

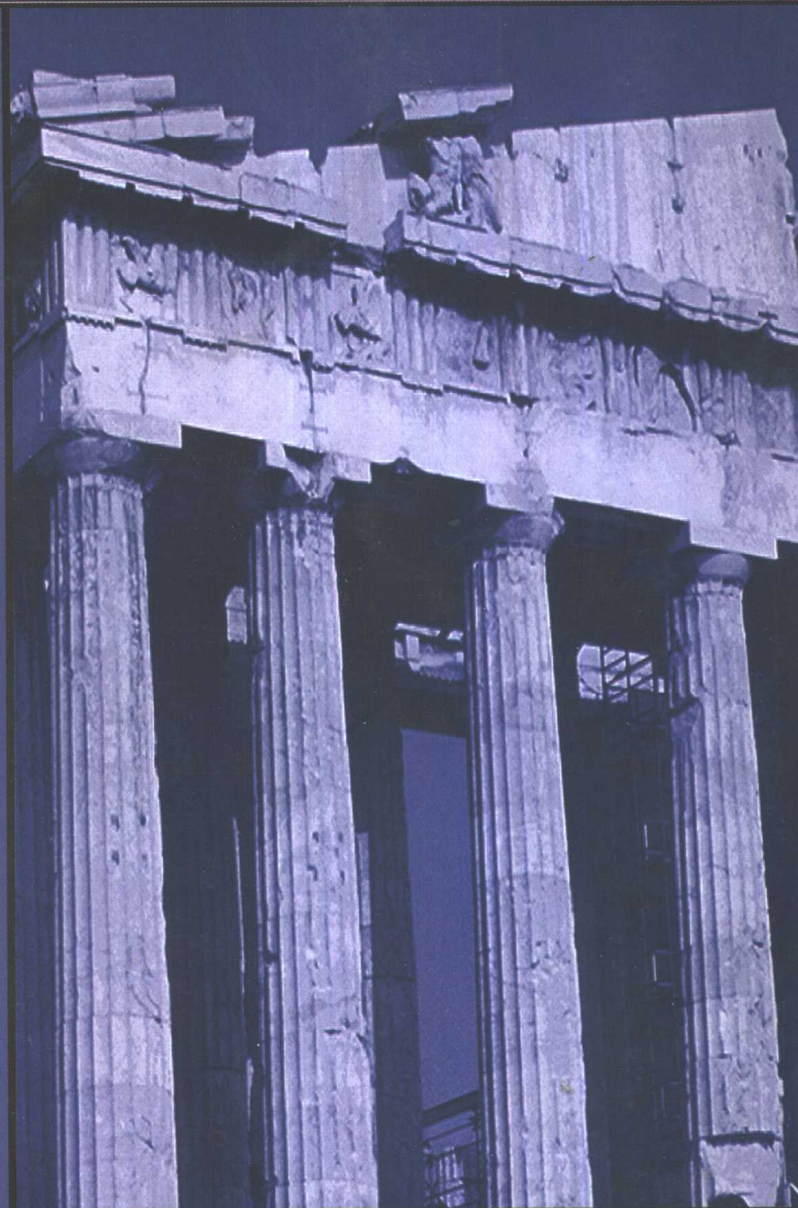


光盘内附书中范例

特别为 Delphi 使用者所设计的范例书

掌握 Delphi 组件的必备宝典

充分领略 Delphi 的强大火力



山风工作室 山风 山月 编著

Delphi

应用实例



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Delphi 应用实例

山风工作室/山风 山月 编著



中国铁道出版社

2001·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2001-2463 号

版 权 声 明

本书中文繁体字版由台湾博硕文化股份有限公司出版(2001)。本书中文简体字版经台湾博硕文化股份有限公司授权由中国铁道出版社出版(2001)。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

本书贴有博硕文化激光防伪标签,无标签者不得销售。版权所有,侵权必究。

图书在版编目(CIP)数据

Delphi 应用实例/山风工作室编. —北京:中国铁道出版社, 2001. 8

ISBN 7-113-04324-0

I. D… II. 山… III. Delphi 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 054328 号

书 名: Delphi 应用实例

作 者: 山风工作室/山风 山月

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: 苏 茜

特邀编辑: 李晓霞

封面设计: 冯龙彬

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张: 29.25 字数: 712 千

版 本: 2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

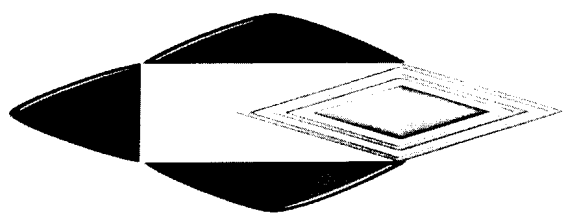
印 数: 1~5000 册

书 号: ISBN 7-113-04324-0/TP·600

定 价: 48.00 元

版权所有 盗印必究

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。



出版说明

本书共分三篇十一个章节，讲解 Delphi 的实例应用方法，内容丰富而详实，所有范例都是实际开发中最实用、最流行的。主要包括 VCL 组件、绘图及网络等方面的内容，可以说是软件开发人员的最佳参考。

书中配有一张范例光盘，每一章的范例源程序存放在各章的目录下。由于时间匆忙，所附光盘为繁体版，若出现乱码，请用“东方快车”等汉化软件进行转换，敬请读者谅解。

本书由台湾博硕文化股份有限公司提供版权，中国铁道出版社计算机图书项目中心审选；姜谷鹏、高胜、王明、沈放、邓力、罗怡、葛兰、彭涛、孟丽花、肖志军、廖康良、陈贤淑等同志完成了本书的编排工作。

中国铁道出版社

2001 年 9 月

目 录

第1篇 VCL 组件

第1章 VCL 组件制作一.....	3
1.1 可绘图的 DrawPanel.....	4
1.2 DrawPanel Part II	10
1.3 VCL 组件的属性(Properties)	24
1.4 VCL 组件的事件 (Events)	25
第2章 VCL 组件制作二.....	35
2.1 新组件的父类.....	36
2.2 非可视化新组件.....	36
2.3 TGraphicControl 类的组件.....	44
2.4 属性编辑器.....	49
2.5 组件编辑器.....	57
2.6 拦截事件.....	64
2.7 VCL 管理.....	67
第3章 E-mail 组件.....	69
3.1 数据编码.....	70
3.2 数据编码组件——TMyEncode.....	80
3.3 E-Mail 信件内容结构	93
3.4 E-Mail 信件内容制作	103
3.5 MySendMail 组件实作	112
3.6 MySendMail 组件使用范例	117

第2篇 绘 图

第4章 OpenGL 一.....	123
4.1 建立 OpenGL 环境.....	124

4.2	TOpenGL VCL 组件	129
4.3	OpenGL 模型的绘制	142
4.4	TOpenGL 的应用	148
4.5	3D 点、线、面	156
4.6	灯光	164
第 5 章 OpenGL 二		169
5.1	贴图	170
5.2	贴图坐标	181
5.3	绘制 VRML 模型	185
5.4	3D 字体	196
5.5	模型制作	207
5.6	TBitmap 与存盘	217

第 3 篇 网 络

第 6 章 CGI		223
6.1	CGI	224
6.2	以 POST 传递资料	232
6.3	使用 VCL 组件	234
6.4	Cookie 的存取	235
6.5	Web Server Application CGI	239
6.6	ISAPI	244
第 7 章 ActiveX for ASP 一		257
7.1	建立 ActiveX 组件	258
7.2	图片显示	269
7.3	在 ASP 网页绘制 OpenGL	273
7.4	OpenGL 组件	278
第 8 章 ActiveX for ASP 二		297
8.1	TFileUploadObj	298
8.2	FileUpload ActiveX 组件	307
8.3	电子邮件	316
8.4	电子邮件高级应用	326

第 9 章 ActiveX Control	333
9.1 ActiveX Control	334
9.2 Active Form	346
第 10 章 WebServer	357
10.1 TServerSocket VCL 组件	358
10.2 WebServer	361
10.3 多线程联机的 WebServer	370
10.4 WebServer 加强版	375
10.5 Web Server 其他信息	392
第 11 章 Mail Server	403
11.1 Mail Server 联机	404
11.2 SMTP Server	411
11.3 POP3 Server	427
11.4 Mail Server	441
附录	449



第1篇

VCL 组件

DDI

第1章

VCL 组件制作一

1.1 可绘图的 DrawPanel

TPanel 是个很好用的组件，在 TPanel 上可放置任何“可视化”组件（这一类的组件通常是继承自 TWinControl 或 TGraphicControl），如 TImage、TEdit、TListBox 等。TPanel 本身也是个“可视化”组件，因此也可以将 TPanel 放在另一个 TPanel 之内。不过 TPanel 有一个美中不足的地方——无法“直接”在上面绘图。

要绘图当然也可以放 TImage 组件在 TPanel 上，然后在 TImage 绘图，如图 1-1：

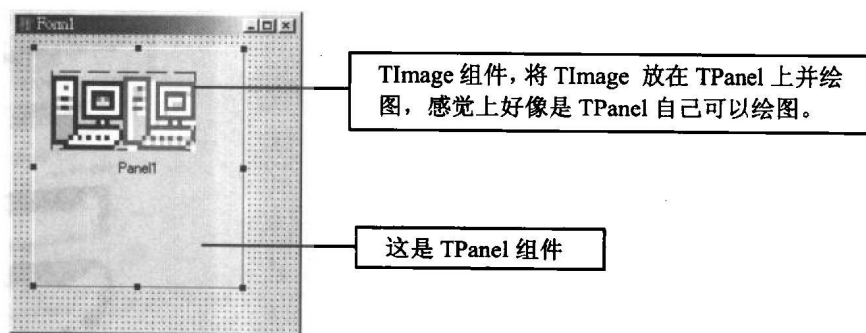


图 1-1

应用程序便可用 Image1.Canvas 的各种工具来绘图。如果觉得每次都要另外放一个 TImage 实在是太麻烦，那么就用新的 TDrawPanel 组件来取代 TPanel+TImage 的组合。

建立新组件

建立新组件有两种方式：

1. 直接建立一个空白 UNIT 文件，并在 UNIT 内输入必要的程序代码，如下：

```
unit UnitName;  
//单元名称必须与文件名称相同  
  
interface  
  
uses  
    Windows, Messages, SysUtils, Classes, Graphics,  
    Controls, Forms, Dialogs, ExtCtrls;  
  
type  
    T 组件名称 = class(T 祖先)  
    private  
    protected  
    public  
    published  
end;
```

```

procedure Register;
implementation

procedure Register;
begin
    RegisterComponents('新组件放置的位置', [T 组件名称]);
end;

end.

```

这就是一个组件的框架，此处有两点要注意：

- (1) 组件名称并不能用中文字，因此请将“T 组件名称”改为以英文字母“T”为开头的英文字符串。“新组件放置的位置”却可以使用中文字符串（如图 1-2），当组件注册成功后，会在“IDE”的组件 Palette 中显示出来，详细情形请参考“方式二”的说明。

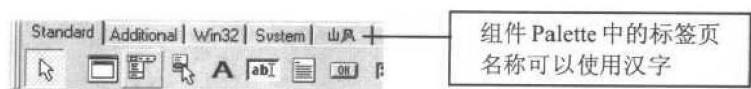


图 1-2

- (2) 在组件单元（Unit）中要有“Register”这个自定义程序（procedure），这是让组件注册用的，而且程序中通常只有 RegisterComponents() 程序。
2. 方式一中的程序代码虽然不多，但是如果每次建立新组件都要输入一次，实在是没有必要，再说 Delphi 也已经帮你写好这些基本程序代码。菜单“Component  New Component...”会打开“New Component”对话框如图 1-3。

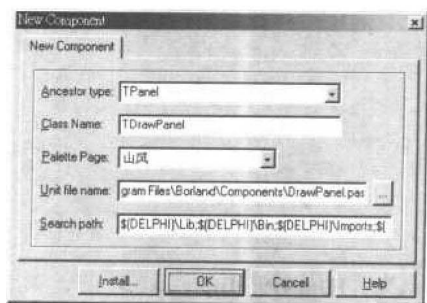


图 1-3

- (1) Ancestor type: 新组件的祖先，以本节 DrawPanel 为例，就是 TPanel。
- (2) Class Name: 新组件的名称，通常在名称之前加个“T”，如“TdrawPanel”，在组件 Palette 中会显示“DrawPanel”。如果名称是“Tower”，则会变成“ower”，因此若要使用正确的名称 Tower，请在名称前多加个 T 成为“Ttower”。Delphi 内建的组件也是如此做法，如 TTimer 等。

- (3) Palette Page: 新组件要放在哪一个标签页中, 可以是已存在的标签页, 如 Standard、Additional...。或是直接输入一个新的标签页, 如“我的组件”, “My VCL”。
- (4) Unit file name: 默认为组件名称 (开头不加 T), 可以更改存储位置, 但是文件名请保留原来名称。
- (5) Search path: 单元文件的搜索路径, 如果新组件放置的位置没有在列表中, 系统会自动加入新路径。

按下“OK”钮后 Delphi 便自动产生必要程序代码:

```
unit DrawPanel;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, ExtCtrls;

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

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('山风', [TDrawPanel]);
end;

end.
```

该有的程序代码都已经有了, 如果现在就注册这个组件, 新组件功能将与 TPanel 完全一样, 因为我们尚未在新组件加入新功能。请将上面 TdrawPanel class 修改成:

```
...  
TDrawPanel = class(TPanel)  
public  
    property Canvas;  
end;  
...
```

private、protected、published 这三个部分目前尚未用到，可以先删除。

编译与注册组件

其实 TPanel 内部有 Canvas，继承自 TCustomControl，查看一下 TCustomControl 的源代码便可知道。

```
unit Controls;  
...  
TCustomControl = class(TWinControl)  
private  
    FCanvas: TCanvas;  
    procedure WMPaint(var Message: TWMPaint);message WM_PAINT;  
protected  
    procedure Paint; virtual;  
    procedure PaintWindow(DC: HDC); override;  
    property Canvas: TCanvas read FCanvas;  
public  
    constructor Create(AOwner: TComponent); override;  
    destructor Destroy; override;  
end;
```

TCustomControl 中 Canvas 被设为 protected，TPanel 内部虽然可以使用这个 Canvas，但是外部就无法使用。TDrawPanel 的目的只是将这个内部的 Canvas 属性设为 public，让外部程序能够直接调用。请用菜单“Component |  Install Component”安装并注册这个新的 VCL 组件。

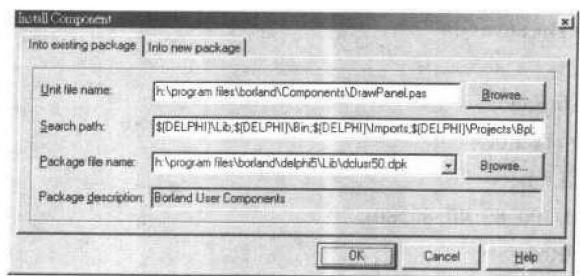


图 1-4

图 1-4 中的设置保留所有默认值，按“OK”钮 Delphi 会自动编译及安装注册。如果编译

时都没有任何错误产生，会出现如图 1-5 对话框。



图 1-5

组件 Palette 中也会立刻多了一个组件。如图 1-6。

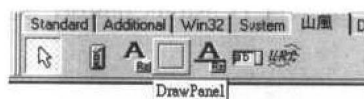


图 1-6

测试新的组件

现在我们建立一个新“应用程序”项目来测试新组件的功能。在新项目 Form1 加入一个 DrawPanel 组件，以及一个 TButton 按钮。如图 1-7。

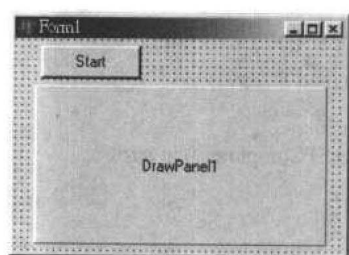


图 1-7

在 Button1 的 OnClick 事件输入下面程序代码：

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    with DrawPanel1.Canvas do
    begin
        Brush.Style := bsSolid;
        Brush.Color := clYellow;
        Pen.Color := clLime;
        Pie(20, 20, 100, 100, 80, 40, 80, 80);
        Pen.Color := clRed;
        Brush.Color := clRed;
        Ellipse(60, 40, 65, 45);
    end;
end;
```



```

end;

procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin
    //请将 Form1 的属性 KeyPreview 设为 True
    if Key=#27 then
        Close;
end;

```

编译并执行结果如下。

新组件 TDrawPanel 确实可以直接在其上面绘图，不再需要通过 TImage 组件绘图，此外 TPanel 原来有的功能也都完全保留，一样可以当作容器在上面放置其他组件。但是新组件 TDrawPanel 还有一个很大的问题，如果将应用程序缩小再还原，或是用另一个窗口将其遮盖再移开，那只可爱的小精灵便消失不见了，下一节我们将解决这个问题，并且为 TDrawPanel 多加一些好用的功能。如图 1-8。

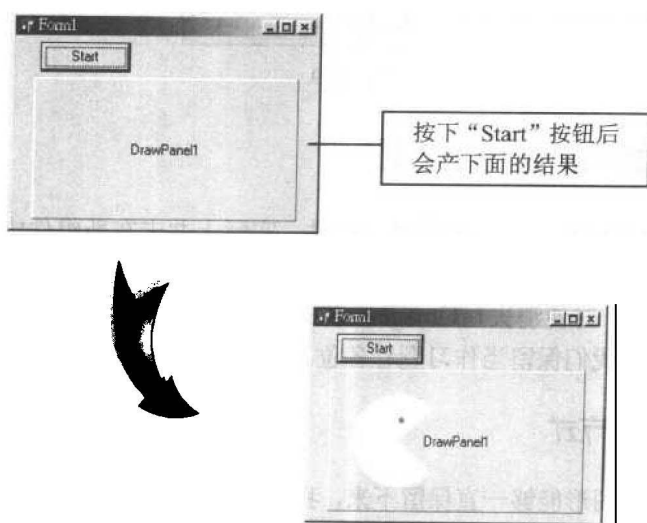


图 1-8

DCLUSR50.DPK

或许不小心在 DrawPanel.pas 中打错字了，使得无法正确编译，而组件又已经安装在组件 Palette 中，那要如何重新编译修改过的组件程序代码呢？

执行菜单“Component | Install Packages...”，会显示 Project Options 对话框，如图 1-9。

“Package - dclusr50.dpk”窗口工具栏第一个按钮便是重新编译的工具钮，修改好组件程序代码，可以按下这个按钮来重新编译组件。而编译时是对所有自制组件都重新编译，并不是对单个组件编译。

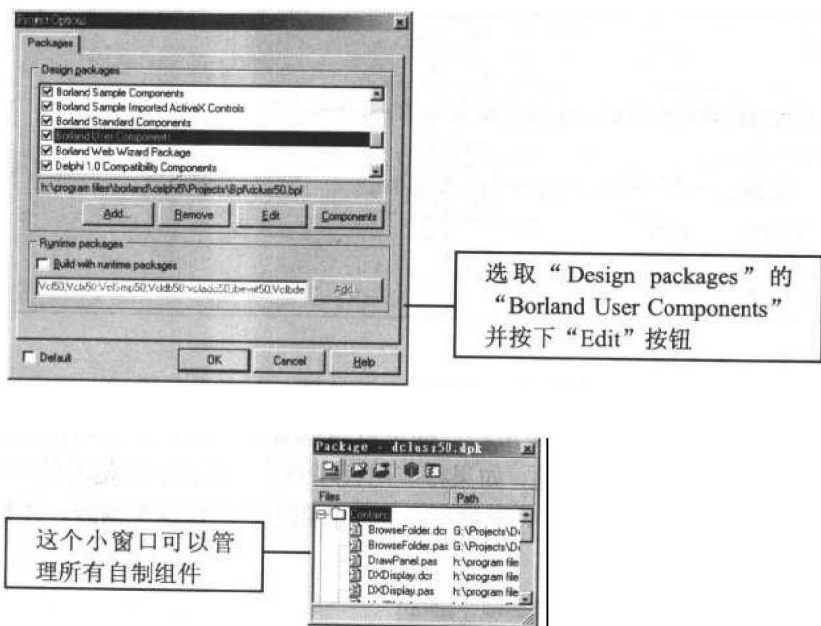


图 1-9

1.2 DrawPanel Part II

上一节 TDrawPanel 组件是一个最简单的组件，实际上并未在新组件中加入任何新功能，只不过是将其内部隐藏的功能公开出来。测试的结果，新组件似乎不是很成功，组件上所绘制的图形在“窗口重绘”时便消失！TImage 一样是用 Canvas 来绘制图形，可是却不会消失，什么原因呢？这个问题我们保留当作习题让各位自己动动脑筋。

建立 Canvas 的方式

为了让辛苦绘制的图形能够一直保留下来，我们必需放弃 TPanel 内部的 Canvas，并且自己建立一个新的 Canvas 来使用。建立 Canvas 有两种方式，

- 直接使用 TCanvas 建立。
- 建立一个 TBitmap 对象，并使用 TBitmap 内的 Canvas。

方式 a

在 TDrawPanel 建立 Canvas 的方式如下：

```

TDrawPanel = class(TPanel)
private
    fCanvas: TCanvas;
    ...
protected
    procedure Loaded; override;

```