

软件开发技术丛书



Visual FoxPro 6

专业版基础类 参考手册

李春葆 编著



2.3
1

机械工业出版社
China Machine Press

软件开发技术丛书

Visual FoxPro 6专业版 基础类参考手册

李春葆 编著



机械工业出版社
China Machine Press

在Visual Foxpro 6专业版中提供了一组功能强大的基础类库, 这些类库中包含实现各种复杂功能(如数据导航、多媒体操作和Internet发布等)的基础类。本书全面讨论了这些基础类的设计和应用方法。

本书内容详尽, 阐述了关于基础类完整的参考信息, 包括其类名、基类、类库、父类、样例目录、包含对象、属性及方法说明等方面。此外, 还提供了类方法的设计代码, 对于想深入了解基础类设计过程的读者, 通过这些代码可以掌握它们的设计原理。在每个基础类的说明后面, 还列出了深入的应用实例, 每个实例均经过了上机调试, 读者可直接套用。

本书中文简体字版由机械工业出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

版权所有, 翻印必究。

图书在版编目(CIP)数据

Visual FoxPro6专业版基础类参考手册/李春葆编著.-北京: 机械工业出版社, 1999.9
(软件开发技术丛书)

ISBN 7-111-07366-5

I. V... II. 李... III. 关系数据库-数据库管理系统, FoxPro6-参考资料 IV. TP311.13

中国版本图书馆CIP数据核字(1999)第29328号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 陈 谊

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

1999年9月第1版第1次印刷

787mm×1092mm 1/16·22印张

印数: 0 001-6 000册

定价: 35.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

前言

Visual FoxPro 6专业版提供了一个功能强大的基础类库，它们位于\Ffc\文件夹中。我们可以不经改动或只需做很少的修改便将它们应用在系统开发中，从而大大提高软件的开发效率。本书对这些基础类进行了深入的分析，不仅给出了它们的参考信息，而且提供了它们的设计过程。这使读者可以充分了解这些基础类的功能及使用方法，正如本书中每个基础类后面的应用实例所示，很短的程序却建立了功能很强的应用程序。总之，Visual FoxPro 6提供的一些新功能和改进，在这些基础类设计中都可以见到，这是在一般Visual FoxPro编程书中难以找到的。较好地掌握它们的设计过程对于更高程度地应用这些基础类是十分有帮助的。

类的类型

Visual FoxPro 6的全部基类分为三部分：

- A:可以拖到一个表单的基类。

复选框	组合框	命令按钮	命令组
编辑框	表格	超级挂接	图像
标签	线条	列表框	OLE控件
选项钮	选项组	形状	微调
文本框			

- B:可以拖到一个表单或在应用程序中运行的基类。

定制	容器	计时器	项目挂接
活动文档			

- C:只能在应用程序中运行的基类。

表单	表单集	工具栏
----	-----	-----

Visual FoxPro 6基础类也是基于某个基类的，因此，根据其基类的可视性，这些基础类也具有同样的特性。

把基础类添加到表单中

我们可以通过Visual FoxPro 6的组件管理器、类浏览器、项目管理和表单控件工具栏将基础类添加到表单中。

- 组件管理器 对于基于A和B类的基础类，拖着该基础类放到一个表单上；或者在该基础类上单击右键显示项目快捷菜单，从中选择“添加到表单”即可(该表单应处于激活状态)。在放入表单后相关设计器将自动启动。

- 类浏览器 对于基于A和B类的基础类，直接单击该基础类左上角的图标将其拖到一个表单中。在放入表单后不会自动启动相关设计器。如果要启动相关设计器，只需单击右键，在项目快捷菜单中选择“设计器”即可。

- 项目管理器 对于基于A和B类的基础类，可以将其先拖放到项目管理器中，然后再拖放到一个表单中。在放入表单后不会自动启动相关设计器。如果要启动相关设计器，只需单击右键，在项目快捷菜单中选择“设计器”即可。

• **表单控件工具栏** 对于基于A和B类的基础类，可以将其先添加到表单控件工具栏中，然后再添加到一个表单中。如果设计器没有被锁定，则自动启动相关设计器；如果设计器被锁定，只需单击右键，在项目快捷菜单中选择“设计器”即可启动相关的设计器。

把基础类添加到项目中

当把一个包含基础类的表单添加到一个Visual FoxPro项目中时，项目管理器自动添加包含该基础类的可视类库。但对于基于C类的基础类，可以通过以下方法添加：

• **从“组件管理器”中添加基础类** 可以直接把一个基础类添加到一个项目中，或者也可以通过单击右键，选择“添加到项目”把该基础类添加到一个项目中。当把一个基础类添加到一个项目时，将会出现一个“添加类到项目”对话框，提示以下选择：

添加到项目 将基础类和它的.vcx类库同时添加到这个项目中。基础类将自动对于基于A和B类的基础类，添加到表单中。但对于有些基于B和C类的基础类，在你的应用程序中调用它们时，你需要选择这个选项。

从选择的类建立新类 从基础类建立一个子类。这常常发生在需要修改该基础类时。

从选择的类建立新表单 这常常发生在基于表单的基础类中，你可以修改该基础类。

• **从Windows Explorer中添加基础类** 从Windows Explorer中选择包含该基础类的可视类库，将其添加到项目管理器中，该可视类库便添加到项目管理器的类库项中了。

• **在项目管理器中添加基础类** 在项目管理器中选择“类”选项卡，单击“添加”按钮，然后从Vfc\文件夹中选择相应的类库；单击“确定”即可。

把基础类结合到应用程序中

大多数情况下，基础类不需要另外的程序，有时仅需要很小的代码。下面通过一个例子说明在应用程序中使用不可视基础类的程序代码：

• 从“组件管理器”中将“文件版本”（参见本书第19小节）添加到一个表单中。

• 添加一个命令按钮，在其上的Click事件中建立代码：

```
with thisform._fileversion1
    .cFileName=HOME()+ 'VFP6.EXE'
    .GetVersion()
    .DisplayVersion()
endwith
```

• 要运行该表单，只需单击该命令按钮即可。

你也可以用如下程序实现上述功能：

```
set classlib to ffc\_utility.vcx          &&指出类库
myob=newobject( "_fileversion" )        &&建立对象
with myob
    .cFileName=HOME()+ ' VFP6.EXE'
    .GetVersion()
    .DisplayVersion()
endwith
```

对于可视类，亦可采用如上程序设计方法，详见本书的应用实例。

由于编者水平所限，加之时间仓促，书中可能存在错误和不当之处，欢迎读者朋友指正。

1999年5月

目 录

前言

1. About对话框(About Dialog)	1
2. ActiveX日历(ActiveX Calendar)	3
3. 应用注册(Application Registry)	4
4. 数组处理器(Array Handler)	6
5. Cancel按钮(Cancel Button)	10
6. 时钟(Clock)	11
7. 冲突捕捉器(Conflict Catcher)	12
8. Cookies类(Cookies)	16
9. 交叉表(Cross Tab)	20
10. 数据编辑按钮(Data Edit Buttons)	22
11. 数据导航按钮(Data Navigation Buttons)	25
12. 数据段管理器(Data Session Manager)	28
13. 数据有效性(Data Validation)	41
14. 数据导航对象(Data Navigation Object)	45
15. 从DBF转换到HTML(DBF → HTML)	48
16. 相异值组合框(Disinct Values Combo)	50
17. 错误对象(Error Object)	52
18. 字段移动器(Field Mover)	62
19. 文件版本(File Version)	70
20. 过滤器对话框按钮(Filter Dialog Button)	72
21. 过滤器对话框(Filter Dialog Box)	73
22. 过滤器表达式对话框(Filter Expression Dialog)	81
23. 查找按钮(Find Button)	85
24. 查找对话框(Find Dialog Box)	87
25. 查找文件/文本(Find Files/Text)	90
26. 查找对象(Find Object)	92
27. 查找(查找下一个)按钮(Find(Findnext)Buttons)	95
28. 字体组合框(Font Combobox)	97
29. 字体尺寸组合框(Font Size Combobox)	98
30. 格式工具栏(Format Toolbar)	100
31. 从FRX转换到HTML(FRX → HTML)	100
32. 获取文件和目录(GetFile and Directory)	102
33. GoTo对话框按钮(GoTo Dialog Box Button)	103

34. GoTo对话框(GoTo Dialog Box)	105
35. 用记录画图(Graph By Record)	106
36. 图形对象(Graph Object)	109
37. 帮助按钮(Help Button)	111
38. 超链按钮(Hyperlink Button)	112
39. 超链图像(Hyperlink Image)	115
40. 超链标签(Hyperlink Label)	117
41. 项定位器(Item Locater)	119
42. INI访问(INI Access)	120
43. 关键字对话框(Keywords Dialog Box)	124
44. 启动按钮(Launch Button)	127
45. 定位按钮(Locate Button)	128
46. 寻找组合框(Lookup Combobox)	128
47. 邮件合并对象(Mail Merge Object)	130
48. 消息框处理器(Message Box Handler)	143
49. 移动器(Mover)	145
50. 鼠标置上效应(MouseOver Effects)	149
51. 导航快捷菜单(Menus)	152
52. 导航工具栏(Data Navigation)	156
53. 对象状态(Application)	158
54. ODBC注册(System Utilities)	162
55. 脱机开关(Offline Switch)	165
56. OK按钮(Ok Button)	169
57. 输出对象(Output Object)	170
58. 输出控件(Output Control)	183
59. 输出对话框(Output Dialog Box)	184
60. 口令对话框>Password Dialog Box)	189
61. 透视表(Pivot Table)	191
62. 预览报表(Preview Report)	202
63. QBF(QBF)	204
64. 注册访问(Registry Access)	206
65. 改变尺寸对象(Resize Object)	216
66. RTF控件(RTF Controls)	219
67. 运行表单按钮(RUN Form Button)	220
68. 从SCX转换到HTML(SCX → HTML)	221
69. 发送邮件按钮(SendMail Buttons)	224
70. 外壳执行(Shell Execute)	225
71. 快捷菜单类(Shortcut Menu Class)	227
72. 简单编辑按钮(Simple Edit Buttons)	233

73. 简单图像导航按钮(Simple Picture Navigation Buttons)	234
74. 简单导航按钮(Simple Navigation Buttons)	235
75. 排序对话框按钮(Sort Dialog Button)	236
76. 排序对话框(Sort Dialog)	237
77. 排序对象(Sort Object)	239
78. 排序选择器(Sort Selector)	243
79. 排序移动器(Sort Mover)	244
80. 声音播放器(Sound Player)	247
81. 闪烁屏幕(Splash Screen)	252
82. 通过SQL传送(SQL Pass Through)	253
83. 秒表(Stop Watch)	255
84. 字符串库(String Library)	257
85. 超级移动器(Super Mover)	259
86. 系统工具栏(System Toolbar)	260
87. 表移动器(Table Mover)	262
88. 文本预览(Text Preview)	274
89. 温度计(Thermometer)	277
90. 跟踪已知定时器(Trace Aware Timer)	280
91. 类型库(Type Library)	282
92. URL组合框(URL Combo)	290
93. URL打开对话框(URL Open Dialog)	294
94. VCR按钮(VCR Buttons)	295
95. VCR图像导航按钮(VCR Picture Navigation Buttons)	296
96. 视频播放器(Video Player)	297
97. Web浏览器控件(Web Browser Control)	303
98. 窗体处理器(Window Handler)	336

1. About对话框(About Dialog)

名称	About Dialog
类别	Misc Forms
缺省目录	Visual FoxPro Catalog\Foundation Classes\Dialogs
类	_aboutbox
基类	Form
类库	_dialogs.vcx
父类	_form
样例目录	...\Samples\Vfp98\Solution\Ffc\dialogs.scx
说明	

该类用于定制一个简单的About对话框模板，该模板包括一个使用Registry类访问系统信息的按钮。

使用它时，把该类拖动到一个项目上，或者从“组件管理库”的项目快捷菜单中选择“添加到项目”。当你把该类拖到一个项目后，可以选择“添加该类”、“建立一个子类”或“建立一个表单”。如果选择“建立一个表单”，Visual FoxPro将显示Open对话框以便你指定该表单的名称，然后在表单设计器中建立和打开该表单。指定标签的值，然后保存并从你的应用中调用该表单。

包含对象 无

属性、事件和方法说明

cMSINFODir属性	指出Msinfo.exe的路径，缺省值为“”。
GetRegisteredCompany方法	一个返回注册公司的存根例程。使用它时，替换屏幕字段lblUserCorp的Caption属性值。
GetRegisteredOwner方法	一个返回注册拥有者的存根例程。使用它时，替换屏幕字段lblUserName的Caption属性值。

实现方法的过程

```
PROCEDURE Init
    parameter tcAppName, tcVersion, tcCopyright, tcTrademark, tcLogoBMP
    #define HKEY_LOCAL_MACHINE -2147483646
    #define KEY_WIN4_MSINFO "Software\Microsoft\Shared Tools\MSInfo"
    local lnError, lcMSINFO
    *如果参数没有包含.F.，则设置新的属性值
    If TYPE("tcAppName") = "C"
        thisform.Caption = "About " + ALLTRIM(tcAppName)
        thisform.lblAppName.Caption = ALLTRIM(tcAppName)
    endif
    If TYPE("tcVersion") = "C"
        thisform.lblVersion.Caption = "Version " + ALLTRIM(tcVersion)
    endif
    If TYPE("tcCopyright") = "C"
        thisform.lblCopyright.Caption = ALLTRIM(tcCopyright)
    endif
    If TYPE("tcTrademark") = "C"
        thisform.lblTrademark.Caption = tcTrademark
        thisform.lblTrademark.WordWrap = 1
    endif
endIf
```

```

If TYPE("tcLogoBMP") = "C"
    thisform.imgLogo.Picture = tcLogoBMP
endif
thisform.lblUserName.Caption = this.GetRegisteredOwner()
thisform.lblUserCorp.Caption = this.GetRegisteredCompany()
lcMSINFO = ""
lnError = this.Registry.GetRegKey("Path", @lcMSINFO, KEY_WIN4_MSINFO, HKEY_LOCAL_MACHINE)
If lnError = 0
    thisform.cmsInfoDir=lcMSINFO &&设置cmsInfoDir属性
endif
*检查MSINFO.EXE是否存在, 如果不存在则删除系统信息命令按钮并缩短该表单
If EMPTY(thisform.cmsInfoDir)
    thisform.cmsInfoDir = " "
    thisform.cmdSysInfo.Enabled = .F.
    thisform.Height = 282
endif
endif
ENDPROC

PROCEDURE getregisteredcompany
    return thisform.lblUserCorp.Caption
ENDPROC

PROCEDURE getregisteredowner
    return thisform.lblUserName.Caption
ENDPROC

```

应用实例

如下程序显示一个有关版本方面的信息。

```

*文件名:exam1.prg
set classlib to ffc\_dialogs.vcx
myfr=createobject('\_aboutbox', "示例程序1", "1.0", "版权所有", "1999.5.20")
myfr.lblusername.caption="李春葆"
myfr.lblusercorp.caption="武汉测绘科技大学"
myfr.show
return

```

程序的执行屏幕如图1-1所示。



图1-1 程序执行屏幕

2. ActiveX 日历(ActiveX Calendar)

名称	ActiveX Calendar
类别	Date/Time
缺省目录	Default Catalog Visual FoxPro Catalog\Foundation Classes\Data/Time
类	_olecalendar
基类	OLEcontrol
类库	_datetime.vcx
父类	_olecontrol
样例目录	...\Samples\Vfp98\Solution\Ffc\dataacal.scx
说明	

该类通过使用ActiveX Calendar控件，在表单或其它容器中，把该控件绑定到一个日期字段上。

使用它时，把该类拖动到一个项目上，或者从“组件管理库”的项目快捷菜单中选择“添加到表单”。当你把该类添加到一个表单时，Visual FoxPro打开一个设计器以便你指定适当的Data_column值。当你把该类拖到一个项目时，可以选择“添加该类”或“建立一个子类”。

包含对象 无

属性、事件和方法说明

Date_column属性	指出链接到该日历的一个日期类型字段。缺省值为“”。
RefreshDisplay方法	用Date_column属性中指定列中的值修改日历显示。 语法:RefreshDisplay() 返回值:无 参数:无

实现方法的过程

```
PROCEDURE _olecalendar.refreshdisplay
    local cColumn
    cColumn = this.date_column
    If !EMPTY(cColumn)
        this.Month = MONTH(&cColumn)
        this.Day = DAY(&cColumn)
        this.Year = YEAR(&cColumn)
    endif
ENDPROC
```

应用实例

如下程序对一个自由表student.dbf的birthday日期字段进行处理，先显示该表第一个记录的birthday字段值(1964.02.20)，用户可进行修改，其修改将反映到表中。

```
*文件名:exam2.prg
myfr = createobject("form")
myfr.NewObject("oleCalendar", "_olecalendar", "_datetime")
with myfr.oleCalendar
    .date_column = "student.birthday"
    .RefreshDisplay()
    .Visible = .T.
endwith
```

```

with myfr
    .BorderStyle = 2
    .MaxButton = .F.
    .MinButton = .F.
    .Height = myfr.oleCalendar.Height
    .Width = myfr.oleCalendar.Width
    .Caption = "示例程序2"
endwith
myfr.Show(1)
return

```

本程序的执行屏幕如图2-1所示。



图2-1 程序执行屏幕

3. 应用注册(Application Registry)

名称	Application Registry
类别	System Utilities
缺省目录	Visual FoxPro Catalog\Foundation Classes\Utilities
类	filereg
基类	Custom
类库	registry.vcx
父类	registry
样例目录	...\Samples\Vfp98\Solution\WinAPI\regfile.scx
说明	

该类提供了一组返回应用的特定信息，如版本和位置的注册函数。

使用它时，把该类拖动到一个项目上，或者从“组件管理库”的项目快捷菜单中选择“添加到项目”或“添加到表单”。当你把该类拖到一个表单时，Visual FoxPro把该类放在表单上，然后指定适当的值和提供必要的输入与输出对象。当你把该类拖到一个项目时，可以选择“添加该类”或“建立一个子类”。

包含对象 无

属性、事件和方法说明

GetAppPath方法

检查并返回与特定扩展名关联的应用的路径。

语法: GetAppPath(cExtension, cExtnKey, cAppKey, lServer)

返回值: GetApplication(cExtnKey, @cAppKey, lServer)

参数:

cExtension 指出应用的扩展名

cExtnKey 指出应用的注册关键字

cAppKey 指出应用的路径

lServer 指出扩展名是否为一个服务器应用

GetLatestVersion方法

返回一个指定应用的最新版本。

语法: Getlatestversion(cClass, cExtnKey, cAppKey, lServer)

返回值: cExtnKey, @cAppKey, lServer

参数:

cClass 指出应用的类关键字
 cExtension 指出应用的扩展名
 cExtnKey 指出应用的注册关键字
 cAppKey 指出应用的路径
 lServer 指出应用是否为一个服务器应用

GetApplication方法

检索指定应用的关键字。

语法: GetApplication(cExtension, cExtnKey, cAppKey, lServer)

返回值: @cAppKey

参数:

cExtension 指出应用的扩展名
 cExtnKey 指出应用的注册关键字
 cAppKey 指出应用的路径
 lServer 指出扩展名是否是一个完整的应用

实现方法的过程

```
PROCEDURE filereg.getapppath
  *检查并返回与特殊扩展名(例如XLS, doc)关联的应用的路径
  parameter cExtension, cExtnKey, cAppKey, lServer
  #define ERROR_SUCCESS 0  &&正确, 后面均相同
  local nErrNum, cOptName
  cOptName = ""
  *检查扩展名参数
  If TYPE("cExtension") # "C" OR LEN(cExtension) > 3
    return -103 &&传送的坏参数
  endif
  cExtension = "."+cExtension
  *打开扩展名关键字
  nErrNum = this.OpenKey(cExtension)
  If nErrNum # ERROR_SUCCESS
    return nErrNum
  endif
  *获取文件扩展名关键字值
  nErrNum = this.GetKeyValue(cOptName, @cExtnKey)
  *关闭扩展名关键字
  this.CloseKey()
  If nErrNum # ERROR_SUCCESS
    return nErrNum
  endif
  return this.GetApplication(cExtnKey, @cAppKey, lServer)
ENDPROC

PROCEDURE filereg.getlatestversion
  parameter cClass, cExtnKey, cAppKey, lServer
  local nErrNum, cOptName
  cOptName = ""
  *打开类关键字(例如Excel.Sheet)
  nErrNum = this.OpenKey(cClass+"\CurVer")
  If nErrNum # ERROR_SUCCESS
    return nErrNum
  endif
  *获取文件扩展名关键字值
```

```

nErrNum = this.GetKeyValue(cOptName, @cExtKey)
*关闭扩展名关键字
this.CloseKey()
If nErrNum # ERROR_SUCCESS
    return nErrNum
endif
return this.GetApplication(cExtKey, @cAppKey, lServer)
ENDPROC

PROCEDURE filereg.getapplication
parameter cExtKey, cAppKey, lServer
local nErrNum, cOptName
cOptName = ""
*lServer - 检查OLE服务器
If TYPE("lServer") = "L" AND lServer
    this.cAppPathKey = "\Protocol\StdFileEditing\Server"
else
    this.cAppPathKey = "\Shell\Open\Command"
endif
*打开扩展名应用的关键字
nErrNum = this.OpenKey(cExtKey+this.cAppPathKey)
If nErrNum # ERROR_SUCCESS
    return nErrNum
endif
*获取应用的路径
nErrNum = this.GetKeyValue(cOptName, @cAppKey)
*关闭应用的路径关键字
this.CloseKey()
return nErrNum
ENDPROC

```

应用实例

如下程序获取扩展名为prg的应用的关键字和名称。

```

*文件名:exam3.prg
clear
set classlib to ffc\registry.vcx
oReg = create("filereg")
cExtKey = "prg"
*获取应用的路径
cAppKey = ""
cAppName = ""
nErrNum = oReg.GetAppPath(cExtKey, @cAppKey, @cAppName)
if nErrNum # 0
    MESSAGEBOX("未找到文件!")
    return
endif
?"应用的关键字:"+cAppKey
?"应用的名称:"+cAppName
return

```

程序的执行结果如下:

应用的关键字:Visual.FoxPro.Program

应用的名称:"c:\program files\microsoft visual studio\vfp98\vfp6.exe" -SHELL

4. 数组处理器(Array Handler)

名称	Array Handler
类别	Data Utilities

缺省目录	Visual FoxPro Catalog\Foundation Classes\Utilities
类	_arraylib
基类	Custom
类库	_utility.vcx
父类	_custom
样例目录	...\Samples\Vfp98\Solution\Ffc\arrays.scx
说明	

该类提供了一些方法用于处理不被Visual FoxPro数组函数执行的某些数据操作，包括插入和删除特定的数组元素以及以列方式扫描一个数组。

使用它时，把该类拖动到一个项目上，或者从“组件管理库”的项目快捷菜单中选择“添加到项目”或“添加到表单”。当你把该类拖到一个表单时，Visual FoxPro把该类放在表单上，然后指定适当的属性值和提供必要的输入与输出对象。当你把该类拖到一个项目时，可以选择“添加该类”或“建立一个子类”。

包含对象 无

属性、事件和方法说明

AcolScan方法	扫描数组aSearch的列nColumn，以查找sExpr中指定的值。 语法:AcolScan(@aSearch, sExpr, nColumn, lRetRow) 返回值:无 参数: aSearch 指出要查找的数组 sExpr 指出要查找的表达式 nColumn 指出要查找的表达式sExpr的数组列 lRetRow 指出用该方法找到的sExpr的行号
DelAitem方法	删除一个数组元素，如果该数组是多维数组，则删除一整行。 语法:DelAitem(@aArray, wziRow) 返回值:无 参数: aArray 指出数组的名称 wziRow 指出要删除的行
InsAitem方法	插入一个数组元素。 语法:InsAitem(@aArray, sContents, iRow) 返回值:无 参数: aArray 指出数组的名称 sContents 指出要添加到aArray数组中的一个字符串值 iRow 指出插入sContents值的行

实现方法的过程

```
PROCEDURE _arraylib.acolscan
    parameter aSearch, sExpr, nColumn, lRetRow
    local npos
```

```
If VARTYPE(nColumn) #'N'
    nColumn = 1
endif
If VARTYPE(lRetRow) #'L'
    lRetRow = .F.
endif
npos = 1
do while .T.
    npos = ASCAN(aSearch, sExpr, npos)
    do case
        case npos=0    &&未找到相匹配的
            exit
        case ASUBSCRIPT(aSearch, npos, 2)=nColumn
            exit
        otherwise
            npos=npos+1
    endcase
enddo
If lRetRow
    return ASUBSCRIPT(aSearch, npos, 1)
else
    return npos
endif
ENDPROC

PROCEDURE _arraylib.delaitem
    parameter aArray, wziRow
    local nColumns
    If ALEN(aArray, 1)<wziRow OR wziRow<0
        return
    endif
    If ALEN(aArray, 1)=1
        store "" TO aArray
    else
        nColumns = ALEN(aArray, 2)
        ADEL(aArray, wziRow)
        If nColumns=0
            dimension aArray[ALEN(aArray)-1]
        else
            dimension aArray[ALEN(aArray, 1)-1, nColumns]
        endif
    endif
ENDPROC

PROCEDURE _arraylib.insaitem
    *lSetAll对于多维数组是否设置行中所有元素
    parameter aArray, sContents, iRow, lSetAll
    local nColumns
    If VARTYPE(iRow) # "N"
        iRow = ALEN(aArray)
    endif
    nColumns = ALEN(aArray, 2)
    do case
        case iRow<0
            return
        case ALEN(aArray)=1 AND EMPTY(aArray[1])
            aArray[1]=sContents
        case nColumns = 0
```

```

        iRow = MIN(iRow, ALEN(aArray))
        dimension aArray[ALEN(aArray)+1]
        If iRow # ALEN(aArray)-1
            AINS(aArray, iRow+1)
        endIf
        aArray[iRow+1]=sContents
    otherwise
        iRow = MIN(iRow, ALEN(aArray, 1))
        dimension aArray[ALEN(aArray, 1)+1, nColumns]
        If iRow # ALEN(aArray, 1)-1
            AINS(aArray, iRow+1)
        endIf
        If !lSetAll
            aArray[iRow+1, 1]=sContents
        else
            for i = 1 TO nColumns
                aArray[iRow+1, i]=sContents
            endfor
        endIf
    endcase
ENDPROC

```

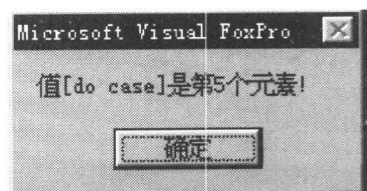
应用实例

该例定义一个两维数组myarray，建立一个不可视对象myfr，在数组中查找指定的元素。

```

*文件名:exam4.prg
dimension myarray[2, 3]
myarray[1, 1]="do while"
myarray[1, 2]="for"
myarray[1, 3]="scan"
myarray[2, 1]="if"
myarray[2, 2]="do case"
myarray[2, 3]="endif"
set classlib to ffc\_utility.vcx
myfr=createobject("_arraylib")
cSaveExact = SET("EXACT")
set exact off
fdata="do case"
incol=1
do while (incol<=3)
    inno= myfr.ACOLSCAN(@myarray, fdata, incol)
    if inno>0
        exit
    endif
    incol=incol+1
enddo
if (inno>0)
    messagebox("值["+fdata+"]是第"+str(inno, 1)+"个元素!")
else
    messagebox("未找到["+fdata+"]元素!")
endif
set exact &cSaveExact
return

```



程序执行结果如图4-1所示，单击“确定”后退出。

图4-1 程序执行屏幕