

VB与动态连接库

朱志良 陈东明 (东北大学计算中心 110006)

摘要:本文介绍了利用 Visual Basic 5.0 设计、引用 ActiveX DLL 的步骤及方法,并提出几种不同的动态连接库的注册方案。

关键词:Visual Basic ActiveX DLL ActiveX EXE 动态连接库 注册

一、问题的提出

如果开发一个大的系统,仅建立一个单一的可执行文件(.exe),那么编译出来的可执行文件太大,将影响执行的效率并会耗费大量的内存。另外,在源代码(Source Code)愈来愈多的情况下,当一个工程(Project)中的许多模块合并时,难免发生声明及相关问题的冲突,造成合并及管理上的困难,从而导致产品生产周期的延长。

笔者结合开发经验,提出将各个相对独立的模块作成一个个独立 DLL 档,然后再合并各个模块的方法。

但就通常 VB 的程序设计来说,并没有使用到“程序库(Libraries)”的程序设计概念。VB 虽已经成为一种真正专业化的开发语言和编程环境,但并没有办法设计出一个能够直接处理电脑硬件界面的程序,也没有办法去处理极为大量、动态的内存控制,当然更是无法直接以组合语言直接命令微处理器(CPU)、浮点运算单元(FPU)进行特定的运算。所以无法设计出如同 C, C++, PASCAL, ASM 等语言所设计出来的动态连接库。此外,对于 Windows 系统的程序控制,VB 请求的是一个使用者界面,并无法处理底层的每一个细节动作或原始资料。

但这并不表示 VB 无法设计动态连接库,透过 D-COM(Distributed Component Object Model)的标准,VB 还是可以设计出动态连接库。但是,这些由 VB 所设计的动态连接库,在使用上与其他语言所设计的动态连接库略有不同。在程序中,它们是以“控件”(ActiveX DLL 及 ActiveX EXE)的形式被使用的。

由于动态连接库是一个独立于应用程序的部分,设计程序时,其实只是建立一个“应用程序与动态连接库之间的界面”,因此在使用动态连接库之前,必定得告诉应用程序,它要与哪个动态连接库进行连接,并且将会使用到其中的哪些函数与子程序,这时就需要进行合适的声明。

二、DLL 档的生成

利用 Visual Basic 5.0 可以生成 ActiveX EXE 和 Ac-

tiveX DLL 程序,这些 ActiveX 构件是支持 OLE 的自动化服务器程序,它就像操作系统的扩展部分,可提供给能访问这些构件的任何应用程序。ActiveX EXE 建立一个独立的构件,即过程外服务器程序。这个服务器在独立过程中运行,是单独执行的线程。由于服务器在自己的进程中运行,客户机可以让它进行处理,然后在服务器工作的同时继续自己的工作。如果生成 ActiveX DLL,则生成过程内的构件。DLL 装入内存后,其他过程可以使用它。构件不必象 ActiveX EXE 构件中那样重新装入。下面就是生成 ActiveX DLL 项目的示例:

1. 开启新的 Visual Basic 5.0 工程,并选取 ActiveX DLL,“确定”后 VB 生成一个工程(Project1)并加入一个新的类模块(Class1)。

2. 将类模块的 Instancing 属性改为 5 - MultiUse,这样就可以将此 DLL 档开放给多个程序分享使用。再在此项目中加入一个模块,此模块的作用是用以放置开放给其他程序使用的相关函数及声明。

3. 在类模块中加入所要执行的功能,其中的每个函数 Sub 或 Function 之前加上关键词 Public,以将此函数开放给其他程序使用。因为在类模块中无法使用 Public 声明常数(如 PI)或是自定义的声明(如 Type...End Type),因此必须将此类的声明放在加入的模块中,以方便管理。

4. 在类模块中放置不开放给外部程序使用的函数,此函数仅供 DLL 内部使用时,在 Sub 或 Function 前加上 Private 保留字,以免外部程序误用,造成不必要的程序执行错误。

5. 将工程存档,并生成 DLL 文件。

三、DLL 的测试与调用

由于 DLL 无法单独执行,必须借由可执行档(.exe 或 .dll)调用连接才能执行。因此,若要测试已生成的 DLL 档,必须先建立一个 EXE 档。下面是一个较为方便的测试方法:

在生成了 DLL 文件后,就可以在其他应用程序中调

用 DLL 中的开放函数,可按以下操作进行:

1. 建立一个新的 Visual Basic 工程,并选取标准 EXE 程序(Standard EXE),以建立新的可执行文件。

2. 将此文件保存,放在某一测试目录下,如 DLL-TEST。

3. 将刚才生成 DLL 的三个文件(.cls, .bas, .dll)复制到 DLLTEST 目录中。

4. 在这个新的工程中,加入刚复制的两个文件(.cls 和 .bas)。

(当您在加入类模块(.cls)时,将会出现一个对话框,确定即可。当测试此类模块没有问题时,欲将它放回原 DLL 的工程中时,请将其 Instancing 属性还原为 5 - MultiUse)。

5. 引用 DLL,选取菜单 Project, References..., 在出现的对话框中选取 Browse... 按钮,选择前面制作好的 DLL,并按下确定按钮即可。

6. 在 Form1 窗体上加一个命令按钮(Command button)。

7. 在 Private Sub Command1-Click() 中加入下面的测试代码:

(假设 DLL 中的类模块名称为 ClassTest,其中有被声明为 Public 的函数 FunTest())

```
Private Sub Command1-Click()  
    Dim obj As Object  
    Set obj = New ClassTest  
    Call obj.FunTest() '若有参数,则对应上  
    Set obj = Nothing  
End Sub
```

在对象 obj 使用完毕后,必须将其设定为 Nothing,这样才能将其占用的内存空间释放。

这样,既可以执行 DLL 文件,又可进入 Debug Mode 进行除错。

8. 执行此程序并按下 Command1 按钮,即可进行测试、排错。

9. 测试完毕,确认 DLL 无错后,便可以由其他程序引用了。

在新的可执行文件中,选取 Project, Reference..., 选择所需的 DLL,然后就可以象第 7 步中一样调用 DLL 中的函数。

四、DLL 注册

以上的操作是在一个系统上执行的,由此生成的可执行文件(其中调用了 ActiveX DLL)拿到其他系统上执行时,就会发生“ActiveX component Can't Create Object”

的错误,这是因为该 DLL(它以进程内服务器的形式运作)没有在系统上注册所致。在 Microsoft Windows 中,应用程序使用初始化文件来存储信息,否则信息在应用程序关闭时会丢失。这些文件通常包括用户偏爱的应用程序配置之类的信息。但现在我们知道,初始化文件有许多固有的局限性。

Microsoft Windows 3.1 版已包括一个新特点:注册数据库。注册数据库包含支持外壳(shell)应用程序和使用对象链接与嵌入的应用程序的信息。数据库中的每条信息都由一个链标识,它可以和一个值(数据)相连。

Win32 应用程序编程接口(API)扩展了 Windows 3.1 注册数据库的使用。数据库现在使用更多的键而且每个键具有更多数据。新的数据库称为注册。

因此,一个系统上的文件到另外系统上运行时可能会出现问題。由 Visual Basic 5.0 生成的 ActiveX DLL 就是这样。为此,可采取下面几种简单易行的方法:

1. 如需在一台客户机上注册 DLL,使其以本地方式运行,可用 Regsvr32.exe 应用程序:选“开始”、“运行”,在命令行输入“Regsvr32 ... [.dll]”,“确定”即可(完成 ActiveX 的本地注册)。

2. 可用“设置向导”(Setup Wizard)创建一个设置程序,这样不仅能安装组件,还能在注册表里对其进行注册,并可向用户提供组件的反安装或撤除途径。

3. 用 Setup ToolKit Dll 中的 DllSelfRegister API 函数:

```
Declare Sub DllSelfRegister Lib "STKIT432.DLL"  
(ByVal lpDllName As String)  
Sub Register()  
    DllSelfRegister("... [.dll]")  
End Sub
```

使用此方法,您可以制作一个实用的注册小工具。

4. 如果您的系统上安装了 Visual Basic,那么您也可以将源代码编译成 EXE 或 DLL,即可自动注册。

(以上“... [.dll]”表示动态连接库所在的完整路径)

5. 与 VB 配套的“客户机注册工具”允许通过命令行注册组件(clireg32.exe)。为使用它,确保对组件进行编译的时候,已复选了“远程服务器支持文件”选项(Remote Server Support Files)。

五、结束语

本文是笔者在从事 VB 开发 Windows 应用程序工作中所积累的经验方法,并在许多场合得到应用,实践证明是可行的。

(来稿时间:1998 年 9 月)