

精品软件“实例”丛书



Delphi 5.0 开发与实例



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

王兴晶 尹立民 等编著

TP312
W37

548

精品软件“实例”丛书

Delphi 5.0 开发与实例

王兴晶 尹立民 等编著



A0917859

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

这本书融合了利用 Delphi 5 开发 Windows 下应用程序的理论和实践,全面和深入的介绍了利用 Delphi 5 开发应用程序的常用方法和技巧。全书共分为九章,主要包括 Delphi 5 编制应用程序的一些基础知识,Delphi 5 的集成开发环境和 ActiveX 控件的用途及使用方法;文本、图形和图象处理的常用方法;菜单、工具栏和对话框设计,在 Delphi 5 中进行文件处理操作的方法,多媒体程序开发的常用方法,如何在 Delphi 5 中进行窗体程序设计和 Delphi 5 中非常吸引人的网格与数据库开发方法。

本书资料丰富、内容详实,列举了大量的技术开发实例,具有易学易用、简洁明了的特点,本书既适合于软件开发人员使用,又能够使 Delphi 初学者在学习和使用的过程中受益匪浅,也可以作为高等院校广大师生和科研人员进行软件开发的必备参考书。

另外,本书所附带的光盘中为读者提供了示例程序的完整源程序代码,并且每个程序都带有可执行文件和独立的安装程序,另外为了方便读者,特地提供了一些常用 Delphi 控件和帮助文件制作工具等。

注意:订购光盘可汇款至:北京 172 信箱今日电子杂志社(100036)

每盘定价:20 元,另加邮费 14 元,今日电子杂志发行部电话(010)68159356

图书在版编目(CIP)数据

Delphi 5.0 开发与实例 /王兴晶,尹立民等编著. -北京:电子工业出版社,1999.11
(精品软件“实例”丛书)

ISBN 7-5053-5445-0

I .D… II .①王… ②尹… III .Delphi 语言-程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(1999)第 63830 号

丛 书 名:精品软件“实例”丛书

书 名:Delphi 5.0 开发与实例

编 著 者:王兴晶 尹立民 等

责任编辑:晨 曦

特约编辑:王 宏

印 刷 者:北京市东光印刷厂

装 订 者:三河司庄装订厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>
北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店经销

开 本:787×1092 1/16 印张:26.00 字数:665 千字

版 次:1999 年 11 月第一版 1999 年 11 月第一次印刷

书 号:ISBN 7-5053-5445-0
TP·2741

定 价:40.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换。联系电话:68214070
版权所有·翻印必究

前 言

随着计算机技术的发展,计算机软件产业也随之迅速的崛起,自从面向对象的编程概念提出之后,人们的编程思想也发生了根本的变化,为了提高应用程序的高效性和统一性,可视化编程的技术应运而生,相继出现了 VB 和 VC 等可视化应用程序开发工具。

Delphi 的基础语言为 Pascal 程序设计语言,它继承了 Pascal 语言的优点,如代码结构清晰、可读性好和代码执行效率高等,所以 Delphi 1.0 一经推出,迅速得到用户的好评。在以后相继推出的 Delphi 2.0、Delphi 3.0、Delphi 4.0 和 Delphi 5.0 版,一方面保持了良好的向下兼容性,另一方面,随着版本的提高,Delphi 程序设计语言在各个方面都有了不同程度的提高,如数据库的体系结构、ActiveX 控件的开发、Web 应用程序和安全线程等,从而使得 Delphi 的功能更加强大,使用起来也更加方便和灵活,提高了应用程序的开发效率。总之,Delphi 是开发 Windows 应用程序最为强大的工具之一,无论是 Windows 程序设计的初学者,还是有经验的 Windows 程序员,利用 Delphi 都可以迅速开发出自己满意的应用程序。

为了满足广大读者的愿望,迅速的掌握 Windows 环境下的 Delphi 编程方法,我们编写了《Delphi 5.0 开发与实例》这本书,书中融合了利用 Delphi 5 开发 Windows 下应用程序的理论和实践,全面和深入的介绍了利用 Delphi 5 开发应用程序的常用方法和技巧。本书以实例为主线,以编程方法的介绍为基本,一步步教读者由浅入深地学习 Delphi 5 的开发技术,书中附带了很多的程序开发实例,所以实用性很强,书中各章的结构安排如下所示:

第 1 章介绍了利用 Delphi 5 编制应用程序的一些基础知识,以及如何利用 Delphi 5 编制简单的 Windows 应用程序;第 2 章介绍了 Delphi 5 中基本的用途和使用方法,熟练的利用 ActiveX 控件开发应用程序是每一个读者应该必备的本领;第 3 章介绍了 Delphi 5 中进行文本、图形和图象处理的常用方法;第 4 章通过几个示例程序讲解了 Delphi 5 中一些高级 ActiveX 控件使用方法,由于在 Delphi 5 中有很多的 ActiveX 控件,在这里不能够一一的加以介绍,还希望读者能够在程序开发的实践过程自己慢慢的加以体会;第 5 章介绍了 Delphi 5 中菜单、工具栏和对话框设计;第 6 章集中介绍了在 Delphi 5 中进行文件处理操作的方法;第 7 章介绍了多媒体程序开发的常用方法,在这一章中还设计了诸如 CD 播放器和动画播放器等实用程序;第 8 章集中的介绍了 Delphi 5 中有关窗体和对话框的编程技术;第 9 章介绍了 Delphi 5 中非常吸引人的网络与数据库开发方法。

本书正文部分由王兴晶同志负责编写,另外尹玉、张竞、王芊、赵春雷、贾仁健、吴为、聂乐乐、郭中华等同志参加了本书的搜集资料工作,王一样、韩雪花、冯玉、苗应那、包冬、王烁、叶盈等同志参加了本书的文字录入工作,郝婷婷、陈海华、张裕、杨华、卢兴智、霍玉华、池学英、常晓丹等同志参加了本书的校对工作,在这里一并表示感谢。

由于作者水平所限,因此书中难免会存在疏漏和错误,诚恳希望各位读者不吝赐教。

编者

1999 年 10 月

第 1 章 Delphi 5 编程基础

随着编程概念的更新，可视化编程已经成为人们关注的焦点，由 Borland Pascal 发展而来的 Delphi 程序设计语言就是一种典型的可视化编程语言，由于它继承了 Borland Pascal 程序设计语言中的优越的全功能平台，弥补了可视化编程与全功能平台之间的障碍，而且 Delphi 继承了编程效率高、代码质量好、面向对象等优点，解决了诸如 Visual Basic 等可视化程序设计语言中的代码执行效率低下等问题。

到今天为止，Delphi 程序设计语言已经发展到了 Delphi 5 版，无论是程序设计的初学者，还是大型应用程序的开发人员，利用 Delphi 5 都可以开发出满意的程序，下面就来介绍一下有关 Delphi 5 的基础知识。

1.1 Delphi 5 的安装和运行

我们假设读者对 Delphi 的前几个版本有所了解并已运行过。但是值得注意的是，要想在您的计算机上安装 Delphi 5，必须先安装相应的硬件和软件系统，这些系统要求包括：

- Windows 95、Windows 98、Windows NT 3.51 或更新版本；
- 80586 或更高微处理器；
- 一个 CD-ROM 驱动器；
- Microsoft Windows 支持的 VGA 或分辨率更高的监视器；
- 16 MB RAM；
- 154 兆以上的硬盘剩余空间；
- 鼠标或其它定点设备。

安装完成后，用鼠标单击菜单“开始”⇒“程序”⇒Borland Delphi 5⇒Delphi 5 选项就会激活 Delphi 5。

在 Delphi 5 的启动的缺省画面中，包括有主窗口、对象查看器、窗体窗口和代码窗口等。主窗口包括标题栏、菜单栏、工具栏和控件栏；对象查看器中包括两个选项——Properties 和 Events，其中 Properties 选项用于显示选中对象的属性信息，而 Events 选项用于显示可以接收的事件列表；用户可以在窗体窗口中对应用程序的界面进行设计，如添加控件等；在程序的设计阶段，当用户在窗体上单击鼠标时，就会自动的激活程序的代码窗口，用户

可以在其中添加程序设计代码。

Delphi 5 的启动欢迎画面如图 1-1 所示。

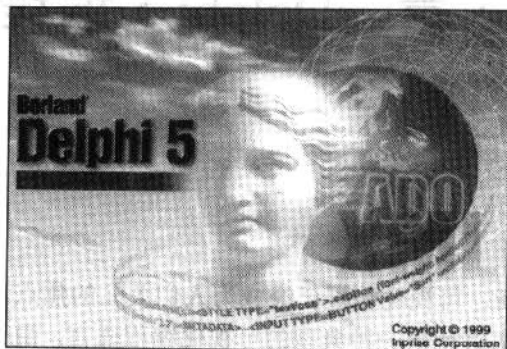


图 1-1 Delphi 5 的欢迎画面

1.2 Hello Delphi 5

下面就以一个简单的示例程序来说明，在 Delphi 5 中进行应用程序开发的一般步骤和基本的方法，为以后的程序设计打一个基础，编制示例程序的基本步骤如下所示。

(1) 开始工作

首先启动一个新的项目，选择菜单 File 中的 New Application 项，在 Delphi 5 的集成开发环境中就会弹出一个新建的窗体，在控件工具栏上选择 Standard 选项后，在 Memo 控件的图标上双击鼠标的左键，这时空白的窗体上就会出现一个 Memo 控件，接着向窗体上添加一个 Button 控件，添加控件后的窗体如图 1-1 所示。

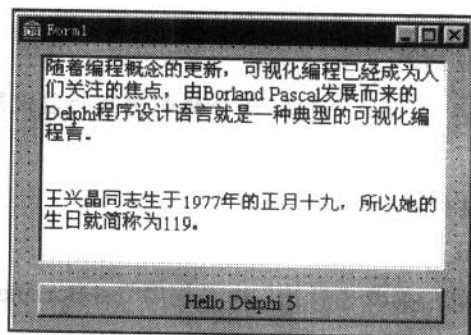


图 1-1 添加控件后的窗体

其中 Memo 控件用于显示文本，而 Button 控件用于显示一个对话框，窗体和控件的属性设置如下所示：

object Form1: TForm1
Left = 192

Top = 107
Width = 328

```

Height = 235
Caption = 'Form1'
OnCreate = FormCreate
object Memo1: TMemo
    Left = 16
    Top = 8
    Width = 289
    Height = 153
    Lines.Strings = ('随着编程概念的更新, 可视化编程已经成为人们关注的焦点, 由 Borland Pascal 发展而来的 Delphi 程序设计语言就是一种典型的可视化编程言。')
end
object Button1: TButton
    Left = 16
    Top = 176
    Width = 289
    Height = 25
    Caption = 'Hello Delphi 5'
    OnClick = Button1Click
end
end

```

(2) 程序的初始化

程序的初始化过程, 实际上就是对窗体 `FormCreate()` 事件的初始化, 在程序设计阶段, 用鼠标的左键双击窗体上的空白处, 在屏幕上就会弹出一个代码窗口, 把光标移动到 `FormCreate()` 事件的过程处理代码中, 并且添加如下所示代码:

```

procedure TForm1.FormCreate(Sender: TObject);
begin
    form1.Button1.Caption:='Click here';
    form1.Memo1.Clear;
    form1.Memo1.Lines.LoadFromFile('c:\windows\win.ini');
end;

```

在程序运行的初期, 首先执行窗体 `FormCreate()` 事件中的代码, 即通过语句 `form1.Button1.Caption:='Click here';` 来设置按钮控件的标题显示为字符串 'Click here', `form1.Memo1.Clear;` 语句用来清空文本框中的显示内容, 最后通过语句:

```
form1.Memo1.Lines.LoadFromFile('c:\windows\win.ini');
```

从文本文件 `c:\windows\win.ini` 中读取文本信息, 并且显示在文本控件中。

(3) 添加代码

在程序的设计过程中, 用鼠标的左键双击窗体上的按钮控件, 在屏幕上就会弹出一个代码窗口。在程序的设计阶段, 把光标移动到代码窗口的 `Button1Click()` 事件处理过程中, 并且添加如下所示的按钮响应代码:

```

unit Hello;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls;
type
    TForm1 = class(TForm)
        Memo1: TMemo;

```

```
Button1: TButton;  
procedure Button1Click(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;  
var  
    Form1: TForm1;  
implementation  
{$R *.DFM}  
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    showmessage('Hello Delphi 5 ');  
end;  
end.
```

在程序运行的过程中, 用户用鼠标的左键单击按钮时, 那么程序就自动激活按钮控件的 `Button1Click()` 事件, 然后通过语句 `showmessage('Hello Delphi 5');` 来显示一个如图 1-3 所示的对话框。

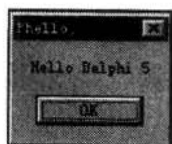


图 1-3 显示对话框信息

(4) 运行程序

做完以上的工作后, 选择菜单 `File` 中的 `Save All` 选项, 就会弹出一个如图 1-4 所示的保存文件对话框。

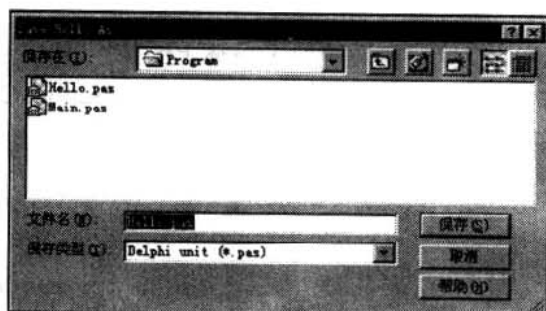


图 1-4 保存文件对话框

在弹出的对话框中选择合适的文件名保存文件。

保存程序文件后,按键盘上的功能键【F9】运行程序,在程序运行的初始画面中,文本控件显示文件 c:\windows\win.ini 中的文本信息,单击按钮控件,在窗体上就会弹出一个对话框,在其中显示一条字符串信息,程序运行的结果如图 1-5 所示。

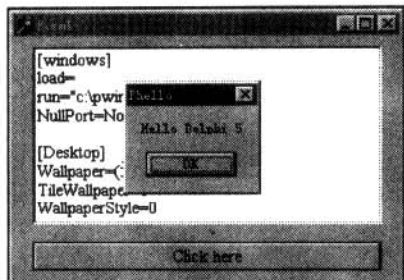


图 1-5 程序运行结果

示例程序完整源代码如下。

程序清单 1-1 Hello Delphi 5

```
unit Hello;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TForm1 = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Memo1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
showmessage('Hello Delphi 5');  
{显示一个对话框}  
end;  
  
procedure TForm1.FormCreate(Sender: TObject);  
begin  
form1.Button1.Caption:='Click here';  
//设置按钮标题  
form1.Memo1.Clear;  
//清除文本框  
form1.Memo1.Lines.LoadFromFile('c:\windows\win.ini');  
//从文本文件中读取信息  
end;  
end.
```

1.3 利用向导生成应用程序

在 Delphi 5 中系统为用户提供了许多个向导应用程序,利用这些向导程序可以节省很多的劳动,可以少用甚至不用添加代码就可以生成一个应用程序,下面就利用 Application Wizard 向导来生成一个简单的应用程序,具体的程序设计步骤如下所示。

(1) 开始工作

首先启动一个新的项目,选择菜单 File 中的 New 项,在 Delphi 5 的集成开发环境中就会弹出一个如图 1-6 标题为 New Item 的对话框,用户在其中可以选择各种向导程序和窗体形式等。

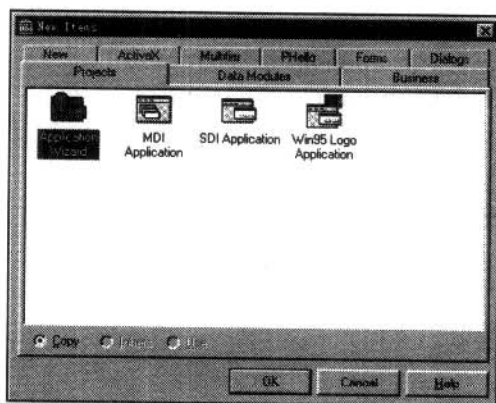


图 1-6 New Items 对话框

在 New Item 对话框中有九个选项——Projects、Data Modules、Business、New、ActiveX、Multitier、App1、Forms 和 Dialogs,而该九个选项下对应的相应选项如下所示:

- New 选项: Application、Batch File、Component、Data Module、DLL、Form、Package、Project Group、Report、Resource DLL Wizard、Service、Service Application、Text、Thread Object、Unit 和 Web Server Application 等;
- ActiveX 选项: ActiveForm、ActiveX Control、ActiveX Library、Automation Object、COM Object、Property Page 和 Type Library 等;
- Multitier 选项: CORBA Data Module、CORBA Object、MTS Data Module、MTS Object 和 Remote Data Module 等;
- Appl 选项: MainForm 等;
- Forms 选项: About box、Dual list box、QuickReport Labels、QuickReport List、QuickReport Master/Detail 和 Tabbed pages 等;
- Dialogs 选项: Dialog with Help、Dialog Wizard、Password Dialog、Reconcile Error Dialog 和 Standard Dialog 等;
- Projects 选项: Application Wizard、MDI Application、SDI Application 和 Win95 Logo Application 等;
- Data Modules 选项: Customer Data 等;
- Business 选项: Database Form Wizard、DB Web Applicati...、Decision Cube Sample、QuickReport Wizard 和 TeeChart Wizard 等;

选择 Projects 选项下的 Application Wizard，单击 OK 按钮进入下一步。

(2) 选择菜单

在由向导生成的应用程序中，用户可以在如图 1-7 所示的菜单选择对话框中选择标准菜单的类型。

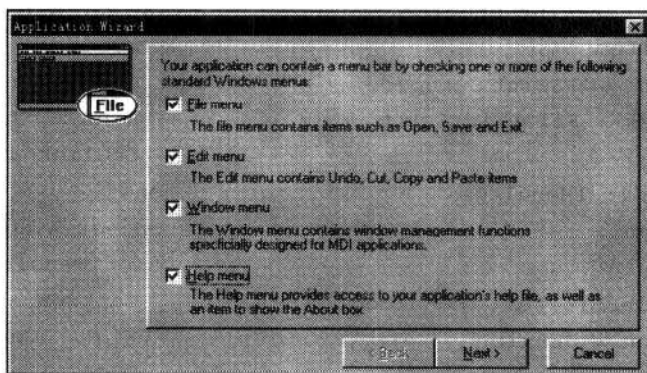


图 1-7 菜单选择窗口

在菜单选择对话框中，用户可以选择 File 菜单、Edit 菜单、Windows 菜单和 Help 菜单，其中各个菜单的属性设置如下所示：

```
object MainMenu: TMainMenu
  Left = 220
```

```
Top = 40
```

```
object FileMenu: TMenuItem
```

```
    Caption = '&File'
  object FileNewItem: TMenuItem
    Caption = '&New'
    OnClick = FileNew
  end
  object FileOpenItem: TMenuItem
    Caption = '&Open...'
    file
    OnClick = FileOpen
  end
  object FileSaveItem: TMenuItem
    Caption = '&Save'
    OnClick = FileSave
  end
  object FileSaveAsItem: TMenuItem
    Caption = 'Save &As...'
    under a new name'
    OnClick = FileSaveAs
  end
  object N1: TMenuItem
    Caption = '-'
  end
  object FilePrintItem: TMenuItem
    Caption = '&Print'
    OnClick = FilePrint
  end
  object FilePrintSetupItem:
    TMenuItem
    Caption = 'P&rint Setup...'
    OnClick = FilePrintSetup
  end
  object N4: TMenuItem
    Caption = '-'
  end
  object FileExitItem: TMenuItem
    Caption = 'E&xit'
    OnClick = FileExit
  end
end
object EditMenu: TMenuItem
  Caption = '&Edit'
  object EditUndoItem: TMenuItem
    Caption = '&Undo'
    Hint = 'Undo the last action'
    OnClick = EditUndo
  end
  object N2: TMenuItem
    Caption = '-'
  end
  object EditCutItem: TMenuItem
    Caption = 'Cu&t'
    Hint = 'Delete selected item'
    OnClick = EditCut
  end
  object EditCopyItem: TMenuItem
    Caption = '&Copy'
    Hint = 'Copy selected item to
      clipboard'
    OnClick = EditCopy
  end
  object EditPasteItem: TMenuItem
    Caption = '&Paste'
    Hint = 'Paste contents of
      clipboard'
    OnClick = EditPaste
  end
end
object WindowMenu: TMenuItem
  Caption = '&Window'
  object WindowTileItem:
    TMenuItem
    Caption = '&Tile'
    Hint = 'Tile all windows'
    OnClick = WindowTile
  end
  object WindowCascadeItem:
    TMenuItem
    Caption = '&Cascade'
    Hint = 'Cascade all windows'
    OnClick = WindowCascade
  end
  object WindowArrangeItem:
    TMenuItem
    Caption = '&Arrange All'
    OnClick = WindowArrange
  end
end
```

```

end
object HelpMenu: TMenuItem
  Caption = '&Help'
  ObjectHelpContentsItem:
    TMenuItem
      Caption = '&Contents'
      OnClick = HelpContents
    end
  end
object HelpSearchItem: TMenuItem
  Caption = '&Search for Help On...'
  OnClick = HelpSearch
end
object HelpHowToUseItem:
  TMenuItem
    Caption = '&How to Use Help'
    Hint = 'Help on using the
    help system'
    OnClick = HelpHowToUse
  end
end
object N3: TMenuItem
  Caption = '-'
end
object HelpAboutItem: TMenuItem
  Caption = '&About...'
  Hint = 'Show program
  information'
  OnClick = HelpAbout
end
end
object OpenDialog: TOpenDialog
  Left = 230
  Top = 87
end
object SaveDialog: TSaveDialog
  Left = 204
  Top = 125
end
object PrintDialog: TPrintDialog
  Left = 262
  Top = 127
end
object PrintSetupDialog:
  TPrinterSetupDialog
    Left = 276
    Top = 93
  end
end

```

相应的菜单设计结构如图 1-8 所示。

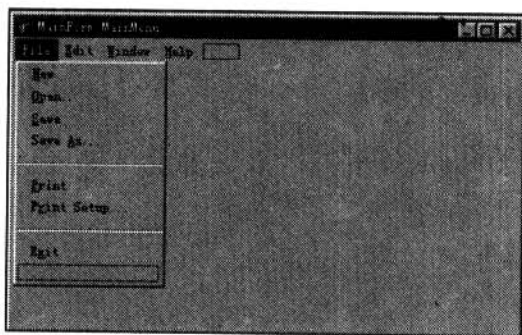


图 1-8 对应的设计菜单

在菜单选择对话框中选中四个菜单，然后用鼠标单击 Next 按钮进入下一步。

(3) 选择工具栏

在应用程序中，我们经常可以看到作为菜单加速控制的工具栏，在 Delphi 5 的向导程序中，用户也可以对工具栏进行选择操作，如图 1-9 即为一个设计工具栏的对话框，单击

Insert 按钮即可添加相应的加速按钮。

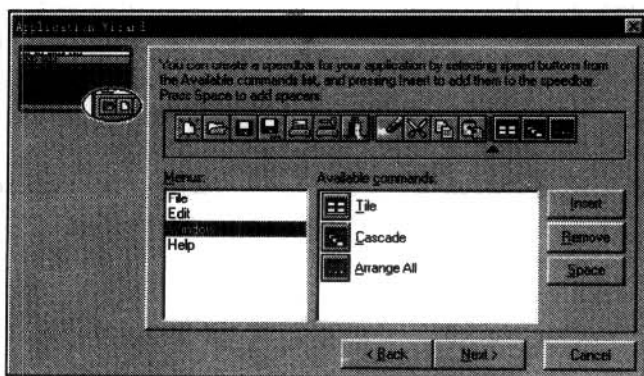


图 1-9 设计工具栏上的加速按钮

其中各个加速按钮的属性设置如下所示:

```
object SpeedBar: TPanel
  Left = 0
  Top = 0
  Width = 458
  Height = 33
  Align = alTop
  object SpeedButton1:
    TSpeedButton
      Left = 8
      Top = 4
      Width = 25
      Height = 25
      OnClick = FileNew
  end
  object SpeedButton2:
    TSpeedButton
      Left = 32
      Top = 4
      Width = 25
      Height = 25
      OnClick = FileOpen
  end
  object SpeedButton3:
    TSpeedButton
      Left = 56
      Top = 4
      Width = 25
      Height = 25
      Hint = 'Save!'
      OnClick = FileSave
  end
  object SpeedButton4:
    TSpeedButton
      Left = 80
      Top = 4
      Width = 25
      Height = 25
      Hint = 'Save As!'
      OnClick = FileSaveAs
  end
  object SpeedButton5:
    TSpeedButton
      Left = 104
      Top = 4
      Width = 25
      Height = 25
      Hint = 'Print!'
      OnClick = FilePrint
  end
  object SpeedButton6:
    TSpeedButton
      Left = 128
      Top = 4
      Width = 25
      Height = 25
      Hint = 'Print Setup!'
```

```
OnClick = FilePrintSetup
end
object SpeedButton7:
TSpeedButton
    Left = 152
    Top = 4
    Width = 25
    Height = 25
    OnClick = FileExit
end
object SpeedButton8:
TSpeedButton
    Left = 180
    Top = 4
    Width = 25
    Height = 25
    OnClick = EditUndo
end
object SpeedButton9:
TSpeedButton
    Left = 204
    Top = 4
    Width = 25
    Height = 25
    OnClick = EditCut
end
object SpeedButton10:
TSpeedButton
    Left = 228
    Top = 4
    Width = 25
    Height = 25
    OnClick = EditCopy
end
object SpeedButton11:
TSpeedButton
    Left = 252
    Top = 4
    Width = 25
    Height = 25
    OnClick = EditPaste
end
object SpeedButton12:
TSpeedButton
    Left = 280
    Top = 4
    Width = 25
    Height = 25
    Hint = 'Tile|'
    OnClick = WindowTile
end
object SpeedButton13:
TSpeedButton
    Left = 304
    Top = 4
    Width = 25
    Height = 25
    Hint = 'Cascade|'
    OnClick = WindowCascade
end
object SpeedButton14:
TSpeedButton
    Left = 328
    Top = 4
    Width = 25
    Height = 25
    OnClick = WindowArrange
end
object SpeedButton15:
TSpeedButton
    Left = 356
    Top = 4
    Width = 25
    Height = 25
    OnClick = HelpContents
end
object SpeedButton16:
TSpeedButton
    Left = 380
    Top = 4
    Width = 25
    Height = 25
    OnClick = HelpSearch
end
object SpeedButton17:
TSpeedButton
```

Left = 404	Left = 428
Top = 4	Top = 4
Width = 25	Width = 25
Height = 25	Height = 25
OnClick = HelpHowToUse	OnClick = HelpAbout
end	end
object SpeedButton18:	end
TSpeedButton	

设计完成的工具栏如图 1-10 所示。

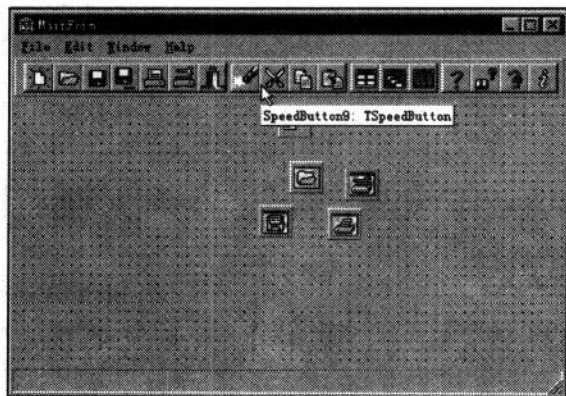


图 1-10 设计完成的工具栏

在工具栏设计对话框中用鼠标单击 Next 按钮进入下一步。

(4) 完成

在向导程序的最后一个对话框中，用户可以为生成的应用程序选择一个有效的驱动器路径和一个应用程序的文件名，如图 1-11 所示。

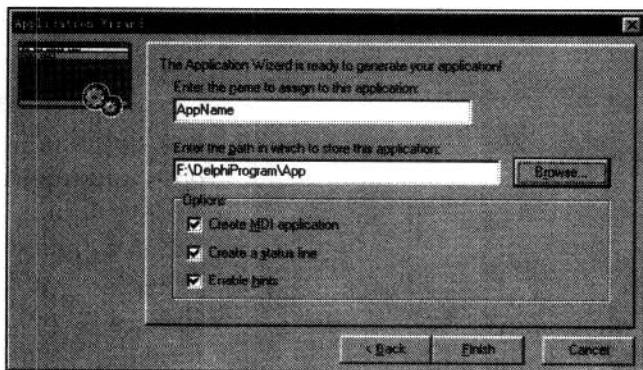


图 1-11 选择程序路径和设置

在这个对话框中不但可以选择程序的路径和存储文件名，还可以对程序的特性进行设

置。例如，是否作为一个 MDI 程序，是否建立一个状态条，在状态条中是否显示状态信息等。

在对话框中选择好应用程序的路径和文件名后，在 Options 选项卡下选中三个选项，单击 Finish 按钮就完成了应用程序的设计工作。

(5) 运行程序

做完以上的工作后，选择菜单 File 中的 Save All 选项，在弹出的对话框中选择合适的文件名保存文件，然后按键盘上的功能键【F9】运行程序，程序的运行的结果如图 1-12 所示。

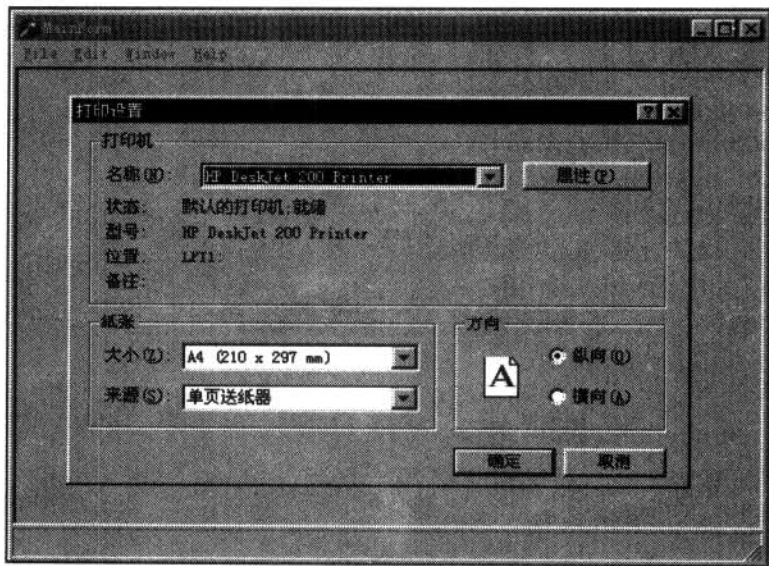


图 1-12 程序运行结果

在这个应用程序中，由于还没有添加各个事件的响应代码，所以它还不能够实现一个真正的程序功能，但是它为我们生成了一个标准的程序框架，用户可以在此基础上添加自己的代码。

由向导生成的程序完整源代码如下。

程序清单 1-2 利用向导生成的应用程序

```
unit Main;
interface
uses
  SysUtils, Windows, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Buttons, ExtCtrls, Menus, ComCtrls;
type
  TMainForm = class(TForm)
```