Delphi 5

网络数据库编程

程不功　程江　编著

广东科技出版社

- 本机数据库编程
- Web 数据库访问
- 分布式多层数据库系统

Delphi 5 网络数据库编程

程不功　程　江　编著

广东科技出版社
广　州

图书在版编目（CIP）数据

Delphi 5 网络数据库编程/程不功，程江编著．—广州：广东科技出版社，2000．2
ISBN 7-5359-2431-X

Ⅰ．D…
Ⅱ．①程…②程…
Ⅲ．关系数据库-数据库管理系统，Delphi 5
Ⅳ．TP311.13

出版发行：广东科技出版社
（广州市环市东路水荫路 11 号　邮码：510075）
E－mail：　gdkjzbb@21cn．com
出 版 人：黄达全
经　　销：广东新华发行集团股份有限公司
排　　版：广东科电有限公司
印　　刷：番禺市新华印刷有限公司
（广东省番禺市环城西路工农大街 45 号　邮码：511400）
规　　格：787mm×1 092mm　1/16　印张 11　字数 264 千
版　　次：2000 年 2 月第 1 版
2000 年 2 月第 1 次印刷
印　　数：1～5 000 册
定　　价：22.50 元

前　言

近几年来计算机网络技术有了蓬勃的发展，它的广泛应用不仅改变着各国的政治、经济结构，也正在改变着人们生活的各个方面，极大地冲击着人们的传统观念。可以说，一个以Internet/Intranet为中心的网络时代已经到来。作为一名21世纪的程序设计员，不懂得网络程序设计，将不是一名称职的程序设计者。

形势对每一位计算机专业工作者或爱好者来说，既是一次极好的机遇，也是一次严峻的挑战。因为网络技术和单机技术相比，理论要深得多，涉及的面要宽得多，因而掌握这门技术的难度也大得多。其捷径是选择学习一门先进的、有代表性的、综合性的工作平台，通过它迅速掌握基本思想、观念和技术，为进一步提高设计技术奠定基础。

根据作者多年的教学和开发经验，Delphi 5是工作平台最佳的选择。Delphi是开发网络应用程序的最有力的工具之一，是网络应用工具百花中的“奇葩”，与其他类似的工具相比，在分布式多层网络数据库方面，一直处于领先地位。1999年夏季推出的Delphi 5.0新版本，将更多的最新技术融入系统中，如将HTML4，XML，ADO，MTS，CORBA等融入系统中，同时提供了对Asp，JavaScript，Java等技术的支持，从而更进一步地巩固了它的领先地位。

本书试图从多方面介绍上述技术，然而这些技术都有丰富的内容，每一项都可以单独写成一本厚厚的书。为读者迅速掌握的需要，本书准备通过典型示例介绍它们最基本的应用和开发方法。读者只要掌握了这些基本方法，然后再有针对性地选读一些其他资料，将很容易扩展设计能力。经验证明，先掌握基本技术，再扩展能力的方法，最适合于自学者。用计算机的俗语来说，即“深度遍历优先”。

本书分三编十七章。第一编为“本机数据库编程”；第二编为“Web数据库访问”；第三编为“分布式多层数据库系统”。其中第一编是基础，第二、三编是重点。通过示例深入浅出地阐述问题，重点讲述设计方法，同时适当地介绍有关理论。本书非常适合于自学者，也可作为大专院校的教材或参考书。

作者

1999年12月

于长沙大学

第一编　本机数据库编程

第一章　Delphi 5.0 的安装环境与界面设计

本章将介绍 Delphi 5.0 的安装环境和编程环境，并通过实例讲述界面程序设计的要点。主要的问题包括：

安装环境；

程序设计的集成环境；

应用项目的组成；

项目管理；

几个信息对话窗口；

界面程序设计示例。

第一节　安 装 环 境

Delphi 5 目前已有标准版（Standard）、专业版（Professional）、企业版（Enterprise）等版本，本书将主要介绍 Enterprise（Client/Server）版本的有关内容。

1．安装 Delphi 5 对系统的需求

Windows 95/98，Windows NT 4/2000 或 Workstation 的操作系统环境，如果是 Windows NT 4 或 Workstation，还必须安装 Service Pack 3 或以上补充软件；

CPU 最低为奔腾 90 芯片（建议奔腾 166 以上）；

内存最少 32MB（建议 64MB 以上）；

硬盘 220MB 以上可用空间（若最小安装时，80MB 以上可用空间）；

光驱（CD_ROM Driver）；

高分辨率显示器；

鼠标；

网络支持（任何 Windows 95/98，NT/2000 或可兼容网络的支持）。

2．安装后的缺省目录

（1）程序文件

C：\ Program Files \ Borland \ Delphi 5

（2）共享文件

C：\ Program Files \ Common Files \ Borland Shared

（3）DBD 文件

C：\ Program Files \ Common Files \ Borland Shared \ DBD

(4) BDE 和 Links 文件

C: \ Program Files \ Common Files \ Borland Shared \ BDE

(5) CORBA 支持文件

C: \ Program Files \ Borland \ Vbroker

第二节　程序设计的集成环境

Delphi 5.0 的编程环境是一个集成环境（Integrated Development Environment，IDE）。所谓“集成环境”是指：将程序设计所需工具以及帮助系统都集中在一起，使得程序的编辑、编译、调试、运行都在同一个环境中进行，十分方便。这个集成环境包括 6 个部分（见图 1-1）。

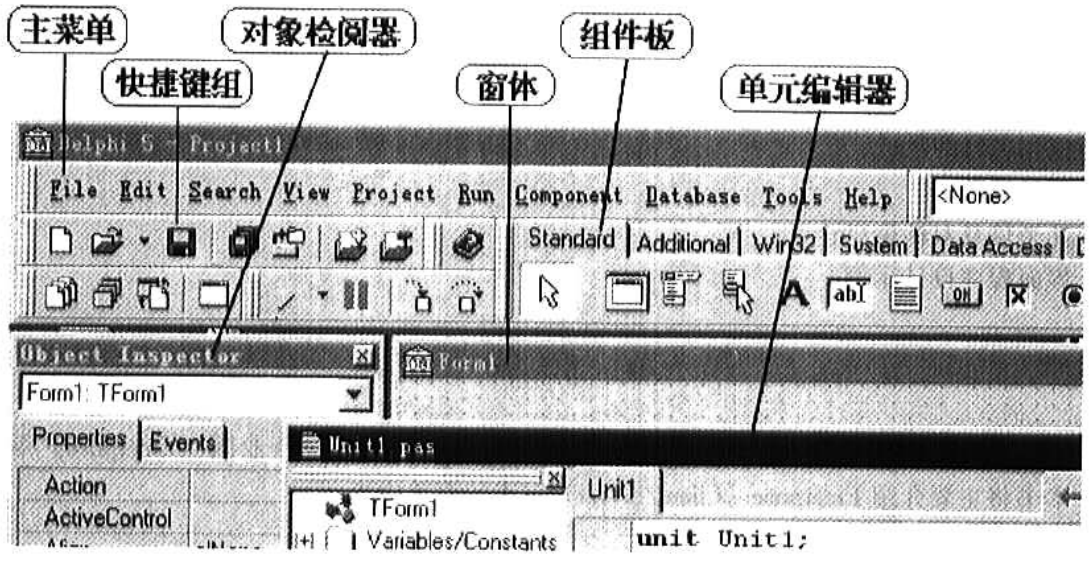

图 1-1　程序设计的集成环境

1．主菜单（Main Menu）

主菜单包括：文件（File）、编辑（Edit）、查询（Search）、察看（View）、项目（Project）、运行（Run）、组件（Component）、数据库（Database）、工具（Tools）、帮助（Help）10 大项，每个大项中又包括多个子项。

2．快捷键组（SpeedBar）

将菜单中常用的子项，如创建新文件（New）、打开文件（Open）、存储文件（Save）、运行程序（Run）、暂停运行（Pause）等 16 项，以图形按钮的形式集中到一起，组成快捷键组，只需用鼠标单点操作即可启动上述操作过程。

3．组件板（Component Palette）

Delphi 5 系统提供了约 230 个组件，归类后分别放在 19 页中，每页组件的作用概述如下：

Standard	标准 Windows 控制组件
Additional	特定的 Windows 控制组件
Win32	对 32 位 Windows（Windows 95/98 或 NT 等）控制的接口组件

System	系统控制组件
DataAccess	通过 BDE（Borland Database Engine）存取数据库的组件
DataControls	数据库控制组件
ADO	通过 ADO（ActiveX Data Object）存取数据库的组件
InterBase	直接连接和存储（不通过 BDE 或 ADO）InterBase 库的组件
MIDAS	创建分布式多层数据库系统的组件
InternetExpress	基于 Web 服务器的分布式多层数据库系统组件
Internet	创建 Web 服务器应用的组件
FastNet	提供 Internet 多种存取规范组件
Decision Cube	决策支持组件
QReport	快速打印组件
Dialogs	通用对话窗口组件
Win 3.1	Windows 3.1 控制组件（便于向前兼容）
Samples	范例或自己创建的组件
ActiveX	由其他系统创建的 ActiveX 组件
Servers	连接和存取 COM 服务器的组件

4．窗体（Form）

窗体中可以放置各种对象（如编辑框、对话框、按钮等），窗体是程序运行的主要界面。

5．对象检阅器（Object Inspector）

对象检阅器是设置或查看窗体以及窗体内各对象的属性和事件的场所。

6．单元编辑器（Unit）

单元编辑器是编写程序的地方。

第三节　应用项目的组成

在 Delphi 中，每一个应用项目由多种文件组成，每种文件发挥着自己独特的作用，组合到一起才能运行。这些文件主要包括：项目文件、窗体和窗体文件以及代码单元。一个独立的应用项目只有一个项目文件，但有一个或一个以上的窗体（窗体文件）及代码单元。

当进入编程环境时，或者用鼠标单击主菜单的 File，再单击 New Application 的选项（上述过程可用 File|New Application 代表，以后同）时，系统将自动产生一个新项目，包括一个项目文件、一个窗体和窗体文件及一个代码单元，系统为它们取的缺省名如下：

项目文件	取名为 Project1.dpr
窗体和窗体文件	窗体取名为 Form1，窗体文件取名为 Unit1.dfm
代码单元	取名为 Unit1.pas

这几个文件组成了一个最小的运行单位。如果立即对它们进行编译和运行，将不会产生任何错误。当然这个项目现在还什么事情都不能做，然而开发一个应用项目常常就是从这个基础开始的。

开发新项目时，应注意给上述文件更名（简单练习时可不更名）。更名的目的是为了便于记忆，更主要的是为了防止被后续文件覆盖。

应直接在窗体的“对象检阅器”（Object Inspector）中的 Name 属性中为窗体更名。项目

文件以及代码单元的改名，则只能在存储时修改。即使用主菜单的（File|Save as）存储单元文件时改名，窗体文件名将跟随单元文件一道更改，但后缀保持 .dfm。

各文件的作用简述如下：

一、项目文件

项目文件是项目的“总管”，它的作用是登录项目内的窗体和代码单元文件，并启动程序运行。项目文件的后缀为 .dpr。经过编译、联接后生成的可执行程序与项目文件名相同，但后缀变成 .exe。例如：项目文件名为 sample1.dpr 时，编译、联接后将生成 sample1.exe 可执行程序。这个程序能够脱离 Delphi 环境，单独在 Windows 环境下运行。

Delphi 中的项目文件由系统生成，能自动反映项目中的各种变化。程序设计者通常不要直接修改这一文件。

二、窗体

窗体是一可视对象，拥有对象的全部特征。除此以外，窗体中还可以放置其他类型的对象，因此窗体又是一种“积聚对象”。积聚是与继承略有区别的面向对象的概念。简单地说，一个积聚对象的内部可以包含其他对象，并引出内部对象的特定接口，以扩充自己的服务。

窗体是程序设计的中心，是程序活动的“舞台”，窗体与其他对象组成了用户的可视界面。

系统用“窗体文件”和“代码单元”两个文件来记录窗体及窗体内各对象的属性和行为。窗体文件是一个 Text 文件，其后缀为 *.dfm。窗体是一个可视界面，而窗体文件是这个界面属性的记录。它记录了窗体的属性，如位置、长宽、字体、颜色等等。

窗体文件也是由系统自动生成的，它能自动反映窗体的各种变化情况。

三、代码单元

代码单元中包括“类”的定义、变量定义、函数与过程等。代码单元是窗体内各对象定义以及行为的记录。程序设计者在其中编写的程序，主要用来规定各对象对消息响应方式。其他部分由系统自动生成。

代码单元文件的后缀是 .pas。

第四节　项 目 管 理

通常情况下，每个窗体可作为一个独立部分单独进行设计，然后再利用项目管理器（Project Manager）将它们组合到项目中来。

一、项目管理器

选用主菜单的 View|Project Manager 项，将启动项目管理器，其界面见图 1-2。

窗口中的树形列表显示出项目中已经包括的文件（Files）及所在的路径（Path）。其中 ProjectGroup1 代表项目组。项目组将几个有关的可执行程序组合到一起，便于调试。

此时若单击主菜单的 File|New Form，系统将给项目增添一个新的空白窗体和初始的代码单元。若需增加已设计好的窗体时，按上方带 New 的按钮，然后在弹出的对话窗口中选择

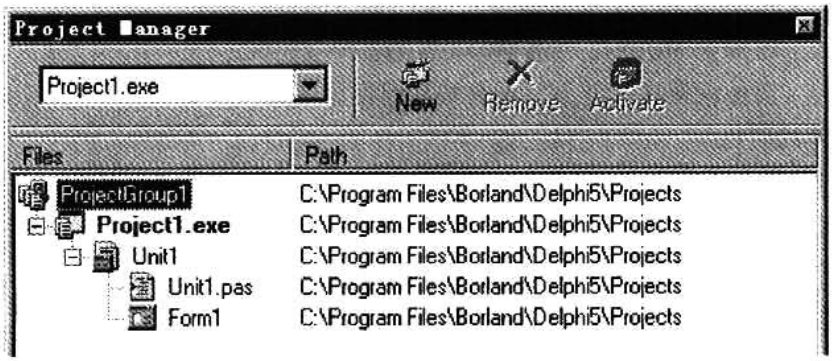

图 1-2　项目管理器

需增加的窗口和代码单元。

项目中，每增加一个新窗体，都将自动带入一个新的窗体文件和代码单元。反过来，当增加一个代码单元时，却不一定带上新窗体，因为这个代码单元可能只是为已有的窗体提供新的函数或过程。

删除一个窗体时，先用鼠标选择窗体，然后按上面的 Remove 按钮，此时的窗体和代码单元虽然从项目中消除，但并未从磁盘中删除。

用鼠标右击窗口，从弹出的菜单中选项，同样能完成上述功能，弹出的菜单如下：

Save Project	存储项目文件
Add Project To Repository	将项目存入模板中
New Unit	增加一新代码单元
New Form	增加一新窗体
Add File	增加文件
Remove File	删除文件

二、多窗体之间的引用

1．窗体的类型

窗体分多文档界面（Multiple Document Interface，简称 MDI）和单文档界面（Single Document Interface，简称 SDI）两种类型。

在 MDI 中，窗体之间存在着父子关系。其中，父窗体只能有一个，而子窗体可以有多个。子窗体可以放大、缩小、移位，但只能在父窗体中活动。关闭子窗体时，子窗体将缩小成图标，仍然处于父窗体之中。用鼠标双击图标时，子窗体再度打开。关闭父窗体时，子窗体自动关闭。

SDI 也能包括多个窗体，也需要指定一个主窗体，打开项目文件时，将首先打开主窗体。但其他窗体打开后，活动范围不受主窗体的限制。如果关闭其他窗体时，该窗体将从屏幕上消失。关闭主窗体时，其他窗体同时关闭。

项目中的窗体允许混用，即有的窗体属于 MDI，有的属于 SDI，此时父窗体必定是主窗体。

窗体的类型是通过窗体的属性 FormStyle 来设定的。该属性提供的选择有：

fsNormal　　窗体为 SDI 类型，这是缺省的状态

fsMDIForm　　窗体为 MDI 中的父窗体

fsMDIChild　　窗体为 MDI 中的子窗体

fsStayOnTop　　窗体保持在最前面，不会被其他窗体覆盖

2．创立窗体的时机

创立窗体的时机有 2 种：

打开项目文件时自动打开（但不一定立即显示）；

被引用时才打开。

确定的方法是：

选用主菜单中的 Project|Options 项，在弹出的界面中选用 Form 页（见图 1-3）。

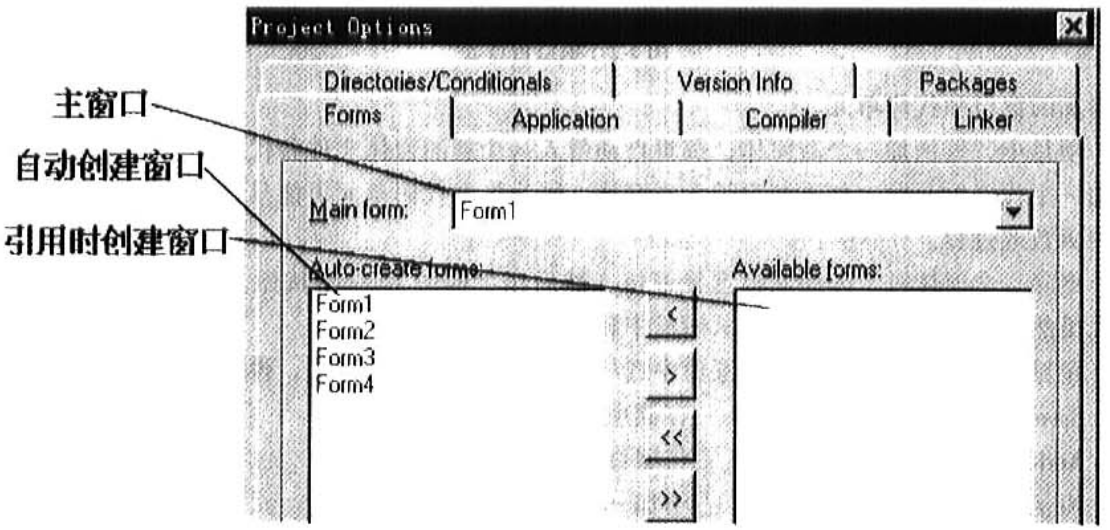

图 1-3　确定窗体创建的时机

（1）确定主窗体

从 Main form 小窗口的下拉菜单中确定一主窗体。有父子关系的窗体中，父窗体必定是主窗体。

（2）确定窗体创立的时机

左边窗口 Auto-create forms 下面列出的是自动创立的窗口，右边 Available forms：窗口中是被引用时才创立的窗口。缺省情况下，所有的窗体都列在左边，也就是说都将自动打开，这样可以缩短引用的时间，但它们占用了内存资源，也许这并不是设计者所希望的。可以指定一些窗体，被引用时才打开，方法是将这些窗体移到右边窗口即可。窗口中的“>”键代表逐个移动，“》”代表全部移动。

（3）确定创立的顺序

窗体中窗体名的先后代表创立的顺序。这个顺序可能影响开始时的外观，因为后创立的窗体将部分或全部覆盖前面创立的窗体。用鼠标上下拖动窗体名时，也就改变了创立的顺序。

3．窗体之间的引用

（1）增加 Uses 子句

只要将被引用窗体的代码单元名，写在引用窗体的 Implementation 下面的 Uses 子句中，就允许窗体之间相互引用。

当没有写上相应的 Uses 子句，引用另一窗体时，系统将提示："是否需补充 Uses 子句"（英文提示），此时只要点一下"Yes"按钮，系统将代为补上。

（2）引用命令

1）引用已创立的窗体时，由于窗体已经创立，只要将它激活并从隐蔽状态转变为显示状态即可。使用的语句是：

〈窗体名〉.Show;

或者

〈窗体名〉.ShowModal;

〈窗体名〉.Show; 的作用相当于〈窗体名〉.visible:=True;

〈窗体名〉.ShowModal; 与以上命令基本相同。区别在于本命令激活和显示的窗体呈"模式"状态。一个模式状态的窗体与其他窗体不同之处在于，用户必须关闭这个窗体后才能继续其他的操作。

2）引用未创立的窗体时，需先创立再激活和显示。例如 Form2 是被引用的窗体，引用前没有创立。引用的程序段如下：

```
Var
  Form2: Tform2;
Begin
  Form2:=Tform2.Create(Self);      {创建窗体}
  Form2.Show;                      {显示窗体}
End;
```

三、编译、运行和停止

选用主菜单中的 Project|Compile〈项目名〉或 Project|Build〈项目名〉项（或直接按"Ctrl"+"F9"键）可进行编译。如果语法有错，根据提示修改后再编译，直到无错误时再选菜单中的 Run|Run 项（或直接按"F9"键）转入运行。

Compile 与 Build 命令的区别是：Compile 只重新编译项目中的修改部分，而 Build 却编译项目中的全部内容。

也可以将两步合并成一步，即直接选用菜单中的 Run|Run 项（或直接按"F9"键），系统将自动进行编译和运行。

终止程序运行时，可用关闭窗体的方法，也可选择主菜单中的 Run|Program Reset（或按"Ctrl"+"F2"键）。

第五节　几个信息对话窗口

窗体中的提示信息要通过信息对话窗口。Delphi 使用的几个主要的信息窗口有：

1. ShowMessage（）信息窗口

过程首部：

```
Procedure ShowMessage(const Msg: String);
```

或者

```
Procedure ShowMessagePos(const Msg: String, X, Y: Integer);
```

前者在固定位置显示一字符串；后者可通过 X，Y 坐标，改变显示位置。

2．MessageDlg（）信息对话窗口

函数的首部：

Function MessageDlg（const Msg: String; Atype: TmsgDlgType; Abutton: TmsgDlgButtons; HelpCtx: Longint）: Word;

其中：

Msg 代表提示字符串；

Atype 代表图标的类型，有 mtWarning，mtError，mtInformation，mtConfirmation，mtCustom 等；

Abutton 代表按钮，有 mbYes，mbNo，mbOK，mbCancel，MbHelp，mbAbort，mbRetry，mbIgnore，mbAll 等，还可以是按钮组合，如 mbYesNoCancel，mbOKCancel，mbAbortRetryIgnore 等；

HelpCtx 是一个与帮助信息有关的长整型数，通常设为 0；

函数返回值属于 Word 类型，可以用以下符号代表：mrYes，mrOK，mrRetry，mrNo，mrCancel。

3．InputBox（）输入窗口

函数首部：

Function InputBox（const Acaption，Aprompt，Adefault: String）: String;

函数将显示一输入窗口，等待用户输入字符串。参数 Acaption 是窗口的标题；参数 Aprompt 是一串提示符；参数 Adefault 是缺省值，可以用空字符作为缺省值。

窗口中有 OK，Cancel 两个按钮。如果按 Cancel 按钮将返回缺省值；按 OK 按钮时，返回输入的字符串。

4．MessageBox（）应用信息窗口

函数首部：

Function Application.MessageBox（Text，Caption: Pchar; Flags: Word）: Integer;

这是一个全中文的信息窗口，其中：

Text 是窗口内的提示；

Caption 是窗口标题；

Flags 代表按钮组合，包括 mb_YesNo（是、否），mb_YESNOCANCEL（是、否、取消），mb_RETRYCANCEL（重试、取消）。

函数返回值是一整型数，也可用符号代表。

符号	数值	含　义
IDYES	6	用户选择“是”按钮
IDNO	7	用户选择“否”按钮
IDCANCEL	2	用户选择“取消”按钮
IDRETRY	4	用户选择“重试”按钮

第六节　界面程序设计示例

由于 Delphi 提供了功能齐全、运行可靠的组件，在这些组件的支持下，程序的设计过程变成了组装对象的过程，设计工作可以高起点、高速度地前进。

下面通过几个示例，说明界面程序设计的要点：

一、字符与图片

1．设计要求

在窗体中显示可改变的字符串及固定的图片。

2．需要用到的组件

TLabel：标签，在 Standard 页中；

TButton：按钮，在 Additional 页中；

TPanel：底板，在 Standard 页中；

TImage：图像，在 Additional 页中。

3．设计步骤

1）进入编程环境，或关闭其他项目后，使用主菜单 File | New Application 进入程序设计环境。在窗体 Form1 的 Object Inspector 中，将 Caption 属性值改写成中文："界面示例之一"。

2）将 TLabel，TButton，TPanel，TImage 组件向窗体派生对象 Label1，Button1，Panel1，Image1。注意将 Image1 放在 Panel1 的上面（见图 1-4）。

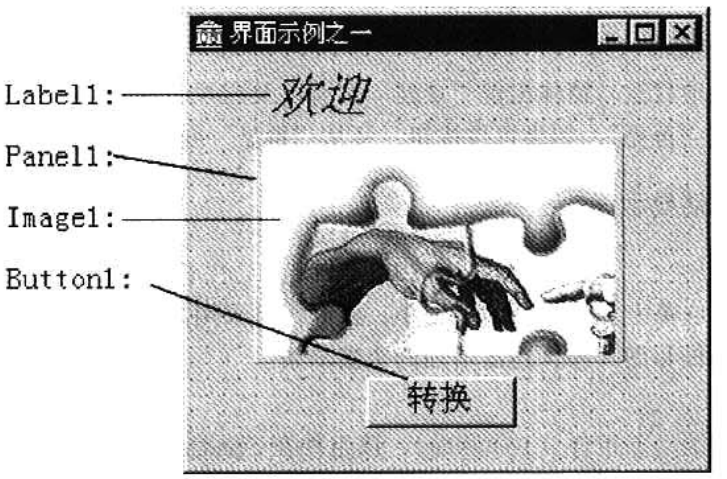

图 1-4　界面设计之一

3）将 Label1 的属性 Caption 改写为"欢迎"；将 Button1 的 Caption 属性改写为"转换"；将 Panel1 及 Image1 的 Caption 清空。

4）确定按钮上鼠标单击时，事件的响应方式、方法是用鼠标双击 Button1；或者单击 Button1 的事件 OnClick，在弹出的界面中编写程序段：

```
procedure TForm1.Button1Click (Sender: Object);
begin
 if Label1.Caption = '欢迎' then
```

```
    Label1.Caption:= '再见'
  Else
    Label1.Caption:= '欢迎';
  End;
```

【注意】 程序中的单引号都只能在英文环境下输入。

5）在 Image1 中放入图片。方法是用鼠标双击 Image1，或者单击 Image1 的属性 Picture 右边的省略符号，都将弹出图 1-5 的界面：

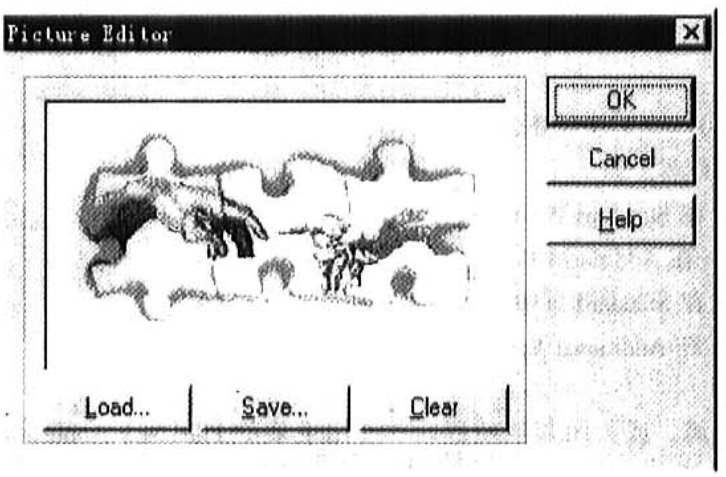

图 1-5 选择图片

单击 Load 按钮，将弹出文件对话窗口，从中选出合适的图形文件（带 .bmp 后缀的文件）后，按 OK 按钮。

编译并运行此项目后，窗体出现“欢迎”字符以及图片。若按动按钮时，窗体中的字符串从“欢迎”变成“再见”，如果再按按钮，又从“再见”变回“欢迎”等。

二、图片的滚动显示

1．设计要求

有时图片大，屏幕上允许显示的范围小，此时可以采用滚动显示图片的方法。除按上面介绍的方法设置底板和图片外，还需增加两个滚动条。

2．有关组件

程序中要用到滚动条组件（TscrollBar），该组件在 Standard 页，主要的属性有：

1）Kind 代表状态，下拉菜单中包括水平和垂直两种状态

SbHorizontal：水平状态；

SbVertical：垂直状态；

2）Max：最大值；

3）Min：最小值；

4）Large Change：改变值；

5）Position：移动块当前的位置。

移动块的位置只能在最大值与最小值之间变动，每当鼠标在滚动条两端点一下，移动块移动量为 Large Change 属性中设置的数值。

3．设计要点

（1）派生对象，设置属性

将 TPanel，TImage，TScrollBar 组件向窗体派生对象 Panel1，Image1，ScrollBar1，ScrollBar2。将 Panel1 调整到显示图片的大小，Image1 放在 Panel1 上，同时将它的属性 AutoSize 设为 True。此时图片虽保持原图的大小，但超出 Panel1 边界的部分暂不显示。

将 ScrollBar1 改变成垂直滚动条，即将它的 Kind 属性设成 SbVertical 后移至 Panel1 的右方；Scrollbar1 的 Kind 属性保持 SbHorizontal 放置下方，作为水平滚动条。

根据图片大小及显示要求设置其他属性（见图 1-6）。

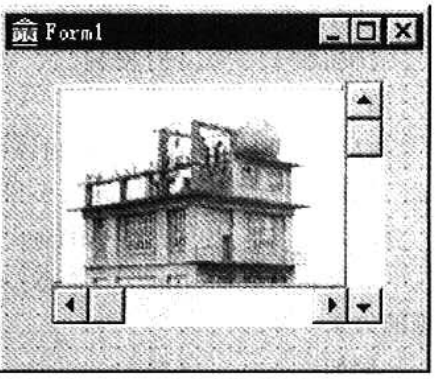

图 1-6　滚动显示图片

（2）编写程序

在两个滚动条各自的 OnChange 事件中编写程序。

例如，要求 Scrollbar1 移动块向下移动时，带动图片上移的程序如下：

```
procedure TForm1.ScrollBar1Change (Sender: TObject);
begin
  Image1.top: = -scrollbar1.position;
end;
```

要求 Scrollbar2 的移动块右移带动图片左移时的程序如下：

```
procedure TForm1.ScrollBar2Change (Sender: TObject);
begin
  Image1.left: = -Scrollbar2.Position;
end;
```

三、复合图片

1．设计要求

为使显示的图片更加生动，可在图片的基础上增加某些其他功能。例如要求在地图上几个重要城市加上闪烁功能，此时可在这些城市的位置上叠加其他组件，利用时钟组件发出信号，周期性地改变这些组件的状态，以产生闪烁的效果。

2．有关组件

设计中需用到形状组件 TShape（在 Additional 页中）及时钟组件 TTimer（在 System 页中）。

（1）TShape 的形状属性 Shape 有

.stCircle	圆形
.stEllipse	椭圆
.stRectangle	长方形
.stRoundRect	圆角长方形
.stRoundSquare	圆角正方形
.stSquarre	正方形

（2）时钟组件 TTimer 的主要属性有

.Enabled　　有效性

.Interval　　时间间隔（mm）

.Name　　对象名

3．设计要点

（1）将 TShape，TTimer 向窗体派生出对象（取名 shape1，Timer1）

根据需要改变 Shape1 的大小和位置。

（2）在 Timer1 的事件 OnTimer. 中编写程序

以便定时改变 shape1 的状态：改变颜色或在隐藏与显示两种状态之间切换，以产生闪烁的效果。

例如：

```
Procedure TForm1.Timer1Timer (Sender; TObject);
begin
shape1.visible; = not shape1.visible; {隐藏与显示之间切换}
end;
```

四、动态图片

1．移动一欢迎标语

（1）设计要求

在屏幕上从右向左移动一欢迎标语，当移至左边界时，迅速返回右边界继续显示该标语。

（2）有关组件

为了拖动欢迎标语，需用到时钟（TTimer）和底板（Tpanel）组件。假定分别将它们派生到窗体中，并取名为 Timer1 和 Panel1。

（3）设计要点

1）将 Panel1 的 Caption 改写成欢迎标语。如："欢迎参观指导！"，并利用属性 Font 将字体适当放大。

为了利用时钟拖动底板，将 Timer1 的属性设置如下：

Enabled　设为 true；

Interval　设为 50（mm）。

2）在时钟的 OnTimer 事件中写出以下程序：

```
procedure TForm1.Timer1Timer (Sender; TObject);
begin
if (panel1.Left > -260) and (panel1.Left < 410)        {拖动范围}
then
panel1.Left; = panel1.Left-1                {每次拖动的距离}
else
panel1.left; = 400;                         {返回右边界}
end;
```

2．活动图像

（1）设计要求

设计一活动的图像。例如一个不断招手表示欢迎的使者。

(2) 有关组件

利用主菜单的 Tools|Image Editor 工具，绘制两个图形。一个手在上，另一个手在下。

(3) 设计要点

1) 方法之一：两张图片之间切换。现举例说明如下：

在同一个底板（Panel1）上，将不同的两张图分别放入 Image1，Image2 中，然后将两个 Image 叠加起来。

将 Image1 与 Image2 设置不同的显示属性（即：一个 Visible：= True；另一个 Visible：= False），利用时钟信号交替显示两张图。即在 Timer1 的 OnTimer 事件中写出程序段如下：

```
procedure TForm1.Timer1Timer (Sender: TObject);
begin
 Image1.visible: = Not Image1.visible; {交换显示 Image1 与 Image2}
 Image2.visible: = Not Image2.visible;
end;
```

2) 方法之二：动态调用图片。采用动态调用图片的方法时，可以在多张图（不限 2 张）之间切换，使图像中的活动，显得更加平滑。现举例说明如下：

按照需要设计多张图，定名存储。设绘制了 3 张图，分别取名为：Bitmap1.bmp，Bitmap2.bmp，Bitmap3.bmp；

将 TPanel，TImage，TTimer 组件派生对象 Panel1，Image1，Timer1 于窗体中，将 Image1 放置于 Panel1 上面；

编写程序实现动态调用图片。设一全局变量 ii，初始值为 0，以后按定时信号加 1（ii：= ii + 1）。等于 4 时返回 1。在 Timer1 的事件 OnTimer 中编写程序：

```
    ...
    var
        Form1: TForm1;
        ii: Integer; {ii 为全局变量}
    ...
    implementation
    {$ R * .DFM}
procedure TForm1.Timer1Timer (Sender: TObject);
begin
ii: = ii + 1;
if (ii = 4) then ii: = 1;
case ii of
1: Image1.Picture.LoadFromFile ('Bitmap1.bmp'); {动态调用图片}
2: Image1.Picture.LoadFromFile ('Bitmap2.bmp'); {图片名包括路径}
3: Image1.Picture.LoadFromFile ('Bitmap3.bmp');
end;
...
Initialization
```