



DELPHI
编程热点技术系列丛书



Delphi 7

图形图像多媒体 高级控件开发与应用

胡争辉 编著

1. 本书主要通过26个实例进行讲解，每个实例基本是按“编写控件+编写测试控件+结果分析”的模式，结构清晰，实例丰富而实用、注重实践
2. 配套光盘中包含了书中所有实例的源程序代码和图片，分类详细、结构清晰、一目了然
3. 适合于Delphi各层次的读者，也可作为计算机图形学的辅导用书

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



Delphi 7 图形图像多媒体 高级控件开发与应用

胡争辉 编著

中国铁道出版社

2003 • 北京

· (京) 新登字 063 号

内 容 简 介

本书通过大量实例说明了 Delphi 7 多媒体控件的使用, 主要内容有: 常用的和复杂的图形图像处理方式; 图形图像效果与鼠标相结合的处理方式; 编写增强的可视化控件; 开发多媒体控件; 以图形图像控件为基础, 开发应用程序的思路与方法。

本书配套光盘中, 提供了书中的所有源程序, 读者可以直接使用。

本书不仅介绍了 Delphi 的图形图像开发技术, 还提供了控件开发与应用程序设计分析思路, 适用于 Delphi 各层次的读者, 也可作为计算机图形学的辅导用书。

图书在版编目 (CIP) 数据

Delphi7 图形图像多媒体高级控件开发与应用/胡争辉编著. —北京: 中国铁道出版社, 2003. 9

(Delphi 编程热点技术系列丛书)

ISBN 7-113-05491-9

I. D… II. 胡… III. 软件工具-程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2003) 第 082283 号

书 名: Delphi7 图形图像多媒体高级控件开发与应用

作 者: 胡争辉

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 郭毅鹏 翟玉峰

责任编辑: 苏 茜 夏华香

封面设计: 孙天昭

印 刷: 北京鑫正大印刷有限公司

开 本: 787×1092 1/16 印张: 21 字数: 509 千

版 本: 2003 年 10 月第 1 版 2003 年 10 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05491-9/TP·1034

定 价: 36.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

由于 Delphi 7 在数据库与网络方面的强大功能,以致长期以来,它在其他方面的功能被忽视了。随着科学技术的进步,人们对于程序的要求也在不断提高,现在人们对于应用程序不再满足于功能,也不再喜欢呆板的界面,更希望应用程序的美观大方。诚然,应用程序不是艺术品,不能像艺术品一样在界面上精雕细刻,但是也不能仅仅满足于操作系统所提供的简单的操作控件,此时就需要图形图像多媒体控件来满足这样的要求。运用图形图像多媒体控件,一方面可以美化界面,满足用户对于应用程序美观大方的要求;另一方面可重用的图形图像控件可以多次使用,提高了开发效率。本书就是在这样的主导思想下完成的。

本书主要内容

本书通过实例讲解了 Delphi 7 开发图形图像多媒体控件的方法,主要内容如下:

为了使初级读者在学习控件开发之前对 Delphi 7 中控件的特性有一个直观的认识,第 1 章用简短的篇幅讲解了图形控件的应用。

然后再讲解图形图像多媒体控件的开发。在图形图像实例的选择方面,本书选取了比较复杂的标签控件为例。标签控件与其他控件不同,对于不包含文字的图形图像控件,实际上不需要编写程序,只需要使用图片即可,而标签上的文字可以随意设置,这就使得在标签上进行图形图像处理就必须设计出一种通用的图形图像处理方式,而这种能满足文字复杂多变的图形图像处理方式,在处理不含文字的控件时将更加轻而易举。基于这个思路在第 2 章中以标签为例,讲解了 8 种常用的图形图像处理方式以及将图形图像效果与鼠标相结合的处理方式。

接下来,第 3 章在第 2 章的基础上,讲解了 5 种复杂的图形图像处理方式。通过第 2 章与第 3 章的学习,读者不仅可以掌握基本的图形图像处理方式在 Delphi 7 中的实现,更可以学习到通过 Delphi 7 进行图形图像效果的设计思路。

在 Delphi 7 中不仅可以实现通用的图形图像效果,结合 Delphi 7 在 Windows 编程与窗体界面编程的特性,还可以对 Delphi 7 中已有的控件进行增强,在第 4 章中通过 5 个实例讲解如何编写增强的可视化控件。

第 5 章主要讲解了利用 Delphi 7 开发多媒体控件的实例,在简要讲解了如何在 Delphi 7 中应用多媒体控件之后,用 3 个不同方面的实例讲解了 Delphi 7 开发多媒体控件的方法。

本书的最后,在第 6 章中通过两个游戏实例进解以图形图像控件为基础开发应用程序的思路与方法,这两个实例均按照软件工程的思路讲解,使读者可以了解到图形图像控件在应用程序中的作用,并学习到将图形图像控件从程序的复杂逻辑中分离出来的技巧。

本书适用的读者

对于初级读者,通过本书可以学习到 Delphi 7 开发的基本知识,为深入学习 Delphi 7 打下基础。对于中级读者,通过本书可以学习到 Delphi 7 在图形图像多媒体开发方面的技术。对于高级读者,通过本书可以掌握 Delphi 7 在控件开发与应用程序分析上的思路。同时,本书也可以作为计算机图形学的辅导用书。

本书配套光盘

本书配套光盘中，包括本书的全部源代码，读者可以直接使用。

作者简介

本书由胡争辉主笔，该作者曾长期从事计算机开发和管理工作，并编写过数本计算机图书。本书的面世还得到了很多朋友的帮助，陈河南老师负责本书的统稿和组织策划工作，贺军、贺民、龚亚萍、李志云、戴军、肖迎、徐江、陈安南、李晓春、谢高联、李志伟、王巧红、陈萍、王学农、陈雷、陈志刚、樊鹏、王锋、周里文、吴少波、孔艳、杨颖、陈安华等人在预读、查错、实例测试、代码调试和教学试验等工作中付出了很多心血。还有陈贤淑、廖康良、陈晓娟等参与了本书的编排工作，在此一并致谢！

联系我们

读者在使用本书的过程中，若发现有疏漏和错误之处，或者，如果有好的想法和建议，均请与我们联系，电子邮件地址为：xiaoxiang-007@sohu.com。同时，我们也会在适当的时间对本书的内容进行修订和补充，并发布在天勤网站：<http://www.tqbooks.net>“图书修订”栏目中。

编者
2003年9月

目 录

第 1 章 图形控件的应用	1
1-1 绘制界面	2
1-2 编写程序	3
1-3 运行效果	4
第 2 章 增强标签	7
2-1 程序设计实例 1: 标签的发光和鼠标响应	8
2-1-1 发光效果	8
2-1-2 快捷键	13
2-1-3 响应鼠标事件	15
2-2 程序设计实例 2: 旋转标签	17
2-2-1 分析 VCL 库中的绘图过程	17
2-2-2 编写程序	18
2-3 程序设计实例 3: 阴文与阳文	24
2-3-1 使用 Enabled 属性	25
2-3-2 专用的阴文阳文属性	27
2-4 程序设计实例 4: 闪烁的标签	29
2-5 程序设计实例 5: 鼠标事件	35
2-6 程序设计实例 6: 突起和凹陷效果	40
2-6-1 编写控件	40
2-6-2 编写测试程序	45
2-6-3 结果分析	46
2-7 程序设计实例 7: 阴影标签	50
2-7-1 编写控件	51
2-7-2 编写测试程序	54
2-7-3 结果分析	55
2-8 程序设计实例 8: 立体效果	58
2-8-1 编写控件	58
2-8-2 编写测试程序	62
2-8-3 结果分析	63
2-9 程序设计实例 9: 渐变填充	66
2-9-1 编写控件	67
2-9-2 编写测试程序	72
2-9-3 结果分析	73

第 3 章 高级标签	79
3-1 程序设计实例 10: 按钮标签	80
3-2 程序设计实例 11: 垂直标签	86
3-3 程序设计实例 12: 阴文与阳文的配色方案	97
3-4 程序设计实例 13: 突起和凹陷的渐变效果	109
3-5 程序设计实例 14: 半透明标签	121
3-5-1 基本的半透明效果	121
3-5-2 失效的半透明标签	129
3-5-3 带背景的半透明标签	134
第 4 章 增强可视化控件	141
4-1 程序设计实例 15: 字体列表框	142
4-2 程序设计实例 16: 屏幕的复制	144
4-3 程序设计实例 17: 时钟	148
4-4 程序设计实例 18: 透明图片	155
4-5 程序设计实例 19: 阴影按钮	165
第 5 章 多媒体	175
5-1 程序设计实例 20: 多媒体播放器	176
5-2 程序设计实例 21: Flash 播放器	178
5-3 程序设计实例 22: 音量控制	181
5-3-1 总音量控制	182
5-3-2 单声道音量控制	184
5-3-3 平衡音量控制	190
5-3-4 多设备音量控制	196
5-4 程序设计实例 23: MP3	199
5-4-1 MP3 浏览器	199
5-4-2 修改 MP3 信息	207
5-5 程序设计实例 24: 播放声音文件	223
第 6 章 综合实例	227
6-1 程序设计实例 25: 二十一点游戏	228
6-1-1 需求分析	228
6-1-2 编写游戏操作者	230
6-1-3 编写游戏操作者——人	234
6-1-4 编写游戏操作者——计算机	235
6-1-5 编写游戏控制	237
6-1-6 编写操作界面	243
6-1-7 编写游戏操作者的信息界面	247
6-1-8 完成主程序	251
6-1-9 初始化程序	254

6-1-10	运行程序	265
6-1-11	结束程序	270
6-1-12	查看完整的牌面	270
6-1-13	自定义绘制牌面	273
6-2	程序设计实例 26: 猜牌面游戏	278
6-2-1	需求分析	278
6-2-2	编写回答程序	279
6-2-3	编写备选牌界面	283
6-2-4	目标牌界面	287
6-2-5	绘制主程序界面	293
6-2-6	测试操作效果	294
6-2-7	游戏控制的需求分析	305
6-2-8	编写游戏控制	306
6-2-9	编写游戏操作	312
6-2-10	编写游戏结果界面	314
6-2-11	完成游戏的主程序	317
6-2-12	运行游戏	319

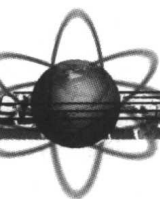
1

Chapter

图形控件的应用

本章重点:

在本章中, 将以一个图形浏览器为例, 讲解 Delphi 7 中提供的图形控件的应用。



1-1 绘制界面

首先创建一个应用程序, 设定窗体的 Caption 属性值为“图片浏览器”, Name 属性值为 FormPicView。在窗体上绘制两个  面板控件 (位于 Insert 面板上的 Standard 选项卡) 和一个  分隔条控件 (位于 Insert 面板上的 Additional 选项卡), 第一个面板控件的 Align 属性值为 alLeft, Name 属性为 Panel1, 第二个面板控件的 Align 属性值为 alClient, Name 属性为 Panel2, 分隔条控件将自动出现在两个面板的中间。

在 Panel1 上绘制一个  驱动器组合框 (位于 Insert 面板的 Win3.1 选项卡上), 设定 Anchors 属性集中 akLeft、akRight、akTop 属性值为 true, akBottom 属性值为 false, Name 属性为 DriveComboBox。

在 Panel1 上绘制一个  目录列表框 (位于 Insert 面板的 Win3.1 选项卡上), 设定 Anchors 属性集中全部为 true, Name 属性为 DirectoryListBox。设定 DriveComboBox 控件的 DirList 属性值为 DirectoryListBox 把驱动器列表控件和目录列表控件关联起来。在 Panel2 上绘制一个面板, 设定面板的 Align 属性为 alTop, Name 属性为 Panel3。

在 Panel3 上绘制一个  文件类型组合框 (位于 Insert 面板的 Win3.1 选项卡上), 设定 Anchors 属性集中, akLeft、akTop、akRight 三个属性值为 true, akBottom 属性值为 false, 对于 Filter 属性按如图 1-1 所示进行设定。

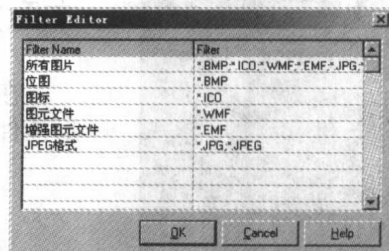


图 1-1 设定文件类型

其中所有图片所对应的值为“*.BMP; *.ICO; *.WMF; *.EMF; *.JPG; *.JPEG”, 设定 Name 属性为 FilterComboBox。

在文件类型组合框的右侧绘制一个  复选框 (位于 Insert 面板的 Standard 选项卡上), 设定 Anchors 属性集的 akTop 和 akRight 属性值为 true, akLeft 和 akBottom 属性值为 false。Caption 属性值为“自动缩放”, Name 属性值为 CheckBoxStretch。

在复选框的右侧绘制一个  按钮 (位于 Insert 面板的 Standard 选项卡上), 设定 Anchors 属性集的 akRight 和 akTop 属性值为 true, akLeft 和 akBottom 属性值为 false。Caption 属性为“全屏查看”, Name 属性值为 ButtonFull。

在 Panel3 的下侧绘制一个  文件列表框 (位于 Insert 面板的 Win3.1 选项卡), 设定 Anchors 属性集中四个属性都为 true, Name 属性值为 FileListBox。设定 DirectoryListBox 的 FileList 属性值为 FileListBox。FilterComboBox 的 FileList 属性值为 FileListBox。在 Panel2 上绘制一个分隔条, 设定 Align 属性值为 alTop。

在 Panel2 上绘制一个  图像控件 (位于 Insert 面板的 Additional 选项卡上)。设定 Align 属性值为 alClient, Name 属性值为 Image。对象之间的包含关系, 如图 1-2 所示。窗体的设计效果如图 1-3 所示。

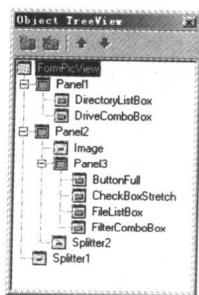


图 1-2 对象之间的包含关系

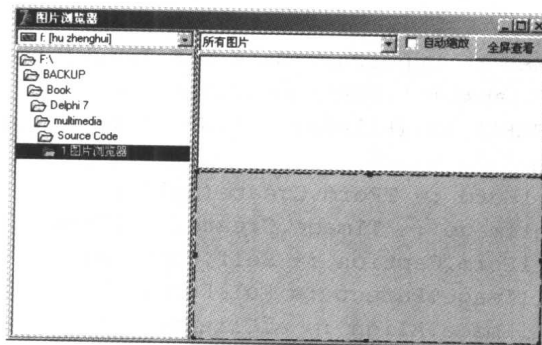


图 1-3 窗体的设计效果

1-2 编写程序

下面编写程序，使用如下所示的代码设定单元名。

```
unit UnitPicView;
```

增加对于 FileListBox 的 OnChange 事件的响应过程，代码如下所示：

```
procedure TFormPicView.FileListBoxChange(Sender: TObject);
var
    FileExt : String;
begin
    FileExt := UpperCase(ExtractFileExt(FileListbox.FileName));
    if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or (FileExt =
        '.EMF') or (FileExt = '.JPG') or (FileExt = '.JPEG') then
    begin
        Image.Picture.LoadFromFile(FileListbox.FileName);
        Self.Caption := '图片浏览器 - ' + ExtractFileName(FileListBox.FileName) +
            Format('(%dX%d)', [Image.Picture.Width, Image.Picture.Height]);
        if FileExt = '.ICO' then
        begin
            Self.Icon := Image.Picture.Icon;
        end;
    end;
end;
```

双击“自动缩放”复选框，增加如下所示的过程：

```
procedure TFormPicView.CheckBoxStretchClick(Sender: TObject);
begin
    Image.Stretch := CheckBoxStretch.Checked;
end;
```

双击“全屏查看”按钮，增加如下所示的过程：

```
procedure TFormPicView.ButtonFullClick(Sender: TObject);
var
    FullForm : TForm;
    FullImage : TImage;
    FileExt : String;
```



```
begin
  FileExt := UpperCase(ExtractFileExt(FileListbox.FileName));
  if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or (FileExt =
    = '.EMF') or (FileExt = '.JPG') or (FileExt = '.JPEG') then
  begin
    FullForm := TForm.Create(Self);
    FullImage := TImage.Create(FullForm);
    FullForm.Caption := Self.Caption;
    FullImage.Parent := FullForm;
    FullImage.Align := alClient;
    FullImage.Picture.LoadFromFile(FileListbox.FileName);
    FullForm.WindowState := wsMaximized;
    FullForm.Visible := true;
  end;
end;
```

1-3 运行效果

下面运行程序，效果如图 1-4 所示。

此时鼠标可以拖动分隔条改变窗体的布局。使用驱动器组合框，文件夹列表框和文件类型组合框可以选择目录和文件类型，由于这几个控件之间使用属性设置了关联，因此不需要进行编码，下面选择一个包含图片的目录，效果如图 1-5 所示。

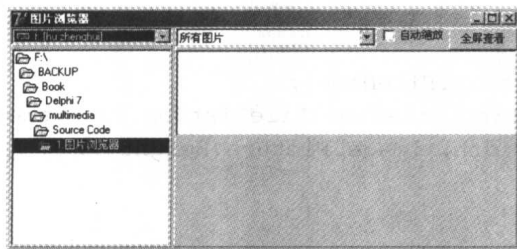


图 1-4 运行时效果

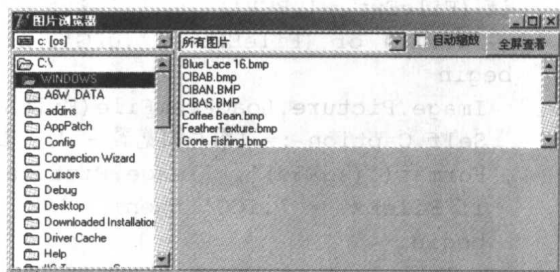


图 1-5 包含图片的目录

接下来在文件列表中选择其中的一个图片，触发 FileListBox 对象的 OnChange 事件。首先取得选中文件的文件名的扩展名，并转化为大写字母。

```
FileExt := UpperCase(ExtractFileExt(FileListbox.FileName));
```



注意

其中 ExtractFileExt 函数用于从文件的完整路径中提取出文件的扩展名，这个扩展名包含扩展名之前的点号“.”。

然后根据扩展名，判断是否为可以处理的文件类型。在 Delphi 7 中可以对 BMP、ICO、WMF、EMF、JPG 和 JPEG 文件进行处理。

```
if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or (FileExt =
  = '.EMF') or (FileExt = '.JPG') or (FileExt = '.JPEG') then
```

如果扩展名符合要求,就把文件读入图像控件中。

```
Image.Picture.LoadFromFile(FileListbox.FileName);
```

然后在窗体的标题中显示文件名以及文件的尺寸。

```
Self.Caption := 图片浏览器 - ' + ExtractFileName(FileListBox.FileName) +  
Format('%dX%d', [Image.Picture.Width, Image.Picture.Height]);
```



说明

Format 语句用于按照格式生成字符串,第一个参数是格式字符串,每二个参数是用集合的形式写成的参数列表。

接下来判断文件的扩展名是否为 ICO,也就是图标文件。

```
if FileExt = '.ICO' then
```

如果读取的是图标文件的话,那么就设定窗体的图标为选中的图标。

```
Self.Icon := Image.Picture.Icon;
```



说明

当 Picture 读取图标文件时,将自动把图标文件读取到 Icon 属性中,可以直接使用。

此时窗体的效果如图 1-6 所示。

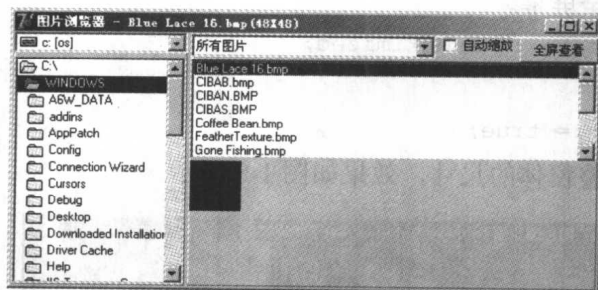


图 1-6 显示图片

可以看到窗体中显示出所选择的图片。此时图片的尺寸为原始尺寸,下面选择“自动播放”复选框,程序将设定图片的自动缩放属性和复选框的值相同。

```
Image.Stretch := CheckBoxStretch.Checked;
```

在选择后,复选框的值变为真,所以图像控件设置为自动缩放,效果如图 1-7 所示。



图 1-7 自动缩放图片



对于较小的图片可以使用自动缩放功能显示,至于较大的图片,将不能完整地显示在图片框中,此时可以单击“全屏查看”按钮。程序首先获取文件的扩展名,并判断是否为可以显示的图片类型。

```
FileExt := UpperCase(ExtractFileExt(FileListbox.FileName));  
if (FileExt = '.BMP') or (FileExt = '.ICO') or (FileExt = '.WMF') or (FileExt =  
= '.EMF') or (FileExt = '.JPG') or (FileExt = '.JPEG') then
```

然后创建一个窗体对象。

```
FullForm := TForm.Create(Self);
```

在创建时,设定窗体对象的父对象为当前窗体,这样在释放当前窗体时将自动释放这个窗体,然后在新的窗体中创建一个图像对象。

```
FullImage := TImage.Create(FullForm);
```

接下来设定新窗体的标题和当前窗体的标题相同,也就是文件名和文件的尺寸。

```
FullForm.Caption := Self.Caption;
```

然后设定图像对象的父对象为新窗体,这样就会在新窗体中显示图像对象了。

```
FullImage.Parent := FullForm;
```

设定图像对象的对齐属性值为客户区,也就是充满全部的新窗体范围。

```
FullImage.Align := alClient;
```

接下来让图像对象读取图像文件。

```
FullImage.Picture.LoadFromFile(FileListBox.FileName);
```

然后设定新窗体为全屏。

```
FullForm.WindowState := wsMaximized;
```

最后显示新窗体。

```
FullForm.Visible := true;
```

此时还可以自由调整窗体的尺寸,效果如图 1-8 所示。

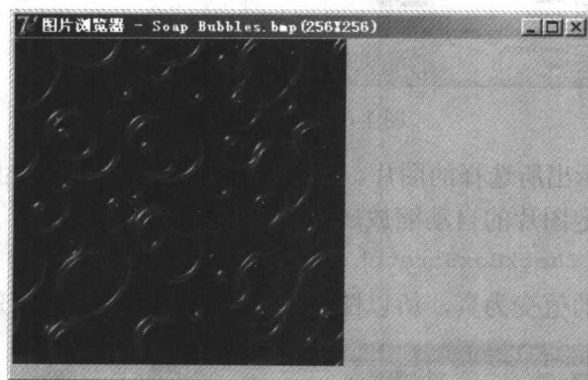


图 1-8 窗体的尺寸

这样就完成了一个图片浏览器的设计。

2

Chapter

增强标签

本章重点：

标签控件是各种控件中最简单的一种，本章将讲解如何在已有标签的基础上增强标签的效果和功能。



2-1 程序设计实例 1：标签的发光和鼠标响应

在本节中将为标签控件增加发光的效果，然后分析 VCL 库的实现方式，最后为发光标签增加响应鼠标事件的过程。

2-1-1 发光效果

在本小节中讲解发光效果的绘制方法。



文件

以下程序在光盘的“代码\2.1.发光标签\Step1”目录下。

创建一个应用程序，这个应用程序将用于测试标签的效果。单击 Component 菜单，单击 New Component 菜单项，出现如图 2-1 所示的创建新组件对话框。

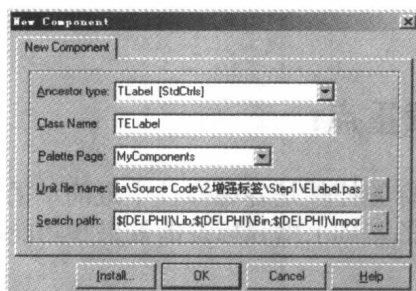


图 2-1 创建新组件对话框

在 Ancestor type 组合框中选择 TLabel [StdCtrls] 条目，这代表 VCL 库中的标签对象，表示新的组件使用 TLabel 作为父类。在 Class Name 文本框中输入 TELabel，表示新组件的类名为 TELabel。在 Palette Page 组合框中输入 MyComponents，表示新组件将放在 Insert 面板上的 MyComponents 选项卡中。在 Unit File Name 文本框中输入 UnitELabel.pas，这是单元文件的文件名。然后单击“Install...”按钮，出现如图 2-2 所示的安装组件对话框。

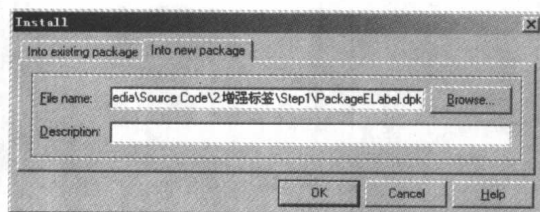


图 2-2 安装组件对话框

在 File Name 文本框中输入 PackageELabel.dpk，表示新创建的组件将放置在这个包中。然后单击“OK”按钮，一个继承自 TLabel 控件的新控件就完成了。自动生成的代码如下所示：

```
unit UnitELabel;
```

```
interface
```



```

uses
  SysUtils, Classes, Controls, StdCtrls;

type
  TLabel = class(TLabel)
  private
    { Private declarations }
  protected
    { Protected declarations }
  public
    { Public declarations }
  published
    { Published declarations }
  end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('MyComponents', [TLabel]);
end;

end.

```

在这里需要注意到 Register 过程，这个过程用于向 Delphi 7 中注册新的控件。为了编程需要，引入 Graphics 和 Windows 单元。

```

uses
  SysUtils, Classes, Controls, StdCtrls, Graphics, Windows;

```

为了把增加的功能和标签原有的功能明确地区分开，所以定义一个标签选项类型。

```

type
  TLabelOption = class(TGraphicsObject)
  private
  protected
  public
  published
  end;

```

在 TLabel 中定义一个选项变量：

```

private
  mOption : TLabelOption;

```

把这个变量发布为标签的属性：

```

published
  property Option : TLabelOption read readOption write writeOption;

```

定义一个读取选项的函数：

```

private
  function readOption : TLabelOption;

```