览器信息	483
实例 143	如何根据图像修剪文字	382	实例 177	如何获取客户端安全证书信息	485
实例 144	如何根据图形修剪文字	384	实例 178	如何获取 Web 服务器变量信息	487
实例 145	如何无级旋转文字	386	实例 179	如何获取 Web 页消息框返回值	489
实例 146	如何特效显示文字	387	实例 180	如何直接向客户端写入文本	491
实例 147	如何混合显示颜色图像	393	实例 181	如何在客户端显示指定图像	494
实例 148	如何控制图像显示质量	395	实例 182	如何在 Web 页中使用消息框	495
实例 149	如何转换图像文件格式	398	实例 183	如何在 Web 页中绘制条形图	497
第 6 章 数据库开发编程实例	403		实例 184	如何在 Web 页中设置快捷键	500
实例 150	如何在程序中创建数据库	403	实例 185	如何在 Web 页中显示提示信息	501
实例 151	如何使用 SQL 组件查询记录	409	实例 186	如何在 Web 页中显示 Excel 风格	502
实例 152	如何使用 BDP 组件查询记录	411	实例 187	如何在 Web 页中实现回车键 跳转	504
实例 153	如何使用 ODBC 组件查询记录	413	实例 188	如何在选项中绑定数据库记录	506
实例 154	如何使用 OLEDB 组件查询记录	417	实例 189	如何在 DataGrid 中自动编号	508
实例 155	如何以时间段方式查询记录	420	实例 190	如何隐藏传递 Web 页间参数	510
实例 156	如何以参数化方式查询记录	422	实例 191	如何向 Web 服务器上传文件	513
实例 157	如何获取多个查询数据表	425	实例 192	如何统计 Web 页中分页数据	514
实例 158	如何使用 XML 保存查询结果	427	实例 193	如何全屏显示和关闭 Web 页	518
实例 159	如何获取 SQL Server 版本号	430	实例 194	如何拦截不可预料异常信息	519
实例 160	如何获取聚合函数返回值	432	实例 195	如何使用缓存技术缓存 Web 页	520
实例 161	如何获取数据表记录信息	434	实例 196	如何设置浏览器首页链接地址	522
实例 162	如何获取数据表字段信息	436	实例 197	如何使用重复器控件显示记录	524
实例 163	如何获取数据库连接状态	439	实例 198	如何分页显示数据库记录信息	527
实例 164	如何浏览数据库记录信息	443	实例 199	如何操作客户端 Cookies 数据	530
实例 165	如何更新数据库记录信息	445	实例 200	如何使用图形验证码校验登录	532
实例 166	如何排序过滤数据库记录	449			
实例 167	如何在组合框中添加记录	453			
实例 168	如何操作数据库存储过程	455			

第 1 章 程序界面编程实例

实例 1 如何创建换肤窗体

实例目的

此实例主要介绍了如何创建能够根据图形图像的需要变换程序窗体界面的编程技术，即通常所说的程序窗体界面换肤。实例程序执行后，将显示一个不规则的图形窗体界面，如图 1-1 所示；在图 1-1 中单击“皮肤二”按钮，则显示该程序另一个不规则的图形窗体界面，如图 1-2 所示；在图 1-2 中单击“皮肤一”按钮，则显示图 1-1 所示的不规则图形窗体界面。实例程序可以通过单击“皮肤一”、“皮肤二”按钮在这两种不规则的图形窗体界面间任意切换，外观效果类似于东方影都播放器。

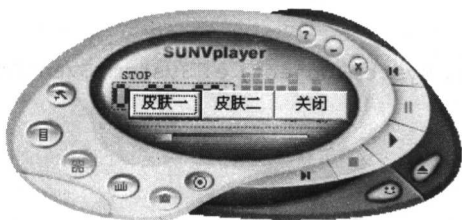


图 1-1



图 1-2

实现方法

在目前流行的大多数多媒体播放器和工具软件中，如 Windows 媒体播放器、东方影都、金山影霸等，它们的界面通常都是一些不规则的、相当个性化的图形图像界面，并且都备有多个不同的皮肤（不规则的图形窗体界面）来供用户进行选择，使用户感到赏心悦目。通常，在程序窗体中通过手工方式处理程序窗体界面的每个像素来实现这种界面换肤效果比较费事，并且具有一定的难度。在 .NET 中，通过设置程序窗体的属性来实现这种能够换肤的个性化窗体界面则比较简单。默认情况下，.NET 创建的 Windows Forms Application 应用程序窗体（TWinForm）有一个 TransparencyKey 属性，该属性能够设置程序窗体透明区域的颜色，它使得 TWinForm 窗体背景具有相同 TransparencyKey 属性所指定颜色的窗体区域透明显示；在窗体的透明区域执行的任何鼠标操作（如鼠标单击）都将传递到该透明区域下的窗口，如同这部分透明窗体完全不存在一样。例如，如果使程序窗体工作区的某部分透明，则在该区域上单击鼠标会将单击的事件通知不受阻挡地发送到该区域下的任何窗口。因此，如果以个性化的图形图像作为 TWinForm 窗体的背景图像，并且指定 TWinForm 窗体的 TransparencyKey 属性的颜色值与个性化窗体图像中的不需要部分（透明）的颜色一致，则程序换肤窗体界面就大功告

成。下面是实例程序据此通过设置 TWinForm 程序窗体的 TransparencyKey 属性实现创建程序换肤窗体界面的操作步骤:

(1) 启动 Delphi 2005, 选择 File\New\Windows Forms Application 菜单, 创建一个 Borland Delphi for the Microsoft.NET Framework 应用程序框架。

(2) 在画图或其他图像处理软件中制作两幅不规则的图像并保存为 Show1.bmp 和 Show2.bmp 两个图像文件, 注意其透明色(不需要的部分颜色)应为一个确定的颜色, 该颜色必须与程序窗体 TWinForm 的 TransparencyKey 属性指定的颜色值一致, 实例程序选择的是白色, 如图 1-1 和图 1-2 所示。然后复制这两个图像文件到程序运行文件夹中。

(3) 在 WinForm (WinForm.pas) 设计视图中选择该设计窗体 TWinForm, 在 Object Inspector 的属性页中设置其 TransparencyKey 属性为 White, 设置其 FormBorderStyle 属性为 None; 单击 BackgroundImage 属性右边的“...”按钮, 在弹出的文件“打开”对话框中选择步骤(2)制作的其中一幅图像文件后, 单击“打开”按钮返回。

(4) 选择 View\Tool Palette 菜单, 指向 Tool Palette 的 Windows Forms, 在程序设计窗体上(TWinForm)拖放三个 Button 控件, 在 Object Inspector 的属性页中设置三个 Button 控件的 Text 属性分别为“皮肤一”、“皮肤二”和“关闭”。

(5) 为“皮肤一”、“皮肤二”和“关闭”Button 控件的 Click 鼠标单击事件添加如下所示的实现代码:

```
//关闭
procedure TWinForm.Button3_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.Close;
end;
//皮肤二
procedure TWinForm.Button2_Click(sender: System.Object; e: System.EventArgs);
begin
    BackgroundImage:= Image.FromFile('Show2.bmp');
end;
//皮肤一
procedure TWinForm.Button1_Click(sender: System.Object; e: System.EventArgs);
begin
    BackgroundImage:= Image.FromFile('Show1.bmp');
end;
```

最后, 编译执行程序即可。

注意: 如果你的操作系统设置的颜色质量为“最高 32 位”, 可能按照此种方法创建的程序换肤窗体界面无效, 在此情况下, 请在系统显示属性中将颜色质量设置为较低的值(如 16 位)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 程序界面编程实例\实例.1 如何创建换肤窗体”。

实例2 如何创建托盘窗体

实例目的

此实例主要介绍了如何创建托盘窗体程序的编程技术。实例程序执行后，选择“新功能显示托盘”菜单，则程序主窗体隐藏，在系统任务栏通知区域（托盘）上显示一个实例程序图标，表示程序仍在运行；用鼠标右键单击程序托盘图标，在弹出的浮动菜单中选择“显示第一幅图像”菜单，则托盘图标和程序主窗体“第一幅图像”标签页会同时显示，效果如图2-1所示；如果选择“显示窗体”菜单，则会只显示程序窗体而不会同时也显示托盘图标；选择其他菜单会实现菜单标题所示的功能。

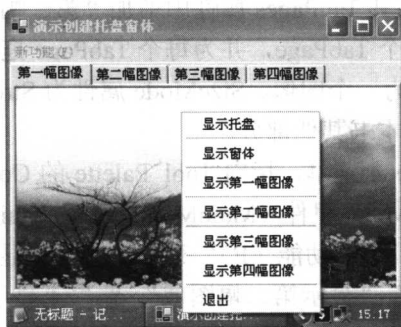


图 2-1

实现方法

托盘窗体通常用于不需要显示程序主窗体的服务类应用程序中，如金山词霸、SQL Server 服务管理等。这类程序在运行时，一般情况下不显示程序主窗体，而在系统任务栏通知区域（托盘）中放置一个图标，表明程序仍在运行。在.NET 中，创建托盘窗体程序可以使用 NotifyIcon 组件实现。用户只需通过设置 NotifyIcon 组件属性就可以创建托盘窗体，NotifyIcon 组件（类）常用的公共属性包括：

- (1) ContextMenu 属性，该属性用于获取或设置托盘图标的快捷菜单。
- (2) Icon 属性，该属性用于获取或设置当前托盘图标。
- (3) Text 属性，该属性用于获取或设置当鼠标停留在状态（托盘）区域图标上时显示的工具提示文本。
- (4) Visible 属性，该属性用于获取或设置一个值，该值指示图标在任务栏的状态通知区域（托盘）中是否可见。

一般情况下，不需要处理 NotifyIcon 组件的默认事件。如果需要使 NotifyIcon 组件实现更多的功能，例如响应鼠标双击等，则可以处理 NotifyIcon 组件的 DoubleClick 事件、MouseDown 事件。NotifyIcon 组件常见的公共事件包括：

- (1) Click 事件，该事件当用户单击状态区域的图标时发生。
- (2) DoubleClick 事件，该事件当用户在任务栏的状态通知区域中双击图标时发生。

(3) **MouseDown** 事件, 该事件当指针位于任务栏的状态通知区域中的图标上方时, 若用户按下鼠标按钮就会发生。

(4) **MouseMove** 事件, 该事件当指针位于任务栏的状态通知区域中的图标上方时, 若用户移动鼠标就会发生。

(5) **MouseUp** 事件, 该事件当指针位于任务栏的状态通知区域中的图标上方时, 若用户释放鼠标按钮就会发生。

下面是据此使用 **NotifyIcon** 组件创建托盘窗体程序的操作步骤:

(1) 启动 **Delphi 2005**, 选择 **File\New\Windows Forms Application** 菜单, 创建一个 **Borland Delphi for the Microsoft.NET Framework** 应用程序框架。

(2) 选择 **View\Tool Palette** 菜单, 指向 **Tool Palette** 的 **Windows Forms**, 在程序设计窗体上 (**TWinForm**) 拖放一个 **TabControl** 控件 (**TabControl1**), 在 **Object Inspector** 的属性页中设置其 **Dock** 属性为 **Fill**; 然后在其 **TabPage** 属性中增加“第一幅图像”、“第二幅图像”、“第三幅图像”和“第四幅图像”四个 **TabPage**, 并为每个 **TabPage** 拖放一个 **PictureBox** 控件, 设置 **PictureBox** 控件的 **Image** 属性为一个图像、**SizeMode** 属性为 **StretchImage**。关于 **TabControl** 控件更详细的操作方法, 请参考本书其他部分。

(3) 选择 **View\Tool Palette** 菜单, 指向 **Tool Palette** 的 **Components**, 在程序设计窗体上 (**TWinForm**) 拖放一个 **MainMenu** 组件 (**MainMenu1**)。为 **MainMenu** 组件 (**MainMenu1**) 增加一个“新功能 (F)”菜单组, “新功能 (F)”菜单组包括“显示托盘”、“-”、“显示窗体”、“-”、“显示第一幅图像”、“-”、“显示第二幅图像”、“-”、“显示第三幅图像”、“-”、“显示第四幅图像”、“-”和“退出”菜单项。关于 **MainMenu** 组件更详细的操作方法, 请参考本书其他部分。

(4) 为“显示托盘”、“显示窗体”、“显示第一幅图像”、“显示第二幅图像”、“显示第三幅图像”、“显示第四幅图像”和“退出”菜单项的 **Click** 鼠标单击事件添加如下所示的实现代码:

```
//显示托盘
procedure TWinForm.MenuItem2_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.NotifyIcon1.Visible:=True;
    Self.Visible:=False;
end;
//显示窗体
procedure TWinForm.MenuItem4_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.NotifyIcon1.Visible:=False;
    Self.Visible:=True;
end;
//显示第一幅图像
procedure TWinForm.MenuItem6_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.Visible:=True;
    Self.TabControl1.SelectedTab:=Self.TabPage1;
end;
//显示第二幅图像
procedure TWinForm.MenuItem8_Click(sender: System.Object; e: System.EventArgs);
```

```
begin
    Self.Visible:=True;
    Self.TabControl1.SelectedTab:=Self.TabPage2;
end;
//显示第三幅图像
procedure TForm1.MenuItem10_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.Visible:=True;
    Self.TabControl1.SelectedTab:=Self.TabPage3;
end;
//显示第四幅图像
procedure TForm1.MenuItem12_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.Visible:=True;
    Self.TabControl1.SelectedTab:=Self.TabPage4;
end;
//退出
procedure TForm1.MenuItem14_Click(sender: System.Object; e: System.EventArgs);
begin
    Self.Close;
end;
```

(5) 选择 View\Tool Palette 菜单, 指向 Tool Palette 的 Components, 在程序设计窗体上 (TwinForm) 拖放一个 ContextMenu 组件 (ContextMenu1)。

(6) 为应用程序窗体 TwinForm 的 Load 事件添加如下所示的实现代码:

```
//创建托盘菜单(ContextMenu1)
procedure TForm1.TwinForm_Load(sender: System.Object; e: System.EventArgs);
begin
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem2.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem3.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem4.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem5.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem6.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem7.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem8.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem9.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem10.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem11.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem12.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem13.CloneMenu);
    Self.ContextMenu1.MenuItems.Add(Self.MenuItem14.CloneMenu);
end;
```

(7) 选择 View\Tool Palette 菜单, 指向 Tool Palette 的 Components, 在程序设计窗体上 (TwinForm) 拖放一个 NotifyIcon 组件 (NotifyIcon1)。在 Object Inspector 的属性页中设置其 Text 属性为“我的托盘窗体程序”, 设置其 ContextMenu 属性为 ContextMenu1, 设置其 Visible 属性为 True, 单击 Icon 属性右边的“...”按钮, 在弹出的文件“打开”对话框中选择一个图标文件后, 单击“打开”按钮返回。则此图标文件中的图标就成为托盘图标。

最后, 编译执行程序即可。

实例源代码

有关程序项目文件的全部源代码及执行程序，请参考配套光盘“源代码/第1章 程序界面编程实例/实例2 如何创建托盘窗体”。

实例3 如何创建渐变色窗体

实例目的

此实例主要介绍如何以渐变色作为程序窗体背景的编程技术。实例程序执行后，将从上到下出现一个由蓝变白的渐变色背景，任意缩放程序窗体大小，则此渐变效果将根据窗体的大小随之改变，效果如图3-1所示。



图 3-1

实现方法

在.NET 程序中，创建颜色渐变效果图形可以使用 GDI+的 `LinearGradientBrush` 类实现。该类封装双色渐变和自定义多色渐变的功能。默认情况下，双色渐变是沿指定直线从起始色到结束色的均匀水平线性混合。通过操作 `LinearGradientBrush` 类的公共属性和方法能够实现多种渐变效果。`LinearGradientBrush` 类常用的公共属性如下：

- (1) `Blend` 属性，该属性用于获取或设置一个 `Blend` 对象，它指定为渐变定义自定义过渡的位置和因子。
- (2) `GammaCorrection` 属性，该属性用于获取或设置一个值，该值指示是否为该 `LinearGradientBrush` 对象启用伽玛修正。
- (3) `InterpolationColors` 属性，该属性用于获取或设置一个定义多色线性渐变的 `ColorBlend` 对象。
- (4) `LinearColors` 属性，该属性用于获取或设置渐变的起始色和结束色。
- (5) `Rectangle` 属性，该属性用于获取定义渐变的起始点和终结点的矩形区域。
- (6) `Transform` 属性，该属性用于获取或设置一个 `Matrix` 对象，该对象为此 `LinearGradientBrush` 对象定义局部几何变形。
- (7) `WrapMode` 属性，该属性用于获取或设置 `WrapMode` 枚举，它指定其环绕模式。`WrapMode` 枚举的枚举成员及其说明如表 3-1 所示。

表 3-1

成员名称	说明
Clamp	将纹理和渐变向对象边界拉拢
Tile	平铺渐变或纹理
TileFlipX	水平反转纹理或渐变, 然后平铺该纹理或渐变
TileFlipXY	水平和垂直反转纹理或渐变, 然后平铺该纹理或渐变
TileFlipY	垂直反转纹理或渐变, 然后平铺该纹理或渐变

LinearGradientBrush 类常用的公共方法如下:

(1) MultiplyTransform() 方法, 该方法通过预先计算指定的 Matrix, 将表示此 LinearGradientBrush 的局部几何变形的 Matrix 对象与该指定的 Matrix 相乘。

(2) ResetTransform() 方法, 该方法将 Transform 属性重置为相同。

(3) RotateTransform() 方法, 该方法将局部几何变形旋转指定大小。该方法预先计算对变形的旋转。

(4) ScaleTransform() 方法, 该方法将局部几何变形缩放指定数量。该方法预先计算对变形的缩放矩阵。

(5) SetBlendTriangularShape() 方法, 该方法创建一个从中心色向两端单个颜色线性过渡的线性渐变过程。

(6) SetSigmaBellShape() 方法, 该方法创建基于钟形曲线的渐变过渡过程。

(7) TranslateTransform() 方法, 该方法将局部几何变形转换指定的尺寸。该方法将预先计算对变形的转换。

实例程序将直接在 LinearGradientBrush 类的构造函数中设置 LinearGradientBrush 的公共属性, 以产生一个渐变画刷, 在程序窗体客户区绘制渐变背景。

注意: 程序窗体的绘制操作在 TWinForm 的 Paint 事件中进行, 为使绘制操作与窗体尺寸改变同步, 通常应该处理 TWinForm 的 Resize 事件。

下面是实例程序据此创建渐变色程序窗体的操作步骤:

(1) 启动 Delphi 2005, 选择 File/New/Windows Forms Application 菜单, 创建一个 Borland Delphi for the Microsoft.NET Framework 应用程序框架。

(2) 为应用程序窗体 TWinForm 的 Paint 事件和 Resize 事件添加如下所示的实现代码:

```
//产生渐变色窗体
procedure TWinForm.TWinForm_Paint(sender: System.Object;
  e: System.Windows.Forms.PaintEventArgs);
var
  g:Graphics;
  MyBrush:System.Drawing.Drawing2D.LinearGradientBrush;
begin
  g:=Self.CreateGraphics;
  MyBrush:=LinearGradientBrush.Create(ClientRectangle,Color.Blue,
    Color.White, LinearGradientMode.Vertical);
  g.FillRectangle(MyBrush, ClientRectangle);
end;
//更新窗体变化
```

```

procedure TWinForm.TWInForm_Resize(sender: System.Object; e: System.EventArgs);
begin
    Self.Invalidate;
end;

```

(3) 在 WinForm.pas 源代码文件中的 Uses 部分中添加 LinearGradientBrush 类的命名空间声明, 如下面的代码所示:

```

uses
....., System.Drawing.Drawing2D;

```

最后, 编译执行程序即可。

实例源代码

有关程序项目文件的全部源代码及执行程序, 请参考配套光盘“源代码\第 1 章 程序界面编程实例\实例 3 如何创建渐变色窗体”。

实例 4 如何创建标签页窗体

实例目的

此实例主要介绍如何创建标签页窗体程序的编程技术。实例程序执行后, 将显示带有两个标签页的程序窗体, 在标签页的标签上带有图标和提示文本。单击“成员简历”标签页, 则弹出一个界面显示成员简历; 单击“成员照片”标签页, 则弹出一个界面显示成员照片, 效果如图 4-1 所示。



图 4-1

实现方法

标签页窗体也叫选项卡窗体, 该类型的窗体在复杂的程序界面中使用较多, 如 Borland C++Builder 的工具箱窗体, 操作系统的显示属性页都是采用此类型窗体。在 .NET 中, 创建标签页窗体通常使用 TabControl 控件 (类), TabControl 控件实际上是一个容器, 该容器中的所有标签页 tabPage 由 TabControl.TabPages 集合管理。一般情况下, 设计标签页窗体可以通过可视化的方式实现, 如果需要在应用程序中动态设置标签页, 则应该通过操作 TabControl 类和 tabPage 类的公共属性实现。TabControl 类常用的公共属性包括:

- (1) DrawMode 属性, 该属性用于获取或设置绘制控件的选项卡页的方式。
- (2) ImageList 属性, 该属性用于获取或设置在控件的选项卡上显示的图像。

- (3) `ItemSize` 属性, 该属性用于获取或设置控件的选项卡的大小。
- (4) `Multiline` 属性, 该属性用于获取或设置一个值, 该值指示是否可以显示一行以上的选项卡。
- (5) `Padding` 属性, 该属性用于获取或设置控件的选项卡页上每个项周围的空间量。
- (6) `RowCount` 属性, 该属性用于获取控件的选项卡条中当前正显示的行数。
- (7) `SelectedIndex` 属性, 该属性用于获取或设置当前选定的选项卡页的索引。
- (8) `SelectedTab` 属性, 该属性用于获取或设置当前选定的选项卡页。
- (9) `ShowToolTips` 属性, 该属性用于获取或设置一个值, 该值指示当鼠标移到选项卡上时是否显示该选项卡的“工具提示”。
- (10) `SizeMode` 属性, 该属性用于获取或设置调整控件的选项卡大小的方式。
- (11) `TabCount` 属性, 该属性用于获取选项卡条中选项卡的数目。
- (12) `TabPage` 属性, 该属性用于获取该选项卡控件中选项卡页的集合。

`TabPage` 类常用的公共属性包括:

- (1) `ImageIndex` 属性, 该属性用于获取或设置在选项卡上显示的图像的索引。
- (2) `Text` 属性, 该属性用于获取或设置要在选项卡上显示的文本。
- (3) `ToolTipText` 属性, 该属性用于获取或设置该选项卡的工具提示文本。

注意: 如果需要在应用程序中动态增加或删除窗体标签页, 则应该以集合的形式执行增加或删除操作。

下面是实例程序使用 `TabControl` 控件创建标签页窗体程序的操作步骤:

(1) 启动 Delphi 2005, 选择 `File\New\Windows Forms Application` 菜单, 创建一个 Borland Delphi for the Microsoft.NET Framework 应用程序框架。

(2) 选择 `View\Tool Palette` 菜单, 指向 `Tool Palette` 的 `Components`, 在程序设计窗体上 (TWinForm) 拖放一个 `ImageList` 组件 (`ImageList1`), 在 `Object Inspector` 的属性页中单击其 `Images` 属性右边的“...”按钮, 弹出“`Image` 集合编辑器”对话框, 在该对话框中单击“添加”按钮, 在弹出的文件“打开”对话框中选择一个图标文件后, 单击“打开”按钮返回, 则该图标文件中的图标将显示在“成员”列表中, 按照相同的方法再增加一个图标, 如图 4-2 所示, 完成后单击“确定”按钮返回。添加的图标将作为标签页的标签图标显示。

(3) 选择 `View\Tool Palette` 菜单, 指向 `Tool Palette` 的 `Windows Forms`, 在程序设计窗体上 (TWinForm) 拖放一个 `TabControl` 控件 (`TabControl1`), 在 `Object Inspector` 的属性页中设置其 `Dock` 属性为 `Fill`, 设置其 `ImageList` 属性为 `ImageList1`, 设置其 `ShowToolTips` 属性为 `True`, 单击其 `TabPage` 属性右边的“...”按钮, 弹出“`TabPage` 集合编辑器”对话框, 在该对话框中单击“添加”按钮, 将在“成员”列表中新增一个 `TabPage1`, 在右边的“`TabPage1` 属性”中设置其 `Text` 属性为“成员简历”, 设置其 `ImageIndex` 属性为 1, 设置其 `ToolTipText` 属性为“显示成员简历”, 如图 4-3 所示; 再单击“添加”按钮, 则会在“成员”列表中再新增一个 `TabPage2`, 在右边的“`TabPage2` 属性”中设置其 `Text` 属性为“成员照片”、设置其 `ImageIndex` 属性为 0, 设置其 `ToolTipText` 属性为“显示成员照片”, 完成后单击“确定”按钮返回。

(4) 在 `TabControl` 控件 (`TabControl1`) 设计视图中单击“成员简历”标签页标签, 切换到 `TabControl` 控件 (`TabControl1`) 的“成员简历”标签页。选择 `View\Tool Palette` 菜单, 指向 `Tool Palette` 的 `Windows Forms`, 在 `TabControl` 控件 (`TabControl1`) 的“成员简历”标签页设计