

精品软件“实例”丛书



Visual Basic 6.0 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 施波 等编著

精品软件“实例”丛书

Visual Basic 6.0

开发与实例

王兴晶 施 波 等编著

357/10

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是在面向对象编程思想的基础上,介绍了 ActiveX 控件使用技巧、图形和图象处理技术、VB6 中的文件处理、多媒体程序设计、窗体与对话框编程技巧、API 函数调用技巧、数据库与网络程序设计以及 VBA 高级编程实例等方面的内容,在介绍各种编程方法和常用技巧的同时,摒弃了繁冗的文字平铺直叙,而是通过程序设计实例的方法向读者介绍 VB6 编程的常见思路 and 它的高级应用,由于本书中的程序都是笔者在实践中总结提炼出来的,所以有很大的实用性。

本书讲解清晰,实例丰富,非常适合于学习 Visual Basic 6.0 的用户和提高 Visual Basic 6.0 编程技术的开发人员使用,也是大专院校、培训学校师生很好的教学参考书。

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

Visual Basic 6.0 开发与实例 /王兴晶,施波编著. -北京:电子工业出版社,1999.12

(精品软件“实例”丛书)

ISBN 7-5053-5569-4

I . V… II . ①王… ②施… III . Basic 语言-程序设计 IV . TP312

中国版本图书馆 CIP 数据核字(1999)第 64818 号

丛 书 名:精品软件“实例”丛书

书 名:Visual Basic 6.0 开发与实例

编 著 者:王兴晶 施波 等

责任编辑:嘉 益

印 刷 者:北京天竺颖华印刷厂

装 订 者:三河市金马印装有限公司

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:27.75 字数:710 千字

版 次:1999 年 12 月第一版 1999 年 12 月第一次印刷

书 号:ISBN 7-5053-5569-4
TP·2828

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

出版说明

软件是计算机的灵魂。近几年来,随着计算机的普及,各种应用与程序设计软件层出不穷,几乎每天都有新的产品面世,每年都有新的升级出台,使广大计算机用户眼花缭乱、疲于奔命,很难赶上时代发展的潮流。通过市场调研,并征求众多计算机专家、教师和读者的意见,我们依据国内软件市场及读者需求编辑出版了这套《精品软件实例丛书》,以满足广大计算机用户的需求。

本丛书以国内各行业计算机开发和应用人员为主要对象,强调以实例为主线,以实际操作为手段,以实际应用为目的。在丛书的编写过程中,采纳众多计算机专家、教师 and 用户的意见,以当前广为流行的最新软件为素材,在写作手法和内容组织上尽量做到通过典型实例,用通俗易懂的语言讲解高深的计算机技术和知识,力求做到科学性、先进性、通俗性的完美统一。本书在编写过程中,针对本套丛书的特点,特将书中制作的实例、实例的效果、实例的源代码以及相关软件程序制作成 **CD-ROM** 光盘,从而利于读者更佳地学习。

本丛书作者大多是科研院所、高等学校和计算机公司多年从事计算机教学、科研和开发应用的人员,他们知识渊博,经验丰富,了解各个专题学习和应用最有效的方法,他们将这些经验有机地融入到本丛书的各个章节中,使得本丛书更具有实用性。

我们编辑出版这套丛书,是想使广大读者以喜闻乐见的形式学习掌握流行软件的使用方法和实用技巧。本丛书适合于从初级到高级计算机用户,适合于作为各类大中专院校的教学参考书籍和专业技术的培训教材。

前 言

Visual Basic 6.0 编程语言是由 Microsoft 公司推出的可视化开发工具,利用 Visual Basic 6.0 可以开发基于 Windows 平台的 32 位应用程序,依靠强大的编译器以及网络与数据库的开发能力,用 Visual Basic 6.0 可以开发出功能强大的应用程序。

Visual Basic 6.0 所使用的程序语言是 BASIC, Visual Basic 6.0 继承了 BASIC 的简单、高效、易学易用的功能,代码结构清晰,可读性好,并且融入了面向对象、过程可视化、事件驱动等软件开发的最新技术,使 BASIC 语言编程技术发展到了一个新的高度。

Visual Basic 6.0 是开发 Windows 应用程序最为强大的工具之一,利用它可以开发出功能强大的 Windows 应用程序,无论是 Windows 程序设计的初学者,还是有经验的 Windows 程序员,利用 Visual Basic 6.0 都可以迅速地开发出自己满意的应用程序。

为了满足广大读者的愿望,迅速地掌握 Windows 环境下的 Visual Basic 6.0 编程方法,作者编写了《Visual Basic 6.0 开发与实例》这本书。本书共分九章,书中融合了利用 Visual Basic 6.0 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 Visual Basic 6.0 开发应用程序的常用方法和技巧,内容涵盖了 Visual Basic 6.0 编制应用程序的一些基础知识、图形和图象处理技术、Visual Basic 6.0 中的文件处理技术、多媒体程序设计、SDI 和 MDI 等窗体的编程技巧、VB6 中调用 API 函数的常用方法与技巧、VBA 高级编程技术、数据库与网络程序设计。由于书中附带了很多的程序开发实例,所以实用性很强。

在本书最后的附录部分,介绍了一个 Visual Basic 开发数据库应用程序的实例,取材于作者所开发的一个数据库应用项目。通过这个典型的实例,有助于读者将已学到的 VB 知识融会贯通,从而真正掌握 Visual Basic 这一目前最方便快捷的 Windows 应用程序开发语言。

本书在版面上进行了精心设计,力求清新,为突出重点和区分不同讲述内容,对操作练习内容、术语、名词解释采用了楷体。

本书正文部分由王兴晶同志负责编写,附录部分由施波同志负责编写,全书由王兴晶统稿,另外尹玉、张竞、王芊、赵济勇、赵春雷、贾仁健、吴为、葛辰、郭中华、陈兵等同志参加了本书的搜集资料工作,王一祥、李涛、韩雪花、冯玉、苗应那、包冬、聂乐乐、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、刘弘治、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 Visual Basic 6.0 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 Basic 发展而来的 Visual Basic 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 Basic 语言的所有优点，如简单、灵活等特点，同时又包括了面向对象等先进的程序设计技术，为用户提供了开发 Microsoft Windows 应用程序的最迅速、最简捷的方法。

不论是 Microsoft Windows 应用程序的专业开发人员还是初学者，Visual Basic 6.0 都为他们提供了整套工具，以方便开发应用程序。

到今天为止，Visual Basic 程序设计语言已经发展到了 Visual Basic 6.0 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 Visual Basic 6.0 都可以开发出满意的程序，下面就来介绍一下 Visual Basic 6.0(以下简称 VB6)。

1.1 VB6 的运行

值得注意的是，要想在您的计算机上安装 VB6，必须在您的计算机上先安装相应的硬件和软件系统，这些系统要求包括：

- Microsoft Windows NT 3.51 或更新版本，或 Microsoft Windows 95。
- 80486 或更高级的微处理器。
- 至少需要 50 MB 的硬盘空间。
- 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。
- 鼠标或其它定点设备。

安装完成后，用鼠标单击“开始”菜单中的“程序”⇒Microsoft Visual Studio 中的 Visual Basic 6.0 选项就会激活 VB6。

在 VB6 的启动的缺省画面中，包括有主窗口、工程资源管理器、窗体窗口、控件工具箱、属性设置窗口和代码窗口等。

主窗口包括标题栏、菜单栏和工具栏。

工程资源管理器中包括两个部分——对象列表和属性列表，其中对象列表中列举了当

前窗体上的所有对象，而属性列表则列举了相应对象所对应的属性。

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，当用户在窗体上双击鼠标时，就会自动激活程序的代码窗口，用户可以在其中添加程序设计代码。

1.2 Hello VB6

下面就以一个简单的示例程序来说明在 VB6 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

首先用鼠标选择 Windows 操作系统“开始”菜单中的“程序”⇒Microsoft Visual Studio 中的 Microsoft Visual Basic 6.0 选项来激活 VB6 应用程序，在 VB6 的集成开发环境中用鼠标选择 File 菜单中的 New Project 选项，在屏幕上就会弹出一个如图 1-1 所示的 New Project 对话框。

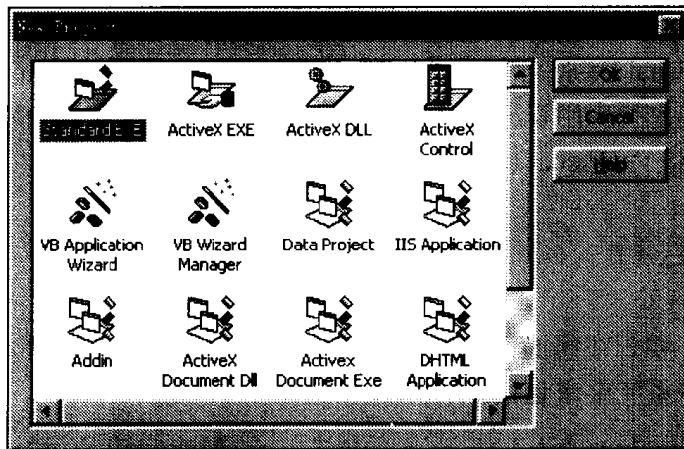


图 1-1 New Project 对话框

在 New Project 对话框中选择 Standard EXE 选项，单击 OK 按钮，在 VB6 中就新建了一个标准的工程文件，同时打开了一个空白的窗体。

窗体的属性设置如下所示：

```
Begin VB.Form Form1
    Caption           =   "Hello VB6"
    ClientHeight      =   3195
    ClientLeft        =   60
    ClientTop         =   345
    ClientWidth       =   4680
    ScaleHeight       =   3195
```

```

ScaleWidth      = 4680
StartPosition   = 2
End

```

经过以上属性设置后的窗体具有如下所示的特性：

- 窗体的标题栏中显示文本“Hello VB6”；
- 窗体的标题栏中没有最大化和最小化按钮，只有关闭按钮。

(2) 添加控件

在程序设计的过程中，向当前空白的窗体添加两个 CommandButton 控件和两个 TextBox 控件，它们的作用在后面会加以说明，添加控件后的窗体如图 1-2 所示。

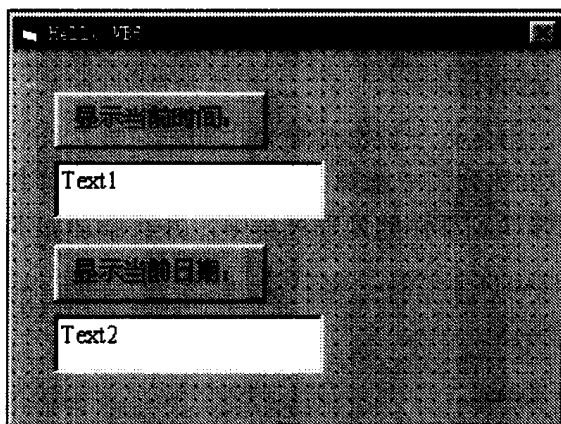


图 1-2 添加控件后的窗体

各个控件的属性设置如下所示：

```
Begin VB.TextBox Text2
```

```
    Height = 495
```

```
    Left = 360
```

```
    Text = "Text2"
```

```
    Top = 2280
```

```
    Width = 2295
```

```
End
```

```
Begin VB.TextBox Text1
```

```
    Height = 495
```

```
    Left = 360
```

```
    Text = "Text1"
```

```
    Top = 960
```

```
    Width = 2295
```

```
End
```

```
Begin VB.CommandButton
```

```
    Command2
```

```
    Caption = "显示当前日期:"
```

```
    Height = 495
```

```
    Left = 360
```

```
    Top = 1680
```

```
    Width = 1815
```

```
End
```

```
Begin VB.CommandButton
```

```
    Command1
```

```
    Caption = "显示当前时间:"
```

```
    Height = 495
```

```

Left      =      360                      End
Top       =      360                      End
Width     =      1815

```

经过以上属性设置后的 ActiveX 控件的特性及作用如下所示:

- Text1 控件: 在程序运行的过程中充当系统时间显示的容器;
- Text2 控件: 在程序运行的过程中充当系统日期显示的容器;
- Command1 控件: 激活控件 Text1 显示系统时间;
- Command2 控件: 激活控件 Text2 显示系统日期。

(3) 程序初始化

在本示例程序中, 所谓程序初始化, 指的就是为窗体 Private Sub Form_Load() 事件和窗体声明段中所添加的响应代码。

在程序设计的过程中, 用鼠标左键双击窗体上的空白处, 在屏幕上就会弹出一个空白的代码窗口, 在代码窗口的事件列表中选择窗体 Form, 在对应的事件列表中选择事件 Private Sub Form_Load(), 把光标移动到事件的处理过程中, 并且添加如下所示的事件响应代码:

```

Private Sub Form_Load()
Form1.Text1.Text = Time
Form1.Text2.Text = Date
'显示时间和日期
End Sub

```

在程序运行的初期, 事件 Private Sub Form_Load() 中的代码就会被执行, 程序通过 Form1.Text1.Text = Time、Form1.Text2.Text = Date 两条语句在窗体的 Text1 和 Text2 中显示程序运行初期的系统时间和日期。

(4) 响应按钮操作

在示例程序运行的过程中, 如果用户在两个按钮上按下鼠标的左键, 在对应的文本框中就会显示当前系统的时间和日期。

为此, 在程序设计的过程中, 用鼠标左键双击窗体上的两个按钮控件, 在屏幕上就会弹出一个空白的代码窗口, 在代码窗口的对象列表中分别选择按钮控件 Command1 和 Command2, 在对应的事件列表中选择事件 Private Sub Command1_Click() 和 Private Sub Command2_Click()。把光标移动到相应事件的处理过程中, 添加如下所示的事件响应代码:

```

Private Sub Command1_Click()
Form1.Text1.Text = Time
'显示当前时间
End Sub

```

```
Private Sub Command2_Click()

Form1.Text2.Text = Date

'显示当前日期

End Sub
```

在程序运行的过程中,当用户在按钮控件 Command1 上按下鼠标左键时,就会激活控件的 Private Sub Command1_Click()事件,程序通过 Form1.Text1.Text = Time 语句在控件 Text1 上显示系统当前时间,如果按钮控件 Command2 上按下鼠标左键时,就会激活控件的 Private Sub Command2_Click()事件,程序通过 Form1.Text2.Text = Date 语句在控件 Text2 上显示系统当前日期。

(5) 运行程序

按照附后的源程序清单 1-1 添加剩余的代码,设置项目的启动窗体为主窗体 Form1 并存储文件,然后运行程序,程序运行结果如图 1-3 所示。

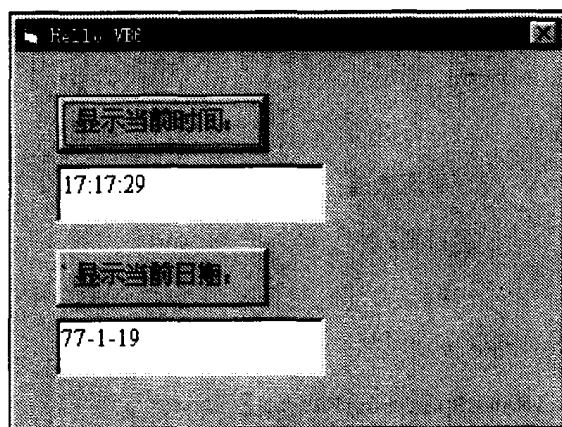


图 1-3 程序运行结果

该示例程序的完整源代码如下所示:

程序清单 1-1 Hello VB6

VERSION 5.00

Begin VB.Form Form1

Caption = "Hello
VB6"

ClientHeight = 3195

ClientLeft = 60

ClientTop = 345

ClientWidth = 4680

ScaleHeight = 3195

ScaleWidth = 4680

StartPosition = 2

Begin VB.TextBox Text2

Height = 495

Left = 360

TabIndex = 3

```
Text = "Text2" Left = 360
Top = 2280 TabIndex = 1
Width = 2295 Top = 1680
End Width = 1815
Begin VB.TextBox Text1 End
Height = 495 Begin VB.CommandButton
Left = 360 Command1
TabIndex = 2 Caption = "显示当前时
Text = "Text1" 间: "
Top = 960 Height = 495
Width = 2295 Left = 360
End TabIndex = 0
Begin VB.CommandButton Top = 360
Command2 Width = 1815
Caption = "显示当前 End
日期: " End
Height = 495
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Command1_Click()
Form1.Text1.Text = Time
'显示当前时间
End Sub

Private Sub Command2_Click()
Form1.Text2.Text = Date
'显示当前日期
End Sub
```

```
Private Sub Form_Load()  
Form1.Text1.Text = Time  
Form1.Text2.Text = Date  
'显示时间和日期  
End Sub
```

1.3 利用向导生成应用程序

在 VB6 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 VB 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 Project 项, 在 VB6 的集成开发环境中就会弹出一个如图 1-4 标题为 New Project 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。

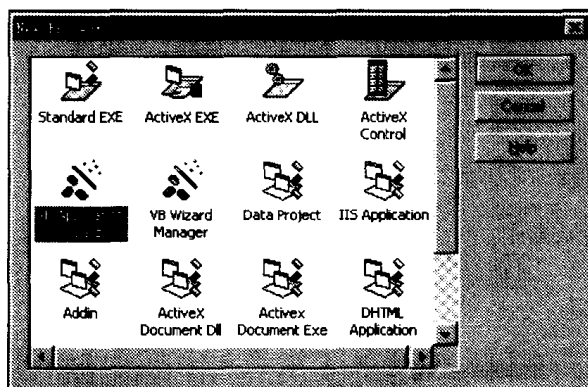


图 1-4 New Project 对话框

在 New Project 对话框中有 13 个选项——Standard EXE、ActiveX EXE、ActiveX DLL、ActiveX Control、VB Application Wizard、VB Wizard Manager、Data Project、IIS Application、Addin、ActiveX Document Dll、Activex Document Exe、DHTML Application 和 VB Enterprise Edition Controls。选择 New Project 对话框中的 VB Application Wizard, 单击 OK 按钮进入下一步。接下来就会显示一个 Introduction 对话框, 在其中可以读取以前存储过的设置, 单击 Next 按钮进入下一步。

(2) 选择界面类型

在由向导生成的应用程序中, 用户可以在如图 1-5 所示的窗体界面类型选择对话框中选择界面的类型。

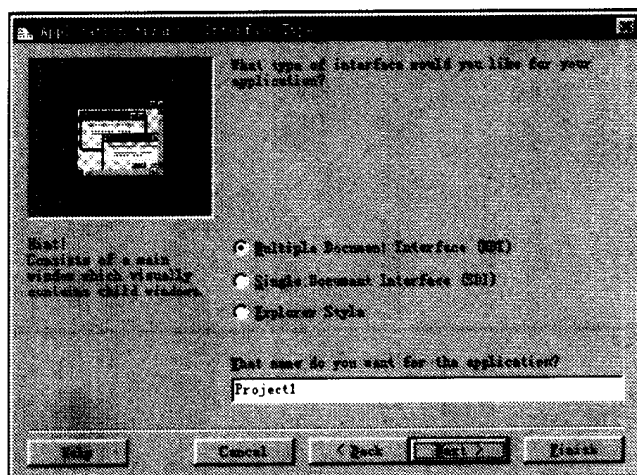


图 1-5 窗体界面类型选择对话框

在窗体界面类型选择对话框中用户可以为自己的应用程序选择一种界面类型，如 MDI、SDI 或者是资源管理器样式等，另外还可以在文本框中为应用程序起一个名字。在文本输入框中输入应用程序的名字为 Project1，选择界面类型为 MDI，单击 Next 按钮进入下一步。

(3) 选择菜单

在如图 1-6 所示的菜单选择对话框中，用户可以选择 File 菜单、Edit 菜单、Windows 菜单、View 菜单、Tools 菜单和 Help 菜单以及其中的子菜单项。

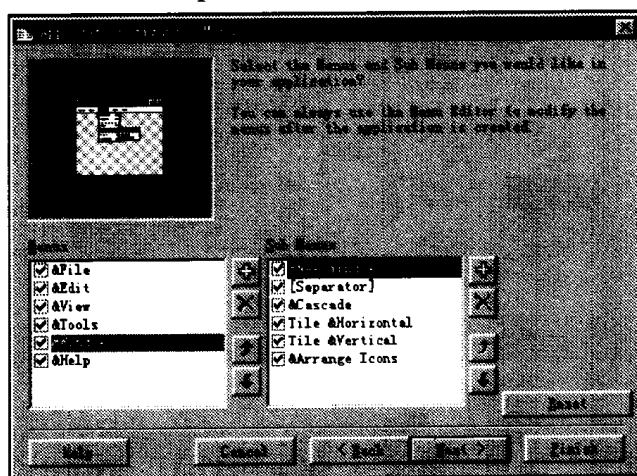


图 1-6 菜单选择对话框

在菜单选择对话框中选中六个菜单以及其中的子菜单项，相应的菜单设计结构如图 1-7 所示。

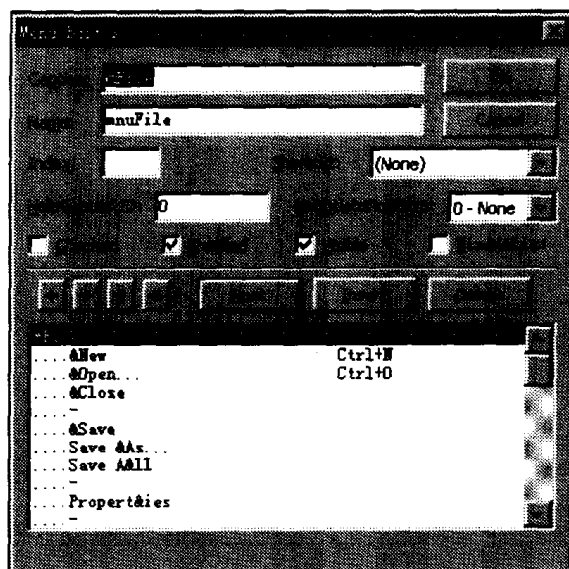


图 1-7 对应的设计菜单

用鼠标单击 Next 按钮进入下一步。

(4) 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 VB6 的向导程序中，用户也可以对工具栏进行选择操作。

在如图 1-8 所示的工具栏设置对话框中用户可以自己添加可以连接菜单操作的工具栏按钮。

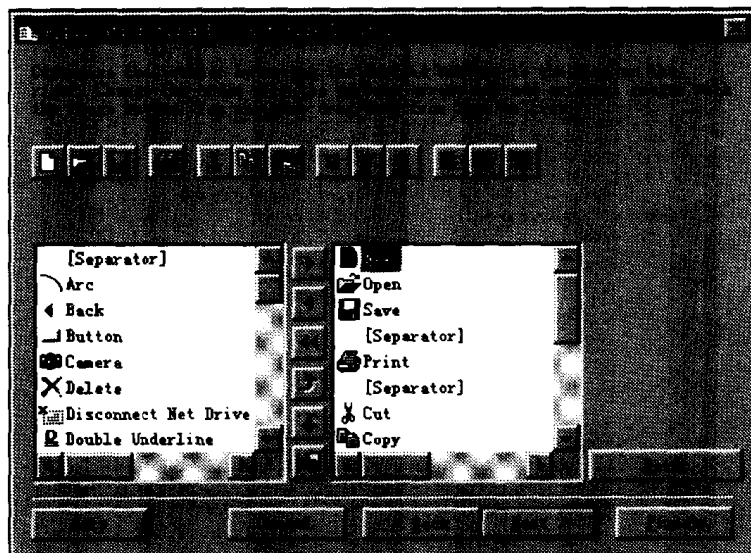


图 1-8 设计工具栏上的加速按钮

在本示例程序中，系统已经为我们缺省的设置了一些常用的工具栏按钮，我们可以直接单击对话框中的 Finish 按钮完成应用程序的设计。

经过以上各步完成的应用程序窗体如图 1-9 所示。

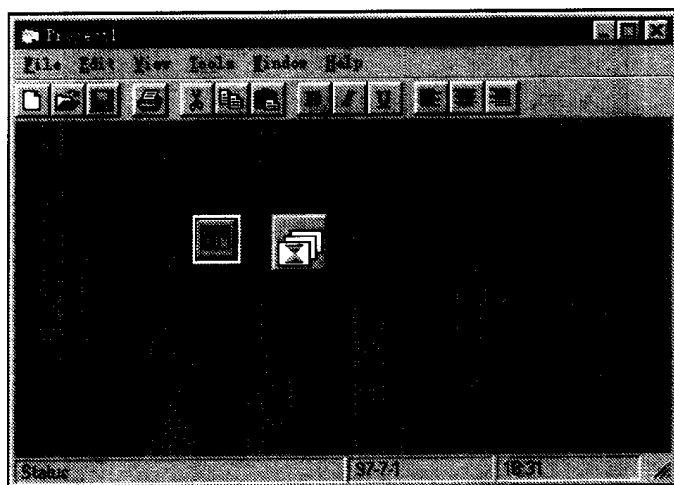


图 1-9 选择程序路径和设置

在这个我们没有填写一行代码的应用程序中，已经包括了菜单、工具栏和状态条等 Windows 应用程序的标准设置了。

(5) 运行程序

做完以上的工作后，选择 VB6 工具栏上的 Save 按钮，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 F5 运行程序，程序的运行的结果如图 1-10 所示。

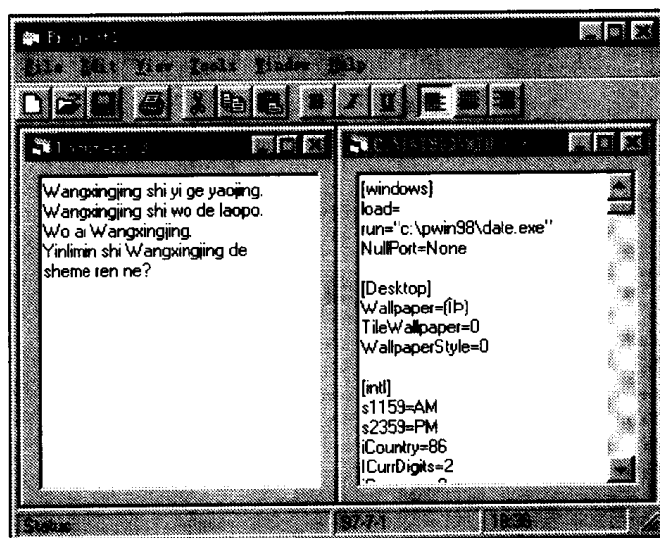


图 1-10 程序运行结果

在这个应用程序中，可以完成新建文件、打开指定文件、存储文件，对文本文件进行编辑等操作，但是由于还没有添加各个事件的详细响应代码，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所示:

程序清单 1-2 利用向导生成的应用程序

主窗体中的响应代码:

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal
hwnd As Long, ByVal wParam As Long, ByVal lParam As Any) As
Long
```

```
Const EM_UNDO = &HC7
```

```
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal
hwnd%, ByVal HelpFile$, ByVal wCommand%, dwData As Any)
```

```
Private Sub MDIForm_Load()
```

```
Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
```

```
Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
```

```
Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
```

```
Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
```

```
LoadNewDoc
```

```
End Sub
```

```
Private Sub LoadNewDoc()
```

```
Static lDocumentCount As Long
```

```
Dim frmD As frmDocument
```

```
lDocumentCount = lDocumentCount + 1
```

```
Set frmD = New frmDocument
```

```
frmD.Caption = "Document " & lDocumentCount
```

```
frmD.Show
```

```
End Sub
```

```
Private Sub MDIForm_Unload(Cancel As Integer)
```

```
If Me.WindowState <> vbMinimized Then
```

```
SaveSetting App.Title, "Settings", "MainLeft", Me.Left
```

```
SaveSetting App.Title, "Settings", "MainTop", Me.Top
```

```
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "New"
            LoadNewDoc
        Case "Open"
            mnuFileOpen_Click
        Case "Save"
            mnuFileSave_Click
        Case "Print"
            mnuFilePrint_Click
        Case "Cut"
            mnuEditCut_Click
        Case "Copy"
            mnuEditCopy_Click
        Case "Paste"
            mnuEditPaste_Click
        Case "Bold"
            ActiveForm.rtfText.SelBold = Not ActiveForm.rtfText.SelBold
            Button.Value = IIf(ActiveForm.rtfText.SelBold, tbrPressed,
                               tbrUnpressed)
        Case "Italic"
            ActiveForm.rtfText.SelItalic = Not ActiveForm.rtfText.SelItalic
            Button.Value = IIf(ActiveForm.rtfText.SelItalic, tbrPressed,
                               tbrUnpressed)
        Case "Underline"
            ActiveForm.rtfText.SelUnderline = Not
```