

# Visual Foxpro 5.0 与其他软件间的资源共享

赵 辉 (浙江财经学院信息系 杭州 310011)

**摘要:**本文介绍了 Visual Foxpro 提供的 OLE 技术及其应用方法,并引用了多个实例详细介绍 VFP 与 DELPHI, EXCEL, WORD, VB, IE 等应用程序间的相互联系。

**关键词:**VFP DELPHI EXCEL WORD VB OLE ActiveX

## 一、前言

Visual Foxpro 因其易学易用而深受广大用户的欢迎,但与 VB, DELPHI 等热门软件相比,其所提供的控件数量和功能都稍逊一筹,但 VFP 提供了对象链接与嵌入 (OLE) 和 ActiveX 控件等技术,可充分利用其他基于 Windows 的应用软件 (如 VB, DELPHI, EXCEL, WORD, ActiveX 控件等) 所提供的强大功能来拓展自身的功能。VFP 用户在开发应用程序时,如果想充分融和各软件的优秀功能,应了解和掌握 VFP 与其他应用程序间的联系方法。本文探讨如何实现 VFP 与其他应用软件间的资源共享。

## 二、VFP 的 OLE 技术

VFP 提供的 OLE 技术,可将其他应用程序或数据看作一个对象,放置在表单上或表中的通用字段中。VFP 可以嵌入方式或链接方式引用其他应用程序或数据。嵌入方式和链接方式的不同在于数据的存储地点。嵌入将数据直接存储到表或表单中,而链接则仅保留了对象的引用途径。VFP 提供了两种形式建立 OLE: 绑定型和非绑定型。

可以用“表单控件”工具栏中的 OLE 绑定型控件创建绑定型 OLE 对象,而用 OLE 容器控件则创建非绑定型 OLE 对象。绑定型 OLE 对象一般与表中的通用字段相联系,如在表的通用字段中添加一个 Word 文档,一个声音,图像或视频文件等。非绑定型 OLE 对象则与表单相联系,如在表单的 OLE 容器中放入 ActiveX 控件,甚至一个基于 Windows 的 .EXE 程序。向表单添加 ActiveX 控件可有两种方式。一种方式是在设计表单时直接设置,具体步骤为:

1. 从“表单控件”工具栏中选择“OLE”,并将其放置在表单上。

2. 在“编辑”栏选择“Insert Object”,在对话框中选择“Insert Control”。

3. 在“Control Type”列表中选择所要的 ActiveX 控件,与 VFP 一起发布的 ActiveX 控件包括 Windows95 控件和系统控件。另一种方式则使用自动化管理对象,用程序代码来动态建立 OLE 对象,具体实例见后。

## 三、VFP 调用 DELPHI 应用程序

Delphi 是当今最优秀的 Windows 应用程序开发工具,拥有面向对象的可视化设计,丰富的控件和众多的功能,日益受到软件开发人员的青睐。VFP 用户可在自己的应用程序中直接调用 Delphi 开发的应用程序或 DLL 例程。

### 1. VFP 调用 DELPHI 开发的应用程序 (.exe 文件)

VFP 可以在表单中通过 OLE 容器控件调用 Delphi 开发的应用程序。因 VFP 不支持带空格的路径,为了在 VFP 中能正常地调用其他的应用程序,最好事先将相关的应用程序文件移至 VFP 目录。下面的例一采用代码方式调用 Delphi 开发的应用程序 Project1.exe。

例一:

Pr01.prg

\* VFP 调用 DELPHI 应用程序 project1.exe

\* 首先创建一个表单,在其上添加 OLE 控件,并链接应用程序 Project1

Set talk off

Clear

frmMyForm = CREATEOBJECT('form') && 创建一个表单

frmMyForm.Closable = .F.

frmMyForm.AddObject('CmdCommand1', 'CmdMy-CmdBtn') && 添加命令按钮

```
frmMyForm.AddObject('OleObject', 'Myclass') && 添加 OLE 对象
frmMyForm.CmdCommand1.Visible = .T. && 显示 QUIT 命令按钮
frmMyForm.OleObject.Visible = .T. && 显示 OLE 控件
frmMyForm.OleObject.Height = 150 && 设置 OLE 控件尺寸
frmMyForm.OleObject.Width = 200
frmMyForm.Show && 显示表单
frmMyForm.OleObject.DoVerb(0) && 执行 OLE 对象
Read Events && 开始事件处理
Define Class MyClass as OleControl && 定义 OLE 控件类型
DocumentFile = 'd:\vfp\project1.exe' && 指定 OLE 执行文件
Enddefine
Define Class cmdMyCmdBtn as CommandButton && 定义命令按钮类型
Caption = '<QUIT' && 命令按钮上的标题
Cancel = .T. && 默认 QUIT 命令按钮
Left = 125 && 命令按钮的尺寸
Top = 210
Height = 25
Procedure Click && 停止事件处理, 清除表单
Clear Events
Enddefine
Return
```

利用此段程序代码甚至可以在 VFP 中调用非 Windows 的应用程序。

## 2. VFP 调用 DELPHI 编写的 DLL 中的例程

在 VFP 应用程序中可以直接引用 DELPHI 编写的 DLL 中的例程。DLL(Dynamic - Link Library)动态链接库是一个以扩展名为 .DLL 的可执行模块, 其中包含了可以被其他应用程序或其他 DLL 共享的例程和资源。DLL 的主要特点是它的代码是在运行期间动态地链接到调用它的程序中。当 DLL 中的例程执行完后, 可以随时将 DLL 从内存中清除。DLL 技术可以实现各应用程序间的资源共享, 并节省内存空间。在 Windows 及其他软件中被广泛应用。在用 DELPHI 编写 DLL 时, DLL 中供其他应用程序调用的例程(函数或过程)要用 Exports 申明引出, 要引出的例程最好用 Stdcall 调用约定, 以便用其他语言编写的应用程序也能调用本 DLL 中的例程。

DLL 中的例程可以直接在工程文件中进行定义及申明引出, 也可在 unit 单元中进行定义并直接用 Exports 申明引出。例二为一个最简单的 DLL 实例, 其中包含例程 Max。

例二:

```
Library MyDLL
uses Forms;
Function Max(x, y: Integer): Integer; Stdcall; {定义例程 Max}
begin
  if x > y then Max := x
  else Max := y;
end;
{申明引出例程 Max}
Exports Max;

| $R * .RES|

begin
Application.Initialize;
Application.CreateForm(TForm1, Form1);
Application.Run;
end.
```

在 Visual Foxpro 程序中, 引用外部的 DLL 及其中的例程可使用 DECLARE - - DLL 命令, 命令的基本格式为:

```
DECLARE [ cFunctionType ] FunctionName IN LibraryName[ As aliasName ]
[ cParamType1 [ @ ] ParamName1, cParamType2 [ @ ] ParamName2, ... ]
```

LibraryName: 为引用的动态链接库名。

CFunctionType: 要引入的 DLL 中的例程的类型。如果例程有返回值, 则包括该项, 如果例程没有返回值, 则不用该项。

FunctionName: 调用的例程名。

cParamType1 [ @ ] ParamName1: 例程中形参的类型及名称。

cParamType2 [ @ ] ParamName2: 例程中形参的类型及名称。

为了能正常地调用自己编写的 DLL, 最好事先将要引用的 DLL 复制到 VFP 子目录以保证 Visual Foxpro 能正确地搜索 DLL。

可使用 CLEAR ALL DLLS 命令, 卸载由 DECLARE - DLL 命令引入的外部 32bitDLL。

程序实例见例三。

例三:

```
set talk off
clea
* 用 declare 命令加载 DLL 及其中的例程 Add
declare integer Max in MyDLL.DLL integer @x, integer @
y
a = 8
b = 5
wait window"正在调用例程 Max" timeout 1
c = Max(a, b)
? c
wait window"释放加载的 DLL" timeout 2
clear all dlls
retu
```

#### 四、VFP 调用 Internet Explorer 浏览器

如果希望在 Visual Foxpro 开发的应用程序中, 直接启动 Internet Explorer 浏览器, 并对指定的网址进行浏览, 最简单的方法是使用 RUN 命令。通过 RUN 命令可启动指定路径上的浏览器软件 IExplore.exe 并可同时将要浏览的网址传递给 IExplore.exe。具体代码如下:

```
address = "http://www.chinabyte.com "
run/n3 c:\internet\iexplore.exe &address
```

其中 /n3 表示运行 Windows 程序, 并以最大窗口显示。&address 为传递给 IExplore 所需的链接网址。当然也可用前述调用 DLL 的方式, 通过 Win32API 中的 ShellExecute() 函数间接调用 IExplore.exe。

#### 五、其他应用程序控制 VFP

##### 1. Word, Excel 直接控制 VFP

Visual Foxpor 可以作为一个客户, 也可作为一个服务器, 其他支持自动化的应用程序可通过 VFP 的 Application 对象控制 Visual Foxpro。当 VFP 启动时, 通过 DDE 或通过自动化自动地创建一个 Application 对象。比如在 Microsoft Excel 或在 Microsoft Word 中访问 VFP 中的 Table。如下例四中代码为 Microsoft Word 模块中用 Visual Basic Application 宏代码创建 Visual Foxpro Application 对象, 打开 VFP 中的表 Table1, 并将其中的记录

内容粘贴到 Word 文档的实例。

例四:

```
Sub Micro1()
Dim oFox as Object
Set oFox = CreateObject("VisualFoxpro.Application")
oFox.DoCmd "use d:\vfp\Table1.dbf"
oFox.DoCmd "Select name, address From Table1 Into temp"
oFox.DataToClip "temp", , 3
Word.ActiveDocument.Content.Paste
End Sub
```

若要在 Excel 中读取 VFP 表中数据, 可对最后一条代码作修改:

```
ActiveSheet.Paste。
```

##### 2. 创建 OLE 服务程序

可以用 Visual Foxpro 创建 OLE 服务程序, 供其他应用程序调用。要实现 OLE 服务程序, 应在项目中包含 OLEPUBLIC 的类, 并编译成一个可执行文件(.EXE 文件)或一个动态链接库(.DLL 文件)。下面例五中的代码创建一个自定义的 OLEPUBLIC 类。

例五:

```
Define Class Person As Custom OLEPUBLIC
    FirstName = Space(20)
    LastName = Space(20)
    Procedure GetName
    Return This.FirstName + " " + This.LastName
    Endproc
Enddefine
```

可以在 Excel, Word 或 VB 中用如下代码调用该服务程序:

```
Set oTest = CreateObject("proj1.person")
cName$ = oTest.GetName()
```

在标准的自动化方案中, 客户和服务程序是在同一台计算机上运行, 在 VFP5.0 下, 也支持远程自动化。通过网络, 客户程序还可以调用远程的服务程序。

#### 参考文献

- [1] Delphi 3 编程指南 宇航出版社
- [2] Visual Foxpro5.0 完全学习手册 科学出版社
- [3] Visual Foxpro 5.0 高级程序设计指南 海洋出版社

(来稿时间: 1998 年 12 月)