



ActiveX 控件注册的 几种方法

肖毅 朱绍文 熊伟 陈绪君 张大斌 (武汉华中师范大学电子与计算机研究所 430079)

摘要: 本文介绍了 ActiveX 控件注册的几种常用方法, 详细阐述了各自的特点及应用范围, 并给出了实例。

关键词: ActiveX 控件 注册表 COM

1 前言

要适应企业环境不断变化的需求, 成功地开发出高质量的应用系统, 必须采用先进的开发技术, 这对于减轻应用开发人员的开发负担, 提高开发速度和质量都有十分重要的意义。为此, 开发人员纷纷运用控件技术进行编程, ActiveX 也已成为许多开发者竞相追寻的目标。

一个 ActiveX 控件就是一个功能相对独立、且具有一定构成和调用规则的程序功能模块, 它是作为基于 COM (Component Object Model, 组件对象模型) 服务器进行操作的, 并且可以嵌入在包容器宿主应用程序中。由于 ActiveX 控件与开发平台无关, 因此, 在一种编程语言上开发的 ActiveX 控件无须作任何修改, 即可在另一种编程语言中使用, 其效果如同使用 Windows 通用控件一样。例如, 在 Visual C++ 中开发的 ActiveX 控件, 不作任何修改即可应用于 Visual Basic 环境中。由此可见, 通过使用 ActiveX 控件即可实现快速小型的组件重用、代码共享, 从而提高编程效率, 降低开发成本。

现在 Internet 上提供了许多免费的 ActiveX 控件, 从 Internet 上下载的 ActiveX 控件, 或从其他途径得到的 ActiveX 控件, 用户并不能直接使用。一般来说, 一个外来 ActiveX 控件要在 Windows 中被正确使用, 首先必须将控件文件 (*.OCX) 复制到硬盘中, 然后将其在 Windows 注册表中注册。Windows 注册表中包含每一个组件的信息以及使用该组件所需文件的位置。

2 注册的作用

Windows 注册表包含有几乎所有关于系统的信息。注册表根据当前配置、当前用户、用户、机器等信息分为几个不同的根键。为了使 ActiveX 组件运行, 这里我们关心的是注册表中的 HKEY_CLASSES_ROOT 字段。组件对象模型从两种方式使用注册表来解决 ActiveX 组件引用的

问题, 即按名字和按类型 (CLSID)。一个组件的名字是在对话框中设置的。当一个单个的 DLL 文件包含多个对象时, 组件的名字以 <Application>.<Object> 的形式出现。CLSID 是 Windows 作为一个全局唯一标识符 (GUID, globally unique identifier) 产生的标识符。GUID 是由 Windows 以一个格式化的十六进制字符串形式显示的一个十六字节长的唯一键。

Windows 把一个组件的名字作为它在注册表中的入口键并据此来查找相应的 CLSID, 然后利用 CLSID 来查找组件运行所需的各种信息。这些信息包括:

- (1) InProcServer32 文件装入运行组件
- (2) 在项目属性页中作为项目名 (设置的 ProgID 与注册名相同);
- (3) TypeLib 识别组件定义的接口 (以便相容的客户程序不能用它);
- (4) 版本号识别安装的组件版本, 通过 RegEdit32.EXE 程序, 用户可以手工检查和修改注册表信息。

注册 ActiveX 控件一般来说有三种途径, 适用范围不同, 各有各的优点, 下面分别讨论。

3 使用 Regsvr32.exe 程序注册 ActiveX 控件

对于未注册过的 ActiveX 控件可使用 Regsvr32.exe 程序对其进行注册, 外来 ActiveX 控件要应用到自己的程序中也必须进行注册。该文件位于 Windows 目录的 system 子目录下。使用方法如下: 点击“开始”→“运行”, 在运行对话框中输入以下命令:

- ```
regsvr32 <文件名> 注册一个 ActiveX 控件
regsvr32 /u <文件名> 解除某 ActiveX 控件的注册
```
- 我们在使用一些带有 ActiveX 控件的应用程序时, 有时会出现不能运行的情况, 这可能是其自带的 ActiveX 控

件未注册所至，这时不妨使用上述命令一试。另外，Visual C++ 使用者若对 regsvr32.exe 的编码感兴趣的话，可在 Visual C++ 的联机帮助中找到其源代码。若使用 Visual C++ 6.0，则在 MSDN 光盘的 \sample \VC98 \MFC \controls \regsvr 中。

除了 regsvr32 应用程序之外，Microsoft 在它的 Web 站点上还提供了一个称为 regclean.exe 的程序。regclean.exe 可以检查注册表并去掉丢失的组件在注册表中的条目。例如，假设你手工清除了 Excel 的所有文件而没有运行它的 Uninstall 程序，所有你注册的 DLLs 都将被清除，但它们的注册条目仍然在注册表中并指向不存在的文件。Excel 的 ActiveX 组件将无法正常运行，regclean.exe 可以删除这些“悬挂”条目，它还能提供一个“undo”文件帮助你恢复错误的删除。

#### 4 使用安装程序制作软件：InstallShield 注册 ActiveX 控件

用 regsvr32.exe 来注册 ActiveX 控件虽然简单，但需要用户手工进行注册，在不用时还得手工解除注册，因此，这对一个要发布的应用程序来说并非是最好的解决方案。大型应用软件一般都有一个安装程序，在安装程序中解决 ActiveX 控件的注册是较为理想的一种方案。

InstallShield 是目前较为流行的制作安装程序的工具，InstallShield 已经推出多种针对不同开发工具的专用版本，如 InstallShield for Delphi、InstallShield for Visual C++ 等等，使用 InstallShield 可以制作出专业级的 Setup 安装程序，还可以注册应用程序要使用的 ActiveX 控件；而且，在以后使用控制面板的安装 / 删除应用软件卸载该软件时，安装程序会自动注销掉以前注册的 ActiveX 控件。

下面以 InstallShield for Delphi 为例说明如何实现 ActiveX 控件的自动注册，其他版本的使用方法类似。

(1) 启动 Installshield，使用 Project Wizard 建立一个新的项目；

(2) 新建一个“File Group”，将需要注册的 ActiveX 控件文件放入此“File Group”中；

(3) 将上述“File Group”的“Self - Registered”属性设置成“Yes”。

上述步骤仅为制作 Setup 安装程序中设置自动注册 ActiveX 控件的操作，至于如何使用 InstallShield 来制作一个完整的 Setup 安装程序不在本文讨论范围之内，请读者自己参看有关书籍。

#### 5 在应用程序中加入注册代码

使用 Installshield 软件虽然可以实现 ActiveX 控件的

自动注册，但必须制作一个安装程序，对于一个小型应用程序或不需分发的程序来说未免太麻烦。这时可以在程序中嵌入注册代码，实现应用程序自注册。其编程方法是：

(1) 使用 Windows API 函数 LoadLibrary 载入 ActiveX 控件；

(2) 使用 GetProcAddress 函数获取 ActiveX 控件中注册函数 DllRegisterServer(注销函数为 DllUnregisterServer) 指针。

下面以 Visual C++6.0 编写的 RegActivex 程序为例介绍怎样在程序中启动注册及注销 ActiveX 控件。RegActivex 的主体框架由 Visual C++6.0 的 AppWizard 自动产生，在其基础上增加两个菜单项：“注册 ActiveX 控件”、“注销 ActiveX 控件”。当选择“注册 ActiveX 控件”菜单时，将对控件 filter.ocx（一个图像滤镜控件，支持图像的柔化、锐化等）进行注册，当选择“注销 ActiveX 控件”菜单时，将解除 filter.ocx 控件的注册。

步骤一：使用 Visual C++6.0 建立一个单文档的应用程序 RegActivex，其他项目接受默认设置；

步骤二：给应用程序增加两个菜单项：“注册 ActiveX 控件”、“注销 ActiveX 控件”，步骤如下：

① 单击 Visual C++6.0 左边窗口的“ResourceView”，双击 MENU 项目下的 IDR\_MAINFRAME 以打开菜单编辑器；

② 在菜单编辑器中，加入一个主菜单“注册”，在注册“主菜单下加入两个菜单项：“注册 ActiveX 控件”、“注销 ActiveX 控件”。

步骤三：给新建的菜单项“注册 ActiveX 控件”、“注销 ActiveX 控件”增加响应函数；

① 在资源编辑器中，双击 MENU 中的 IDR\_MAINFRAME 打开菜单编辑器，单击“注册 ActiveX 控件”，按 Ctrl+W 打开 MFC Class Wizard；

② 双击 Messages 框中的 COMMAND，给“注册 ActiveX 控件”菜单项增加响应函数 OnRegisterReg()，单击右边“Edit”按钮，给函数 OnRegisterReg() 增加如下代码：

```

Void CmainFrame::OnRegisterReg ()
{
 //ActiveX 控件的路径及文件名
 LPCTSTR pszDllName = "filter.ocx";
 // 装载 ActiveX 控件
 HINSTANCE hLib = LoadLibrary (pszDllName);
 if (hLib < (HINSTANCE)HINSTANCE_ERROR)
 {
 MessageBox ("不能载入Dll文件!", "结果", MB_OK);
 Return;
 }
}

```

```

}
//获取注册函数 DllRegisterServer 地址
FARPROC lpDllEntryPoint;
 lpDllEntryPoint = GetProcAddress (hLib, _T
("DllRegisterServer));
//调用注册函数 DllRegisterServer
if (lpDllEntryPoint!= NULL)
{
if (FAILED ((*lpDllEntryPoint)()))
{
 MessageBox ("调用 DllRegisterServer 失败!", "结果
", MB_OK);
 FreeLibrary (hLib);
 Return;
};
MessageBox ("注册成功", "结果", MB_OK);
else
 MessageBox ("调用 DllRegisterServer 失败!", "结果",
MB_OK);
}

```

③ “注销 ActiveX 控件” 菜单项响应函数的编写方法同上, 代码也相似, 只是将 “lpDllEntryPoint = GetProcAddress (hLib, \_T (“DllRegisterServer”));” 改为:

```
lpDllEntryPoint = GetProcAddress (hLib, _T
("DllUnRegisterServer")); 即可。
```

步骤四: 编译连接上述程序, 即可得到一个可执行文件。另外需要注意的是应将 ActiveX 控件文件 filter.ocx 复制到 Windows 的 system 目录下。

步骤五: 测试上述程序。测试程序是一个引用了 filter.ocx 控件的宿主程序, 由于本文的重点在于介绍 ActiveX 控件的注册方法而不在与宿主程序的设计, 因此宿主程序引用《程序员大本营》中的 filter 程序, 读者也可以自己设计测试程序。测试的方法是运行 filter 程序, 若使用 RegActivex 注册了 filter.ocx 控件, 则该程序可以正常运行, 若使用 RegActivex 注销了 filter.ocx 控件则 filter 不能运行。

## 6 结束语

上面所述的 ActiveX 控件注册方法各有特点, 并且在笔者为北京市地税局开发的税务稽查系统中得到了应用, 经实际使用, 效果良好。读者可根据实际情况灵活选用。■

### 参考文献

- 1 郑若忠、宁供、数据库原理, 国防科技大学出版社, 1998。
- 2 张龙祥、黄正瑞、陶影, 数据库原理与设计, 西南交通大学出版社, 1995。
- 3 Jeny Anderson. Visual C++5 ActiveX 编程指南, 清华大学出版社, 1998。
- 4 B.Heyman,R.Groom. Visual Basic 6.0 数据库开发.清华大学、版社, 1999。©《计算机系统应用》编辑部 <http://www.c-s-a.org.cn>