

商业开发代码库系列

Delphi

案例开发集锦

康祥顺 张小刚 刘亮 编著

黄显堂 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

Delphi案例开发集锦

康祥顺 张小刚 刘亮 编著
黄显堂 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实例介绍如何运用Delphi编程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、图像处理、多媒体应用、系统文件处理、硬件控制、API函数的应用、数据库基本处理、网络处理、网络与数据库高级应用、综合实例共11个章节。每个案例的编排都是严格按照读者的阅读习惯进行组织的,实用性强、解释详尽。随书附有光碟源代码,读者可以在实例中学习编程思想和编程技巧。

本书适用于大中专院校学生、程序设计人员和Delphi爱好者阅读。

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

Delphi案例开发集锦/康祥顺等编著. —北京:电子工业出版社, 2005.8

(商业开发代码库系列)

ISBN 7-121-01476-9

I. D… II. 康… III. 软件工具—程序设计 IV. TP311.56

中国版本图书馆CIP数据核字(2005)第069703号

责任编辑:徐云鹏

特约编辑:杨 荟

印 刷:北京天竺颖华印刷厂

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

经 销:各地新华书店

开 本:787×1092 1/16 印张:33.625 字数:860千字

印 次:2005年8月第1次印刷

定 价:50.00元(含光碟1张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:010-68279077。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。

前 言

你需要阅读本书吗

Delphi是Borland公司推出的优秀的前端开发工具。自Delphi问世以来,其友好的集成开发界面、可视化的双向开发模式、良好的数据库应用支持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Delphi技术的进步和发展,它必将成为今后几年中最佳的编程语言。

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,当再翻看别人的书或者使用别人的小软件时,才会问自己,原来这个函数还能这样用。这就是自己钻研所不能取得的成果,而且会事倍功半。本书汇集了Delphi编程语言所能解决的问题的所有经典实例,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,使你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、自我打击的心理负担。因此,如果你已经开始学习Delphi,具有了一定的Delphi语法基础,那么本书将是引领你进入真正的专家级聪明程序员的指南性图书,它将会使你的编程水平发生质的飞跃!

本书涵盖的内容以及编排格式

本书共11章,每一章都是由既相互独立,又相互间有一定逻辑关系的实例组成。每一个实例都分为5个部分编写,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来实现这个案例,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;
- ✓ **代码添加与解释:** 告诉读者如何实现此案例的功能,让读者理解代码的运用以及解决问题的思路;同时在添加的代码中,还会告诉读者代码的功能和解决的问题;
- ✓ **程序员代码与解释:** 附带了所有的该案例的源代码,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,以方便读者进行验证;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解释,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,你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;



- ✓学习如何在程序中处理多媒体编程的问题;
- ✓学会在程序中处理系统文件;
- ✓学会如何在程序中控制系统硬件和外围设备;
- ✓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和API函数处理网络操作;
- ✓学会如何在程序中处理复杂的跨平台的应用的问题;
- ✓学会如何应用Delphi提供的组件、函数、对象实际开发功能复杂的应用系统,包括单机应用系统、B/S结构的应用系统、C/S结构的应用系统,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。

本书配套光碟的使用

为了方便读者阅读、测试和使用本书提供的案例,本书附带了每个案例的源代码。源代码严格按照章节编排,例如第1章第一个案例的源代码存放位置为:源代码\1\程序启动画面,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WinXP+Delphi 7.0下经过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:案例工程文件打开后无法找到引用的文件、资源文件,这主要有两个方面的原因,一个是作者保存案例文件的位置与你的不同,另一个就是作者的机器名与你的设置不一致。如果出现这两个方面的问题,你可以遵照实现步骤自己制作案例程序,然后把光碟中的源代码复制到相应的程序部分,或者打开程序中的对象检查器,重新根据自己的机器设置相关属性。

但是,鱼和熊掌不能兼得。虽然方便了读者,但同时很可能也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,强烈建议读者先看看书中案例的运行效果,然后再看看源代码,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。

本书由康祥顺、张小刚、刘亮编写,全书由黄显堂审校。最后,感谢和我们共同完成此书的合作者,他们是王巍、张强、张音、李正非、张庆、崔竞、董养斌、保春艳、李欣、梅光耀、张小丽、康祥琴。感谢电子工业出版社北京美迪亚电子信息有限公司的各位老师,谢谢你们帮助和指导。

本书中所有实例,编者都在WinXP+Delphi 7.0的开发环境中进行了测试,由于水平有限,尽管做了严格的审核和测试,书中难免仍有一些错误,敬请广大读者不吝赐教,编者在此表示感谢。

目 录

第1章 界面设计	1
案例1 系统启动画面	1
案例2 可移动分割的窗体	4
案例3 Office风格的可融合窗体	6
案例4 程序进程条	10
案例5 限制窗体的大小	16
案例6 统一风格的窗体	18
案例7 具有背景花纹的窗体	21
案例8 “MICROSOFT”英文字形窗体	24
案例9 动态显示窗体	27
案例10 在列表框中加入图片	29
第2章 图像处理	34
案例1 捕捉并保存全屏和指定区域图像	34
案例2 透明的图像	44
案例3 移动的字幕和图像	51
案例4 图像的整体和局部放大	56
案例5 渐变彩色显示	65
案例6 动态多边形	75
案例7 图像镜像显示	78
案例8 图像漫游技术	85
案例9 图像的淡入淡出效果	89
案例10 图像上下左右卷动打开	94
案例11 图像的百叶窗效果	101
第3章 多媒体应用	105
案例1 多功能媒体播放器	105
案例2 屏幕保护程序	110
案例3 多媒体与字幕同步显示	117
案例4 在窗体中播放FLASH动画	124
案例5 旋转的字幕	127
案例6 自动抓取媒体瞬间图像	130
案例7 控制声音设备的声道和音量	136

目 录

第1章 界面设计	1
案例1 系统启动画面	1
案例2 可移动分割的窗体	4
案例3 Office风格的可融合窗体	6
案例4 程序进程条	10
案例5 限制窗体的大小	16
案例6 统一风格的窗体	18
案例7 具有背景花纹的窗体	21
案例8 “MICROSOFT”英文字形窗体	24
案例9 动态显示窗体	27
案例10 在列表框中加入图片	29
第2章 图像处理	34
案例1 捕捉并保存全屏和指定区域图像	34
案例2 透明的图像	44
案例3 移动的字幕和图像	51
案例4 图像的整体和局部放大	56
案例5 渐变彩色显示	65
案例6 动态多边形	75
案例7 图像镜像显示	78
案例8 图像漫游技术	85
案例9 图像的淡入淡出效果	89
案例10 图像上下左右卷动打开	94
案例11 图像的百叶窗效果	101
第3章 多媒体应用	105
案例1 多功能媒体播放器	105
案例2 屏幕保护程序	110
案例3 多媒体与字幕同步显示	117
案例4 在窗体中播放FLASH动画	124
案例5 旋转的字幕	127
案例6 自动抓取媒体瞬间图像	130
案例7 控制声音设备的声道和音量	136



案例8	WAV声音文件的循环播放	143
案例9	可爱、聪明的小人动画	146
第4章 系统文件处理		153
案例1	FSO模型的应用	153
案例2	获取文件或文件夹属性	160
案例3	显示驱动器中的所有目录	167
案例4	实现目录遍历	171
案例5	读写文本文件和二进制文件	174
案例6	访问注册表	179
案例7	用Delphi实现文件搜索	184
案例8	模拟关闭系统	191
案例9	文本文件的加密与解密	195
案例10	利用INI文件给程序创建动态菜单	201
第5章 硬件的控制		206
案例1	限制鼠标的活动区域	206
案例2	检测是否安装了打印机	209
案例3	打开和关闭系统声道	211
案例4	测试光驱中是否有光碟	214
第6章 API函数的应用		218
案例1	取得系统内存信息	218
案例2	设置和获取屏幕分辨率	221
案例3	获取和设置计算机名	224
案例4	获取和设置系统时间	227
案例5	获取Windows系统版本信息	231
案例6	隐藏和显示Windows任务栏	235
案例7	设置输入法	238
案例8	在任务栏上隐藏和显示应用程序窗口	242
案例9	启动资源管理器打开目录	245
案例10	添加自己的系统状态栏图标	248
第7章 数据库应用		252
案例1	自动登录数据源	252
案例2	ADO控件编程	254
案例3	单记录浏览数据	256
案例4	编程实现记录编辑	258

案例5 多个数据库间的关联查询	265
案例6 在SQL数据库中存取图形数据	267
案例7 用TREEVIEW显示数据库记录	272
案例8 动态创建数据库和数据表	277
案例9 多条件查询系统的构建	283
案例10 DBChart统计图表	290
案例11 创建动态数据报表	296
 第8章 网络应用	 303
案例1 获取自己的IP地址	303
案例2 在窗口中Ping一个IP地址	305
案例3 显示指定工作组内的所有计算机	313
案例4 发送E-mail电子邮件	318
案例5 迷你Web浏览器	322
案例6 简易网络聊天室	330
 第9章 网络与数据库应用实例	 336
案例1 网站音乐Web Services服务指南	336
案例2 应用Web Services集成多层异构系统	343
案例3 高效数据查询分布式应用系统	352
案例4 IntraWeb应用举例——NBA球队实力在线调查	357
案例5 自动新闻编辑器	365
案例6 WebSnap应用举例——文件上传系统	371
 第10章 字符串处理	 378
案例1 中英文字数统计	378
案例2 数值与字符串的相互转换	381
案例3 将小写金额转换为大写金额	385
 第11章 综合实例	 390
案例1 学生信息管理系统	390
案例2 小区物业管理系统 (C/S)	419
案例3 Internet网上图书征询系统 (B/S)	489



第1章

界面设计



本章内容

- ◆ 系统启动画面
- ◆ 可移动的分割窗体
- ◆ Office风格的可融合窗体
- ◆ 程序进程条
- ◆ 限制窗体的大小
- ◆ 统一风格的窗体
- ◆ 具有背景花纹的窗体
- ◆ “MICROSOFT”英文字形窗体
- ◆ 在窗体中播放Flash动画
- ◆ 在列表框中加入图片



案例1 系统启动画面



案例运行效果与操作

本案例主要是针对大型的应用系统启动时，比如 Delphi 7.0，由于加载的文件比较多，启动需要相当长的时间，这时为了提醒用户系统正在启动，而没有出现故障，可以在启动的过程中显示一个启动画面，在系统所有文件加载完后，再释放该画面，显示系统窗口。程序运行效果如图1-1所示。

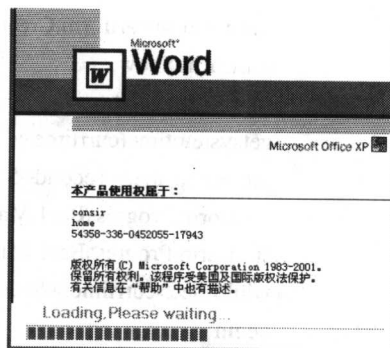


图1-1 程序启动画面



制作要点

1. TImage组件;
2. TProgress组件;
3. TForm的OnCreate事件;
4. GetSystemTime函数。





步骤详解

1. 在Delphi环境下, 选择菜单File>New>Other, 新建一个Application类型的普通工程, 并将系统自动创建的窗体命名为MainForm。

2. 再新建一个窗体TForm, 取名Startform, 在属性检查器中, 将startform窗体的borderstyle属性设置为bsnone, 以便启动窗体时没有边框和标题栏, 并将窗体的FormStyle设置为fsStayOnTop, 使之始终显示在顶层。

3. 在startform窗体中分别拖放一个TImage组件和一个TProgress组件, 另外再拖放一个TLabel组件。其中, TImage组件用于显示启动时的图形画面, TProgress组件用于显示系统启动的进度, TLabel组件用于显示一条提示消息, caption属性为“loading, please waiting...”。界面设置结果如图1-1所示。

4. 打开主窗体单元, 选择菜单file>uses unit, 在主窗体单元中引用启动窗体单元, 以便于稍后在主窗体中引用启动单元的对象。

5. 选择系统菜单project>options, 在弹出的project options for project1工程选项对话框中, 将startform从auto-create forms对话框中移动到avaiabel forms对话框, 这样, 启动窗体不会自动创建并显示。



代码添加与解释

6. 双击主窗体MainForm, 在OnCreate事件中添加如下代码:

```
procedure TMainForm.FormCreate(Sender: TObject);
var startform:tstartform;           //声明窗体变量
    currtime:systemtime;           //当前系统时间
    sec:word;
begin
    startform:=tstartform.Create(self); //创建启动窗体对象实例
    startform.Show;                  //显示启动窗体
    startform.Update;                //更新窗体
    getsystemtime(currtime);         //取得系统时间, 并保存在currtime变量中
    sec:=currtime.wSecond+5;
    startform.ProgressBar1.Max:=sec;
    startform.ProgressBar1.Min:=0;
    while sec>currtime.wSecond do    //为了效果演示, 进行延时
    begin
        getsystemtime(currtime);
        startform.ProgressBar1.Position:=currtime.wSecond; //设置进度条的状态位置
    end;
    startform.Free;                  //系统完全启动后, 释放启动画面
end;
```



程序源代码与解释

主窗体程序代码:

```
unit Unit2;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;
type
  TMainForm = class(TForm)
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  MainForm: TMainForm;

implementation

uses Unit1;                                //引用启动窗体单元
```

```
{ $R *.dfm }
```

//在系统主窗体启动时, 首先触发窗口的OnCreate事件或者OnActivate事件, 在这两个事件中, 主要进行系统初始化。由于没有触发窗口的OnShow事件, 主窗体并没有显示, 因此, 我们可以在OnCreate事件中创建并显示启动画面, 当初始化完成后, 再将启动窗口释放。

```
procedure TMainForm.FormCreate(Sender: TObject);
var startform:tstartform;                    //声明窗体变量
    currtime:tsystemtime;                    //当前系统时间
    sec:word;
begin
  startform:=tstartform.Create(self);        //创建启动窗体对象实例
  startform.Show;                             //显示启动窗体
  startform.Update;                           //更新窗体
  getsystemtime(currtime);                    //取得系统时间, 并保存在currtime变量中
  sec:=currtime.wSecond+5;
  startform.ProgressBar1.Max:=sec;
  startform.ProgressBar1.Min:=0;
  while sec>currtime.wSecond do              //为了效果演示, 进行延时
  begin
    getsystemtime(currtime);
```



```

startform.ProgressBar1.Position:=curtime.wSecond; //设置进度条的状态位置
end;
startform.Free; //系统完全启动后，释放启动画面
end;
end.

```



案例2 可移动分割的窗体



案例运行效果与操作

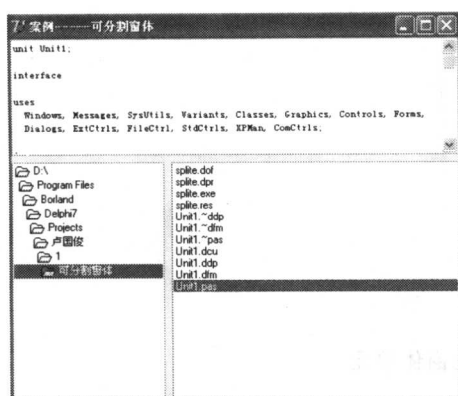


图1-2 可移动分割的窗体

为了在有限的窗体界面中进行合理的布局，程序中经常使用可分割的窗体，在用户操作时，可以通过拖动窗口分割线来改变窗口的大小。本案例设计了一个文本文件预览程序，窗体左下方用于显示本地文件夹，双击文件夹，就可以在右下方显示该文件夹中包含的文件，然后，单击一个文件就可以在窗体上部的MEMO文本框中显示该文本文件的详细内容。为了便于用户操作，我们使用了可分割的窗体，通过拖动窗体中的分割线，可以将三个窗口分别进行上下、左右拖动，以此改变窗口的显示空间。运行结果如图1-2所示。



制作要点

1. TSplitter分割组件；
2. TDirectoryListBox组件；
3. TFileListBox组件；
4. TRichEdit组件；
5. TFileListBox组件的OnClick事件。



步骤详解

1. 选择系统菜单file>new>other，新建一个application普通工程。
2. 在系统自动创建的窗体中拖放组件。从组件面板Win32中拖放一个TRichEdit组件到窗口中，该组件主要用于显示有格式的文本，关键是将该组件的align属性设置为alTop，使之占满窗体上部区域。
3. 从Additional面板上拖放一个TSplitter组件到窗口中，该组件主要用于显示分割线，将该组件的align属性设置为alTop，使之紧挨着TRichEdit组件，成上下分割。
4. 从Win3.1组件面板上拖放一个TDirectoryListBox组件到窗口中，该组件用于显示驱动器中的文件夹，并将align属性设置为alLeft，使之占满窗体左边剩下的区域。
5. 在从Additional面板上拖放一个TSplitter组件到窗口中，将align属性设置为alLeft，使

之紧挨着TDirectoryListBox组件，成左右分割。

6. 然后在从Win3.1面板上拖放一个TFileListBox组件到窗口中，该组件用于显示文件夹中的文件，将align属性设置为alClient，使之填满窗体剩下的整个区域。最后将TDirectoryListBox组件的FileList属性设置为TFileListBox，以便将TDirectoryListBox组件与TFileListBox组件关联。窗体设置结果如图1-2所示。



代码添加与解释

7. 选择TFileListBox对象，双击OnClick事件，添加如下代码：

```
procedure TForm1.FileListBox1Click(Sender: TObject);
var filename:string;
begin
    filename:=self.FileListBox1.FileName;           //取得用户在filelistbox列表框中单击时的文件
    self.RichEdit1.Lines.Clear;                     //清除richedit组件中的文本
    try
        self.RichEdit1.Lines.LoadFromFile(filename); //将文件内容加载到richedit格式文本框中
    except
        on e:exception do
            self.RichEdit1.Lines.Add('应用程序出现错误: '+e.Message); //显示异常消息
        end;
    end;
end;
```



程序源代码与解释

```
unit Unit1;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, FileCtrl, StdCtrls, XPMAN, ComCtrls;
type
    TForm1 = class(TForm)
        DirectoryListBox1: TDirectoryListBox;
        Splitter1: TSplitter;
        FileListBox1: TFileListBox;
        Splitter2: TSplitter;
        RichEdit1: TRichEdit;
        procedure FileListBox1Click(Sender: TObject);
        procedure FileListBox1Change(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
```



```
end;

var
    Form1: TForm1;

implementation
{$R *.dfm}

//在richedit组件中显示用户所单击的文本文件的内容
procedure TForm1.FileListBox1Click(Sender: TObject);
var filename:string;
begin
    filename:=self.FileListBox1.FileName;           //取得用户在filelistbox列表框中单击时的文件
    self.RichEdit1.Lines.Clear;                     //清除richedit组件中的文本
    try
        self.RichEdit1.Lines.LoadFromFile(filename); //将文件内容加载到richedit格式文本框中
    except
        on e:exception do
            self.RichEdit1.Lines.Add('应用程序出现错误: '+e.Message); //显示异常消息
        end;
    end;
end;
```

说明：很多组件都可以用于创建可分割的窗口，问题的关键是splitter组件和相应组件的align属性的设置。一般来说，应该首先添加分割窗口，然后再添加分割线，最后再添加另一个分割窗口，分割窗口可以上下分割，也可以左右分割。



案例3 Office风格的可融合窗体



案例运行效果与操作

可融合技术原意为“docking”，即对接、融合的意思，利用docking技术可以将一个对象移动到另一个对象中，进而与另一个对象融合在一起，如Office软件的工具栏，既可以固定在窗口上部的适当位置，又可以单独成为一个窗口。本案例仍然是一个文件浏览器，与上一个应用程序的区别就是，三个窗口不再是通过拖动分割线来改变窗体的大小，而是每一个窗口都可以拖放出来，形成一个独立的窗口，然后剩下的窗口将填满窗体的整个区域。当不再使用该窗口时，还可以拖放到原来的位置，嵌入到原窗体中。在窗口拖放过程中，还将在窗口状态栏中显示操作的状态信息。程序首次执行的效果如图1-3和图1-4所示。

拖动richedit文本窗口成为一个单独窗体后的效果。



制作要点

1. TDirctoryListBox组件;
2. TFileListBox组件;
3. TRichEdit组件;
4. TPanel组件;

5. TStatusBar组件;
6. OnStartDock事件;
7. OnEndDock事件。

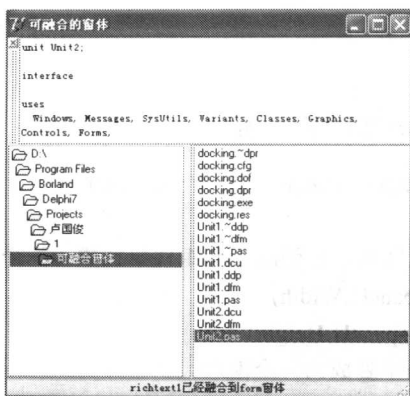


图1-3 可融合窗体

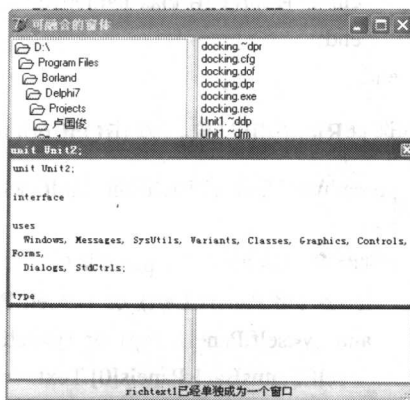


图1-4 单独成为一个窗体



步骤详解

1. 选择系统菜单file>new>other, 新建一个application普通工程。
2. 从Standard组件面板上拖放一个TPanel组件到窗体中, 该面板主要用于停放拖动的richedit窗体。将align属性设置为alTop, docksite属性设置为true, 设置成为可融合对象, 并将usedockmanager属性设置为true。
3. 从组件面板Win32中拖放一个TRichEdit组件到窗口中, 该组件主要用于显示有格式的文本, 关键是将该组件的align属性设置为alTop, 使之占满窗体上部区域, 并将dragkind属性设置为dkDock, dragmode设置为dmAutomatic。
4. 从Win3.1组件面板上拖放一个TDirectoryListBox组件到窗口中, 该组件用于显示驱动器中的文件夹, 并将align属性设置为alLeft, 使之占满窗体左边剩下的区域。
5. 然后在从Win3.1面板上拖放一个TFileListBox组件到窗口中, 该组件用于显示文件夹中的文件, 将align属性设置为alClient, 使之填满窗体剩下的整个区域。最后将TDirectoryListBox组件的FileList属性设置为TFileListBox, 以便将其与TFileListBox组件关联。窗体设置结果如图1-3和图1-4所示。



代码添加与解释

6. 选择FileListBox1组件, 双击OnClick事件, 添加如下代码:

```
procedure TForm1.FileListBox1Click(Sender: TObject);
begin
    self.RichEdit1.Clear;
    try
        self.RichEdit1.Lines.LoadFromFile(filelistbox1.FileName);
    end;
end;
```



```

except
on e:exception do
self.RichEdit1.Lines.Add('error:'+e.Message+
chr(13)+chr(10)+'classname:'+e.ClassName);
end;
end;

```

7. 选择RichEdit对象，双击OnEndDock事件，添加如下代码：

```

procedure TForm1.RichEdit1EndDock(Sender, Target: TObject; X, Y: Integer);
begin
//判断窗口是否超出了panel面板，如果超出面板区域，证明richtext组件单独成为一个窗体
if (x<self.Panel1.Left) or (x>self.Panel1.Left+panel1.Width)
and (y<self.Panel1.Top) or (y>self.Panel1.Top+panel1.Height) then
self.StatusBar1.Panels[0].Text:='richtext1已经单独成为一个窗口'
else
self.StatusBar1.Panels[0].Text:='richtext1已经融合到form窗体';
end;

```

8. 双击RichEdit对象的OnStartDock事件，添加如下代码：

```

procedure TForm1.RichEdit1StartDock(Sender: TObject;
var DragObject: TDragDockObject);
begin
self.StatusBar1.Panels[0].Text:='正在拖动richtext使之成为单独的窗体';
self.Panel1.Visible:=not self.Panel1.Visible; //显示或者隐藏面板
self.Refresh;
end;

```



程序源代码与解释

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, XPMan, StdCtrls, ExtCtrls, Buttons, ToolWin, FileCtrl;
type
  TForm1 = class(TForm)
    XPManifest1: TXPManifest;
    StatusBar1: TStatusBar;
    Panel2: TPanel;
    DirectoryListBox1: TDirectoryListBox;
    FileListBox1: TFileListBox;
    Panel1: TPanel;
    RichEdit1: TRichEdit;

```