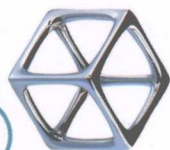


源代码免费下载

商业开发



代码库系列

Visual FoxPro

案例开发集锦

(第二版)

陆 岚 编著
康祥顺 审校



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

商业开发代码库系列

Visual FoxPro案例开发集锦

(第二版)

陆 岚 编著

康祥顺 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要通过具体的案例介绍如何运用Visual FoxPro开发工具开发实际的应用程序,从基本应用到高级处理都有介绍,包括界面设计、网络处理、多媒体应用、系统文件和基本数据处理、报表设计、数据库应用编程等。每个案例的编排都严格按照读者的阅读习惯进行,都是由具有丰富经验的项目开发人员亲手编写,大部分案例已经在项目开发过程中经过了实践的检验,它们是指导读者进入实战型程序设计师领域的一座明亮的灯塔。

本书适合大中专院校学生、程序设计人员和Visual Studio爱好者学习和使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Visual FoxPro案例开发集锦/陆岚编著. —2版. —北京:电子工业出版社, 2008.10
(商业开发代码库系列)

ISBN 978-7-121-07245-1

I. V… II. 陆… III. 关系数据库—数据库管理系统, Visual FoxPro IV. TP311.138

中国版本图书馆CIP数据核字(2008)第123547号

责任编辑:李红玉

印 刷:北京天竺颖华印刷厂

装 订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

开 本:787×1092 1/16 印张:28.5 字数:720千字

印 次:2008年10月第1次印刷

定 价:51.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至zlts@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

Visual FoxPro曾经是微软公司推出的优秀桌面型数据处理工具，而且现在也有很多人利用Visual FoxPro来开发项目，很多大型公司仍然采用Visual FoxPro系统管理日常工作。自Visual FoxPro问世以来，其友好的集成开发界面、可视化的双向开发模式、良好的数据处理方式以及高效的程序开发和程序运行，备受广大程序设计师的好评。随着计算技术的发展，对于系统的跨平台集成和相互通信的需求越来越多，到目前为止，微软公司已经将Visual FoxPro升级到9.0版本，在原来强大的数据处理功能的基础上，优化和增加了报表设计、网络处理和对.NET的支持，也使得Visual FoxPro能够利用.NET技术平台提高系统的平台异构能力。可以说，Visual FoxPro是紧跟当前IT技术，顺应历史而前进的一种优异的数据处理工具。

学习一门编程语言，最为关键的不是去死记函数或者方法，而是掌握如何应用这些函数、方法去解决实际的问题。在学习编程语言的过程中，我们很可能都有过这样的经历：这门编程语言的函数、方法也知道不少，利用函数、方法也能够编写几个简单的小程序，但是，当翻看其他书籍或者使用别人编写的小软件时，才知道原来这个函数还能这样用。这是自己钻研所不能取得的成果，而且会事倍功半；然而本书汇集了Visual FoxPro开发工具所能解决问题的一些经典案例，无论是界面设计，还是网络处理，无论是简单的函数应用，还是高级的底层API函数的应用，都包括在本书之中。阅读本书后，读者能够在最短的时间内，迅速提升自己的编程水平以及解决问题的方式、方法，避免闭门造车所造成的低水平徘徊，甚至丢失信心。因此，如果你已经学习了Visual FoxPro，具有了一定的Visual FoxPro的语法基础，那么本书将是引领你成为真正的程序员的指南性图书，他将会使你的编程水平发生质的飞跃！

本书共7章，每一章都是由既相互独立，又相互间有一定逻辑关系的案例组成的。每一个案例基本上都分为4个部分，包括：

- 案例运行效果与操作：让读者对本案例有一个直观的印象，以便于读者进一步了解如何解决这个问题；

- 制作要点：让读者明白，要解决这个问题，应该采取什么方法，使用哪些函数、组件和技巧；

- 步骤详解：告诉读者如何一步一步地实现这个案例，以便于读者自己动手实现案例效果，达到加深理解、锻炼自我的目的；

- 程序源代码添加与解释：附带了案例的源代码，读者只要将这些源代码复制到自己的程序中就能够实现程序的功能；同时对于程序中应用到的难以理解的函数或者对象都添加了详细的注释，以便于读者弄清楚问题的根源。

这些案例的源代码，既可以用来试验某个编程的概念，也可以剪贴到自己的程序中，实现某种特定的功能，达到迅速解决问题的效果。通过阅读本书，将学习到以下几个方面的知识：

- 学会如何设计专业级的用户界面；
- 学会如何在程序中处理多媒体编程的问题；

- 学会如何在程序中处理系统文件；
- 学会如何在程序中调用Windows的底层API函数；
- 学会如何在系统中处理数据库；
- 学会如何应用网络组件、系统提供的类处理网络操作；
- 学会如何在程序中处理复杂的跨平台的应用问题；
- 学会如何应用提供的组件、函数、对象实际开发功能复杂的应用系统，让读者了解实际开发一个应用系统的过程；
- 其他更多的内容。

为了方便读者阅读、测试和使用本书提供的案例，本书还在华信教育资源网上提供了每个案例的源代码。源代码严格按照章节编排，例如第1章“案例1”的源代码的存放位置为：源代码\第1章 用户表单设计\设计透明窗体，每个案例目录下附带了所有用到的资源文件，如图片、数据表等。每个案例都是作者在Windows XP+Visual FoxPro 9.0环境下进行严格测试后再组织起来的，请尽管放心使用。但是可能存在这样的问题：案例文件打开后无法找到引用的文件、数据表等，如果出现这方面的问题，你可以遵照实现步骤自己制作案例程序，然后把源代码复制到相应的部分，或者打开程序中的对象检查器，重新设置相关属性。

但是，鱼和熊掌不能兼得。这样虽然方便了读者，但同时很可能使读者对代码的熟悉程度、技术要点和编程思路等问题的领会程度降低。因此，强烈建议读者先看书案例的运行效果，然后再看源代码，如果心中已经有数了，可以自己动手举一反三，设计类似的程序来检验自己的想法，或者对程序进行修改和改进。

本书每一章内容都是由具有丰富经验的实际从事项目开发的人员编写的，大部分的案例都已经在项目开发过程中得以运用，因此其操作性、实践性和指导性都相当强，是当今市面上难得的实战级和专业级的图书。参加本书编写工作的有：张庆、崔竞、张强、董军、欧立奇、保春艳、李欣、梅光耀、黄显堂、康祥琴、赵辉。这些同志虽然从事着繁重的项目开发工作，但是仍然在工作之余，兢兢业业、严谨认真地编写了这本图书，保证了本书的质量和进度。

在本书的编写过程中黄显堂提出了很多宝贵意见，包括亲自审核把关；另外，为本书最后出版做出不懈努力的电子工业出版社和北京美迪亚电子信息有限公司的所有同志，在此对他们一并表示最诚挚的谢意！

本书中所有案例，编者都在Windows XP+Visual FoxPro 9.0环境下进行了测试，由于水平有限，尽管进行了严格的审核和测试，书中还是难免有疏漏之处，敬请广大读者不吝赐教，编者在此表示感谢！

为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”（<http://www.hxedu.com.cn>），在“资源下载”频道的“图书资源”栏目下载。

目 录

第1章 用户表单设计	1
案例1 设计透明窗体	1
案例2 创建不规则窗体	4
案例3 可停靠工具栏的设计	9
案例4 动态创建自定义菜单	18
案例5 创建独立于菜单栏的菜单	23
案例6 为表单添加可伸缩的菜单	29
案例7 将菜单、工具栏添加到顶层表单	35
案例8 如何为表单创建功能键	45
案例9 在表单中使用滑块控件	51
案例10 在列表框中添加自定义图形	53
案例11 为表单提供What's This样式的帮助	56
案例12 创建Windows XP主题风格的表单	60
案例13 嵌入式桌面计时器	62
第2章 网络应用	66
案例1 E-mail邮件发送和接收	66
案例2 程序中的超级链接	74
案例3 HTML报表生成	76
案例4 在Visual FoxPro中调用Web服务	78
案例5 通过远程数据源创建综合报表	85
案例6 TCP点对点(P2P)联机聊天	90
案例7 TCP点对点(P2P)传送文件	97
案例8 在程序中加入网上图片	102
案例9 自定义IE浏览器	103
第3章 图形图像与多媒体处理	107
案例1 图像浏览器	107
案例2 读写大型二进制媒体数据	110
案例3 模拟Windows画图程序	115
案例4 在数据表格中批量浏览图像	129
案例5 电子石英钟	131

案例6	在VFP较低版本中处理图像数据	133
案例7	自定义的Windows Media Player媒体播放器	138
案例8	使用MCI接口播放多媒体文件	141
案例9	CD唱机	148
案例10	可爱的小人动画	151
案例11	Flash动画播放器	156
案例12	为应用程序添加多媒体Logo	159
第4章 文件和基本数据处理		163
案例1	问卷调查表单的设计	163
案例2	在资源管理器中拖放文件	166
案例3	将小写金额转换为大写	173
案例4	时间元素的分别提取	175
案例5	子串的替换	180
案例6	简单的文本编辑器	183
案例7	遍历文件和目录	186
案例8	通过FSO文件模型对象操作文件和文件夹	189
案例9	获取驱动器属性	191
案例10	搜索包含指定字符串的文件	192
案例11	读写文本文件	195
案例12	随机数的应用	201
第5章 数据库处理		204
案例1	为应用程序加载数据环境	204
案例2	创建数据库	209
案例3	创建数据库表	211
案例4	动态取得数据表的属性信息	215
案例5	为数据表字段创建唯一的字段默认值	222
案例6	使用数据库定义存储过程	226
案例7	给数据表动态排序	227
案例8	动态修改表结构	236
案例9	使用带有多个参数的视图查询数据	243
案例10	动态创建SQL视图	246
案例11	限制查询结果的记录数	248
案例12	检查数据表是否损坏	253
案例13	单数据浏览和编辑	256
案例14	一对多关系表的浏览和更新	259

案例15	在数据表格中进行计算	267
案例16	将数据表转换为统计图表	269
案例17	将任意DBF数据表转换为Word文档	272
案例18	将任意DBF数据表转换为Excel电子表格	281
案例19	搜索备注字段中的字符串	287
案例20	跟踪数据表	290
案例21	多条件查询设计	294
案例22	应用程序的发布	306

第6章 报表、标签设计与处理 309

案例1	分组报表的设计	309
案例2	精确打印报表字段	313
案例3	一对多报表设计	318
案例4	账务式格式化报表的设计	320
案例5	有条件地显示数据记录	323
案例6	设计带分组的报表	327
案例7	使用向导设计标签	330
案例8	使用“标签设计器”设计标签	332
案例9	在报表中打印动态统计图表	334
案例10	在表单中预览报表	341
案例11	在自定义的表单中控制报表的显示	350
案例12	在报表的标头带区输出计算值	359
案例13	将报表输出为多种文件类型	364
案例14	通过Word精确打印表中的“General”类型字段	367
案例15	自定义报表的“打印设置”对话框	370
案例16	取得打印报表的打印信息	380
案例17	自定义打印预览工具栏	386

第7章 图书馆信息化管理系统 395

7.1	系统需求分析	395
7.2	系统功能设计	395
7.3	系统功能模块运行效果与操作	397
7.4	创建项目文件	407
7.5	数据库及数据表的设计与创建	408
7.6	设计系统启动画面和用户登录表单	415
7.7	创建主控表单	417
7.8	创建系统菜单	418

7.9	设计系统工具栏	421
7.10	设计“图书信息管理”表单	423
7.11	设计“读者信息管理”表单	425
7.12	设计“信息查询”表单	428
7.13	设计“图书借阅”表单	431
7.14	设计“还书”表单	434
7.15	设计“报表打印”表单	436
7.16	设计“数据备份”表单	440
7.17	设计“数据恢复”表单	442
7.18	设计“数据清空”表单	445
7.19	“读者信息”报表的设计	446
7.20	其他报表的设计	447
8.1	设计“读者信息”报表	448
8.2	设计“图书信息管理”报表	449
8.3	设计“信息查询”报表	450
8.4	设计“图书借阅”报表	451
8.5	设计“还书”报表	452
8.6	设计“报表打印”报表	453
8.7	设计“数据备份”报表	454
8.8	设计“数据恢复”报表	455
8.9	设计“数据清空”报表	456
8.10	设计“读者信息”报表	457
8.11	设计“图书信息管理”报表	458
8.12	设计“信息查询”报表	459
8.13	设计“图书借阅”报表	460
8.14	设计“还书”报表	461
8.15	设计“报表打印”报表	462
8.16	设计“数据备份”报表	463
8.17	设计“数据恢复”报表	464
8.18	设计“数据清空”报表	465
8.19	设计“读者信息”报表	466
8.20	设计“图书信息管理”报表	467
8.21	设计“信息查询”报表	468
8.22	设计“图书借阅”报表	469
8.23	设计“还书”报表	470
8.24	设计“报表打印”报表	471
8.25	设计“数据备份”报表	472
8.26	设计“数据恢复”报表	473
8.27	设计“数据清空”报表	474
8.28	设计“读者信息”报表	475
8.29	设计“图书信息管理”报表	476
8.30	设计“信息查询”报表	477
8.31	设计“图书借阅”报表	478
8.32	设计“还书”报表	479
8.33	设计“报表打印”报表	480
8.34	设计“数据备份”报表	481
8.35	设计“数据恢复”报表	482
8.36	设计“数据清空”报表	483
8.37	设计“读者信息”报表	484
8.38	设计“图书信息管理”报表	485
8.39	设计“信息查询”报表	486
8.40	设计“图书借阅”报表	487
8.41	设计“还书”报表	488
8.42	设计“报表打印”报表	489
8.43	设计“数据备份”报表	490
8.44	设计“数据恢复”报表	491
8.45	设计“数据清空”报表	492
8.46	设计“读者信息”报表	493
8.47	设计“图书信息管理”报表	494
8.48	设计“信息查询”报表	495
8.49	设计“图书借阅”报表	496
8.50	设计“还书”报表	497
8.51	设计“报表打印”报表	498
8.52	设计“数据备份”报表	499
8.53	设计“数据恢复”报表	500
8.54	设计“数据清空”报表	501
8.55	设计“读者信息”报表	502
8.56	设计“图书信息管理”报表	503
8.57	设计“信息查询”报表	504
8.58	设计“图书借阅”报表	505
8.59	设计“还书”报表	506
8.60	设计“报表打印”报表	507
8.61	设计“数据备份”报表	508
8.62	设计“数据恢复”报表	509
8.63	设计“数据清空”报表	510
8.64	设计“读者信息”报表	511
8.65	设计“图书信息管理”报表	512
8.66	设计“信息查询”报表	513
8.67	设计“图书借阅”报表	514
8.68	设计“还书”报表	515
8.69	设计“报表打印”报表	516
8.70	设计“数据备份”报表	517
8.71	设计“数据恢复”报表	518
8.72	设计“数据清空”报表	519
8.73	设计“读者信息”报表	520
8.74	设计“图书信息管理”报表	521
8.75	设计“信息查询”报表	522
8.76	设计“图书借阅”报表	523
8.77	设计“还书”报表	524
8.78	设计“报表打印”报表	525
8.79	设计“数据备份”报表	526
8.80	设计“数据恢复”报表	527
8.81	设计“数据清空”报表	528
8.82	设计“读者信息”报表	529
8.83	设计“图书信息管理”报表	530
8.84	设计“信息查询”报表	531
8.85	设计“图书借阅”报表	532
8.86	设计“还书”报表	533
8.87	设计“报表打印”报表	534
8.88	设计“数据备份”报表	535
8.89	设计“数据恢复”报表	536
8.90	设计“数据清空”报表	537
8.91	设计“读者信息”报表	538
8.92	设计“图书信息管理”报表	539
8.93	设计“信息查询”报表	540
8.94	设计“图书借阅”报表	541
8.95	设计“还书”报表	542
8.96	设计“报表打印”报表	543
8.97	设计“数据备份”报表	544
8.98	设计“数据恢复”报表	545
8.99	设计“数据清空”报表	546
8.100	设计“读者信息”报表	547



第1章

用户表单设计

本章内容

- ❖ 设计透明窗体
- ❖ 创建不规则窗体
- ❖ 可停靠工具栏的设计
- ❖ 动态创建自定义菜单
- ❖ 创建独立于菜单栏的菜单
- ❖ 为表单添加可伸缩的菜单
- ❖ 将菜单、工具栏添加到顶层表单
- ❖ 在表单中使用滑块控件
- ❖ 在列表框中添加自定义图形
- ❖ 为表单提供What's This形式的帮助
- ❖ 创建Windows XP风格的表单
- ❖ 嵌入式桌面计时器

案例1 设计透明窗体

案例运行效果与操作

窗体的透明效果以及淡入淡出效果在一些专业软件中经常使用，例如方正电子图书阅读软件，窗体逐渐地出现，然后退出时渐渐地消失，这样可以为窗体增添亮点和乐趣。创建透明的窗体需要使用到Windows API函数SetWindowLong()和SetLayeredWindowAttributes()，利用这两个函数就可以调用系统提供的功能设计窗体的特性。窗体的透明度可以设置为0~255之间的任意整数值，值越小，透明度越高。

注意：透明窗体只能在Windows 2000或者更高的版本中才有效，而且必须是顶层窗体。

案例运行效果如图1.1所示。

程序运行后，可以通过拖动滑块来改变窗体的透明度。

制作要点

1. 分层窗口的应用

在应用程序中，根据需要，可能希望定制自己的独具特色的窗口，这需要用到SetWindowRgn()函数来指定希望作为新窗口的区域或者路径。但是这样的窗体会带来一些执行效果上的不足，例如需要重新绘制原窗体和新指定的窗体区域，两者重合的区域在绘制时，可能产生闪烁点，而且执行时也会产生更多、更为频繁的系统消息。为了避免这种现象的发生，在Windows 2000以后版本的操作系统中，增加了分层窗口的技术支持。也就是在使用分层窗口后，系统会自动重绘两个不同的窗体，产生更为平滑的绘制效果，不会出现闪

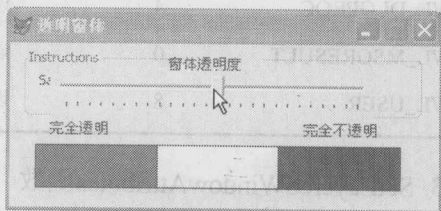


图1.1 透明的窗体

烁等现象。更为关键的是，自己定制的窗体区域只能实现全透明效果，因为要将原来的窗体隐藏，而分层窗口可以实现0~255之间的任意透明效果，值越小透明度越高，也就是说0表示完全透明，255表示完全不透明。在定义分层窗口时，需要用到两个Windows API函数：SetWindowLong()和SetLayeredWindowAttributes()。指定一个分层窗口首先需要调用SetWindowLong()函数指定窗体的属性为WS_EX_LAYERED（扩展分层窗口），然后再调用SetLayeredWindowAttributes()函数指定分层窗口的透明度，指定值的范围为0~255。

2. Windows API函数SetWindowLong

该函数用于将一个窗口指定为分层窗口，API函数声明格式为：

```
Declare Integer SetWindowLong in User32 Long hwnd,Long nIndex,Long dwNewLong
```

其中，参数的含义如下：

Hwnd：窗口句柄，可以通过表单的属性Hwnd来获取。

nIndex：表示要设置的窗口的属性，如表1.1所示。

dwNewLong：要为属性设置的新值。在此，我们需要将新值设置为WS_EX_LAYERED（扩展分层窗口），而该值定义的常量地址为0x00080000。

表1.1 nIndex窗口属性

常量	属性值	属性功能
GWL_WNDPROC	-4	为窗口过程设置一个新的地址
GWL_HINSTANCE	-6	设置窗口的句柄
GWL_HWNDPARENT	-8	设置父窗口的句柄
GWL_ID	-12	设置窗口的标识符
GWL_STYLE	-16	设置窗口的风格
GWL_EXSTYLE	-20	设置窗口的扩展风格
GWL_USERDATA	-21	设置与窗口有关的32位值，每一个窗口均有一个由创建该窗口的应用程序使用的32位值
DWL_DLGPROC	4	设置对话框过程的新地址
DWL_MSGRESULT	0	设置在对话框过程中处理的消息返回值
DWL_USER	8	设置应用程序私有的额外信息，例如一个句柄或者指针

3. SetLayeredWindowAttributes函数

此函数用于设置分层窗口的透明度，也就是透明颜色的键值。在VFP中的声明格式为：

```
Declare Integer SetLayeredWindowAttributes in User32
Long hwnd,Long crKey,Long bAlpha,Long dwFlags
```

函数参数的功能如下：

Hwnd：指定为分层窗口的窗口句柄。

crKey：分层窗口中将要指定为透明的颜色值，所有这种颜色都将设置为透明，而且Windows消息不对设置为透明的部分进行消息检测，颜色值可以通过RGB()函数取得。

bAlpha：要设置为透明的透明度，在0~255之间。

dwFlags: 函数的方式, 如果设置为1, 则将参数crKey指定的颜色设置为透明, 如果设置为2, 则将根据参数aAlpha的值来设置整个窗体的透明度。如果设置为0, 则还原原窗体的属性, 使窗体不透明。本案例需要根据用户指定的透明度来设置窗体透明, 因此, 应该将此参数设置为2。

步骤详解

1. 新建一个表单, 将表单的Caption属性设置为“透明窗体”, 再将ShowWindow属性设置为“2-顶层表单”; 然后在表单中添加3个Shape (形状) 控件, 然后分别将BackColor属性设置为红色 (255,0,0)、白色 (255,255, 255)、蓝色 (0,0,255); 再添加3个标签控件, Caption属性分别为“透明度”、“完全透明”、“完全不透明”。

2. 选择系统菜单“工具”>“选项”, 然后在弹出的“选项”对话框中打开“控件”选项卡, 将“ActiveX控件”选项选中, 在“选定”列表框中列出了系统中所有的Activex控件, 选定Microsoft Slider Control选项, 然后单击“设置为默认值”按钮, 再单击“确定”按钮, 即可将此控件安装到“表单控件”工具栏上, 如图1.2所示。

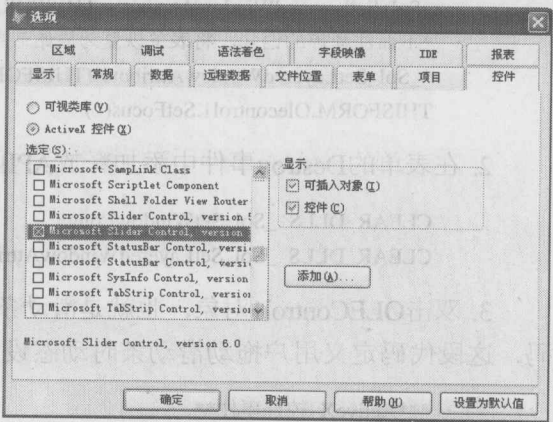


图1.2 安装ActiveX控件

3. 选定工具栏, 将ActiveX控件Microsoft Slider Control添加到窗体中, 然后在“属性查看器”中设置该对象的相应属性, 如表1.2所示。

表1.2 OLEControl1对象 (滑动条控件) 的属性设置

属性	属性值	功能描述
Orientation	0-水平	滑动条的方向为水平放置
Selstart	25	初始位置
Smallchange	1	滑动滑竿增加值的数值
Textposition	0-上端/左端	拖动滑竿时显示的提示信息位置
Tickfrequency	1	刻度出现的频率为一个刻度
Tickstyle	0-底端/右端	刻度出现的位置
Value	25	初始位置为25

程序源代码添加与解释

1. 双击表单, 在Init事件中添加对Windows API函数的引用:

```

DODEFAULT( )
#define BADOS_LOC "This sample only runs on Windows 2000 or higher."
*检测系统版本
IF VAL(OS(3))<5
```

```
MESSAGEBOX(BADOS_LOC)
RETURN .F.
ENDIF
*声明API函数
DECLARE SetWindowLong In Win32Api AS _Sol_SetWindowLong Integer, Integer, Integer
DECLARE SetLayeredWindowAttributes In Win32Api AS _Sol_SetLayeredWindowAttributes Integer, Integer, Integer, Integer
*将表单设置为扩展表单
_Sol_SetWindowLong(THISFORM.hWnd, -20, 0x00080000)
*设置表单属性，即透明度
_Sol_SetLayeredWindowAttributes(THISFORM.hWnd, 0, 255, 2)
*下面注释掉的语句，将表单恢复为原始表单
*_Sol_SetLayeredWindowAttributes(THISFORM.hWnd, 0, 0,0)
THISFORM.Olecontrol1.SetFocus( )
```

2. 在表单的Destroy事件中添加释放API函数动态链接库的代码：

```
CLEAR DLLS _Sol_SetWindowLong
CLEAR DLLS _Sol_SetLayeredWindowAttributes
```

3. 双击OLEControl1对象，也就是滑动条ActiveX控件对象，在Change属性中添加如下代码，这段代码定义用户拖动滑动条时动态设置窗体透明度的属性值：

```
***ActiveX 控件事件**
LOCAL lnValue
DO CASE
CASE THIS.Value = 0
lnValue = 0
CASE THIS.Value = 25
lnValue = 255
OTHERWISE
lnValue = THIS.Value * 10
ENDCASE
_Sol_SetLayeredWindowAttributes(THISFORM.hWnd, 0, lnValue, 2)
```

这段代码调用了在表单的Init事件中声明的API函数来定义窗体的透明度属性值。

4. 保存表单，运行程序，然后在表单中拖动滑动滑竿。

案例2 创建不规则窗体

案例运行效果与操作

越来越多的应用程序中使用到各种不同规则的窗体，这些不同规则的窗体给应用程序带来异常的情趣和不寻常的效果。本案例通过一组Windows API函数和位图实现了一个自定义的、可换肤的窗体效果，如图1.3所示。

程序运行后，可以通过单击窗体的“Close”按钮关闭窗口体。

制作要点

1. 自定义窗体的原理

自定义窗体实际上并非是完全舍弃原先的窗体,而是在原窗体的基础上通过视觉差产生的,因为操作系统不检测被设置为完全透明的窗体区域,因此我们只需要将原始窗体中不需要显示的部分定义为统一的颜色,然后通过API函数将这部分颜色设置为在表单中完全透明即可。

2. Windows API函数SetWindowLong

该函数用于将一个窗口指定为分层窗口,函数声明为:

Declare Integer SetWindowLong in User32 Long hwnd,Long nIndex,Long dwNewLong

其中的参数含义如下:

Hwnd: 窗口句柄,可以通过表单的属性Hwnd来获取;

nIndex: 表示要设置的窗口的属性项,窗口属性项如表1.3所示;

dwNewLong: 要为属性设置的新值。在此,我们需要将新值设置为WS_EX_LAYERED以扩展分层窗口,而该值定义的常量地址为0x00080000。

不规则窗体

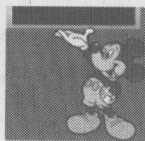


图1.3 由文字和图形组成的不规则窗体

表1.3 nIndex窗口属性

常量	属性值	属性功能
GWL_WNDPROC	-4	为窗口过程设置一个新的地址
GWL_HINSTANCE	-6	设置窗口的句柄
GWL_HWNDPARENT	-8	设置父窗口的句柄
GWL_ID	-12	设置窗口的标识符
GWL_STYLE	-16	设置窗口的风格
GWL_EXSTYLE	-20	设置窗口的扩展风格
GWL_USERDATA	-21	设置与窗口有关的32位值,每一个窗口均有一个由创建该窗口的应用程序使用的32位值
DWL_DLGPROC	4	设置对话框过程的新地址
DWL_MSGRESULT	0	设置在对话框过程中处理的消息返回值
DWL_USER	8	设置应用程序私有的额外信息,例如一个句柄或者指针

3. GetWindowLong函数

此函数用于取得指定窗口的标识符,函数声明为:

DECLARE INTEGER GetWindowLong IN user32.DLL;
INTEGER hwnd, INTEGER nIndex

其中的参数含义如下:

Hwnd: 窗口句柄,可以通过表单的属性Hwnd来获取;

nIndex: 表示要设置的窗口的属性项，窗口属性项如表1.3所示。

4. SetLayeredWindowAttributes函数

此函数用于设置分层窗口的透明度，也就是透明颜色的键值。在VFP中的声明为：

```
Declare Integer SetLayeredWindowAttributes in User32
Long hwnd,Long crKey,Long bAlpha,Long dwFlags
```

其中的参数含义如下：

Hwnd: 指定为分层窗口的窗口句柄；

crKey: 分层窗口中将要指定为透明的颜色值，所有这种颜色都将设置为透明，而且Windows消息不对设置为透明的部分进行消息检测，颜色值可以通过RGB()函数取得；

bAlpha: 要设置为透明的透明度，0~255之间；

dwFlags: 函数的方式。如果设置为1，则将参数crKey指定的颜色设置为透明；如果设置为2，将根据参数aAlpha的值来设置整个表单的透明度；如果设置为0，则还原原表单的属性，使表单不透明。本案例需要根据用户指定的透明度来设置表单透明，因此，应该将此参数设置为2。

5. ADir()函数的应用

如何搜索某一类型文件、取得某个文件的大小、修改日期及属性，VFT为我们提供了一个非常强大的函数ADIR()，该函数的语法如下：

```
ADIR(ArrayName [, cFileSkeleton [, cAttribute [, cCreatorType]]])
```

函数参数ArrayName为数组变量，用于保存取得的文件、文件属性等信息，参数cFileSkeleton为文件通配符，可以用于搜索某一类型的文件，参数cAttribute为指定文件的属性，参数cCreatorType为文件创建者的类型，通常为1。

例如：

```
X=Adir(Myml,'C:\TEMP\*.DBF')
```

返回值：

X是符合cFileSkeleton条件的文件或目录个数。

Myml是一个具有五列内容的数组。

这五列的含义是：

- 1 文件名 字符型
- 2 文件大小 数值型
- 3 文件日期 日期型
- 4 文件时间 字符型
- 5 文件属性 字符型

例如，我们想得到某个文件的大小或文件修改日期及属性：

```
X=Adir(Myml,'C:\TEMP\abc.DBF')
IF X#0
    wjsize=Myml(1,2) &&取得文件大小
    wjdate=Myml(1,3) &&取得修改日期及属性
    wjAttr=Myml(1,5) &&取得文件属性
```

```
ELSE  
wait wind '没有您指定的文件'  
ENDI
```

文件属性的含义:

- A 档案文件——可读写
- H 隐藏文件
- R 只读文件
- S 系统文件
- D 目录或文件夹

步骤详解

1. 新建一个表单，然后在表单中添加一个Label（标签）控件，在标签控件的下面添加一个Image（图形）控件，再添加一个按钮控件；表单及其包含的各个控件的属性设置如表1.4所示。

表1.4 表单各个控件的属性设置

控件名称	属性	属性值
Form1	AutoCenter	.T.
	Boderstyle	0-无边框
	Caption	none
	Closable	.F.
	ControlBox	.F.
	ShowWindow	2-作为顶层表单
	TitleBar	0-Off
Label1	Caption	“不规则窗体”
	FontSize	48
Image	AutoSize	true
	Picture	在程序中动态设置
Command1	Caption	“Close”
	BackColor	255,0,0（绝对不能与表单背景色一致，否则将无法显示）

2. 选择系统菜单“表单” ➤ “新建属性”，在打开的对话框中添加两个表单属性RunPath和nFlags，分别用于保存程序执行路径和表单原始标识符。

3. 选择系统菜单“表单” ➤ “新建方法”，在打开的对话框中添加一个自定义方法MakeIregularForm。

程序源代码添加与解释

1. 在表单的属性查看器中双击自定义的函数MakeIregularForm，然后在打开的代码编辑器中，添加如下代码。

参数说明如下：

- nHWND表示需要将哪个表单设置为不规则的表单。
- nColor表示要设置为透明的颜色。

• nAction表示要如何设置分层窗体的属性，0表示恢复为原始表单，1表示将nColor颜色设置为在表单中完全透明，2表示根据指定的透明度值来设置表单的透明度。

LPARAMETERS nHWnd, nColor, nAction

*定义SetLayeredWindowAttributs函数中需要用到的参数常量

#DEFINE LWA_COLORKEY 1

#DEFINE LWA_ALPHA 2

*定义SetWindowLong和GetWindowLong函数常量

#DEFINE GWL_EXSTYLE -20

#DEFINE WS_EX_LAYERED 0x00080000

LOCAL lnFlags

do case

case nAction = 1 &&将指定颜色设置为透明

lnFlags = GetWindowLong(nHWnd, GWL_EXSTYLE) &&取得指定窗体的标识符

thisform.nFlags = lnFlags &&为表单自定义属性赋值

lnFlags = BITOR(lnFlags, WS_EX_LAYERED)

&&将分层标识符添加到窗体标识符中

SetWindowLong(nHWnd, GWL_EXSTYLE, lnFlags) &&设置新的窗体标识符

*将把指定的nColor颜色设置为透明，如果设置为2，则按照第3个参数来设置分层窗口的透明度

SetLayeredWindowAttributs(nHWnd, nColor, 0, LWA_COLORKEY)

case nAction = 2 &&完全不透明

SetWindowLong(nHWnd, GWL_EXSTYLE, thisform.nFlags) &&恢复原始窗体标识符

*设置为完全不透明的表单

SetLayeredWindowAttributs(nHWnd, nColor, 0, 0)

endcase

2. 双击表单，在Init（初始化）事件中，添加如下代码，用于在程序开始执行时就显示自定义的窗口：

Thisform.RunPath = ADDBS(JUSTPATH(SYS(1271, thisform))) &&取得当前表单所在的路径

TRY

*声明函数

*设置分层窗口属性的函数

DECLARE INTEGER SetLayeredWindowAttributs IN win32api;

INTEGER hWnd, INTEGER crKey, INTEGER bAlpha, INTEGER dwFlags

*设置和取得窗口属性的函数

DECLARE INTEGER SetWindowLong IN user32.DLL ;

INTEGER hWnd, INTEGER nIndex, INTEGER dwNewLong

DECLARE INTEGER GetWindowLong IN user32.DLL ;

INTEGER hWnd, INTEGER nIndex

CATCH

MESSAGEBOX("Could not load the required libraries. Make sure you are running Windows 2000 or higher.")

ENDTRY

TRY

This.RunPath = ADDBS(JUSTPATH(SYS(1271, thisform)))

CD (This.RunPath)

ADIR(laPictures, "*.bmp", "", 1)