

# 试谈 COM 智能注册系统的实现

## The Discussion of How to Implement COM Smart Register System

杨宗长 徐继生 (武汉大学电信学院 430072)

摘要: COM文章就如何在Windows系统中,实现COM智能注册系统进行了设想和探讨;并由此给出了实现的步骤和必要的VC++6.0下的程序代码。

关键词: Windows COM 智能注册 VC++6.0

基于COM的应用程序在Windows平台已经崭露头角,而使用COM组件首先必须注册组件;因此能实现COM组件的智能注册将是一件非常有实际意义的工作。实际上,在Windows中,比如:只要将符合“控制面板”规范CPL文件安装在C:\windows\system或c:\winnt\system32目录中,系统将自动把它装载到“控制面板”中去。鉴于此思路,本文将试图探讨如何实现COM组件的智能注册的系统。

### 1 COM 智能注册系统的探讨

#### 1.1 COM 注册系统的默认目录的建议

理论上,我们将COM组件安装在Windows中的任意目录中,这样做,一方面管理上容易混乱;另一方面,对于智能注册系统来说将要检测更多的文件目录,花费更多的系统资源。因此,我们有必要参照Windows系统的一些已知的规范,提出以下建议:

- (1) 智能注册系统的默认主目录为: C:\COMDIR (新建);
- (2) 添加欲注册COM组件的子目录为: C:\COMDIR\ADDNEWDIR
- (3) 已成功注册的COM组件的子目录为: C:\COMDIR\REGDIR
- (4) 注册失败查看的子目录为: C:

\COMDIR\ERRORDIR

#### 1.2 实时监控目录的实现

对于智能注册系统来说,需要对于ADDNEWDIR目录进行实时的监控;一旦检测到目录属性变化事件发生(文件名、大小和修改变等),便可进行注册事宜。而欲对某个目录进行实时监控。在Windows操作系统中,Microsoft提供相关的目录监控API函数:

##### 1.2.1 启动实时监控函数: FindFirstChangeNotification()

```
FindFirstChangeNotification(
    LPCSTR lpPathName, //要监控
    的目录
    BOOL bWatchSubtree, //是否
    监控其子目录
    DWORD dwNotifyFilter
) // dwNotifyFilter该参数可以为以下:
// FILE_NOTIFY_CHANGE_FILE_NAME,
监控目录或子树下文件名的变化(重命名,
创建,删除等);
//FILE_NOTIFY_CHANGE_DIR_NAME,
监控目录或子树下目录名的变化(重命名,
创建,删除等);
//FILE_NOTIFY_CHANGE_ATTRIBUTES,
监控目录或子树下目录属性的变化;
//FILE_NOTIFY_CHANGE_SIZE,监控目
```

录或子树下文件大小变化;  
 //FILE\_NOTIFY\_CHANGE\_LAST\_WRITE,  
 监控目录或子树下文件最近修改写入;  
 //FILE\_NOTIFY\_CHANGE\_SECURITY,监  
 控目录安全属性变化;  
 成功将返回监控句柄,一般使用一个线  
 程来等待其监控变化。否则句柄值将为  
 INVALID\_HANDLE\_VALUE

##### 1.2.2 继续监控

```
FindNextChangeNotification (
    HANDLE hChangeHandle/
/监控句柄
);
```

##### 1.2.3 关闭监控

```
FindCloseChangeNotification(
    HANDLE
hChangeHandle//监控句柄
);
```

##### 1.2.4 程序实现

```
#define COM_ADDNEWDIR "C:
\\COMDIR\\ADDNEW\\"
#define COM_REGDIR "C:
\\COMDIR\\REGDIR\\"
#define COM_ERRORDIR "C:
\\COMDIR\\ERROEDIR\\"
HANDLE hFindChange;
HANDLE hThread;
(1) 实时监控线程
UINT SmartRegThread(LPVOID lp)
{ BOOL bf;
do{
::WaitForSingleObject
(hFindChange,-1);//等待目录监控事件的发生
bf=::FindNextChangeNotification
(hFindChange);
if(bf)
{。。。//处理部分,如放送消息
给系统等}
}while(1);
return 0;
}
(2) 启动实时监控
```

启动实时监控, 该系统需要监控的属性如下

```
hFindChange = ::
FindFirstChangeNotification(
    COM_ADDNEWDIR,
    0,
    FILE_NOTIFY_CHANGE_FILE_NAME|
    FILE_NOTIFY_CHANGE_SIZE|
    FILE_NOTIFY_CHANGE_LAST_WRITE);
if (
hFindChange==INVALID_HANDLE_VALUE)
{
    AfxMessageBox("启动失败");return;
}
```

```
hThread = ::AfxBeginThread(
SmartRegThread,(LPVOID)this>m_hWnd);
```

### 1.3 文件的搜索的实现

一旦注册系统检测到注册事件发生, 注册系统将搜索ADDNEWDIR目录: 首先将文件移至REGDIR目录; 然后执行注册, 若注册失败, 将文件移至ERRORDIR目录; 同时将注册操作完成情况报告系统。在WINDOWS系统中, 对于文件的搜索, 系统也提供了相关的API函数:

(1) 启动查找函数: FindFirstFile()

它有两个参数, 第一个是要查找的文件名, 第二个是保存查找到的信息, 类型为WIN32\_FIND\_DATA, 返回一个句柄。

(2) 继续查找: FindNextFile()

(3) 关闭查找: FindClose()

程序的实现:

```
{
WIN32_FIND_DATA fd;
HANDLE hd = ::FindFirstFile(LPCTSTR)c:
\\comdir\\addnew\\*.dll",&fd) //开始查找;
if(hd==INVALID_HANDLE_VALUE) {return;}
//查找到的结果保存在: fd.cFileName
中; 可以处理, 如注册等
while(FindNextFile(hd,&fd)) //继续查找
{。。。//处理事宜。
};
FindClose(hd); //关闭查找
```

```
}
```

### 1.4 COM 组件的注册实现

在COM组件规范中, 一般都提供了标准的注册和注销的函数:

#### 1.4.1 注册函数

```
STDAPI DllRegisterServer(void)
{
// registers object, typelib and all inter-
faces in typelib
return _Module.RegisterServer(TRUE);
}
```

#### 1.4.2 注销函数

```
// DllUnregisterServer - Removes entries
from the system registry
STDAPI DllUnregisterServer(void)
{ return _Module.UnregisterServer(TRUE);
}
```

#### 1.4.3 调用注册函数的程序实现

```
::CoInitialize(NULL); //COM 初始化
//以注册ATL1.DLL 为例
HINSTANCE hLib=LoadLibrary("C:
\\COMDIR\\REGDIR\\ATL1.DLL");
if(hLib==NULL)
{
//:不能装载库文件";
//使用MoveFileEx()函数将文件移至C:\\
COMDIR\\ERRORDIR\\目录中
return ;
}
FARPROC pDllReg = ::GetProcAddress
```

```
((HMODULE)hLib,"DllRegisterServer");
if(pDllReg==NULL)
{ // : 获取注册组件函数
DllRegisterServer()地址失败;
//使用MoveFileEx()函数将文件移至C:\\
COMDIR\\ERRORDIR\\目录中
::FreeLibrary((HMODULE)hLib);
return ;
}
int ret=(*pDllReg); //调用、执行
组件注册函数
if(ret==0){ //:注册成功返回为零;
}
else { //注册失败!";
//使用MoveFileEx()函数将文件移至C:\\
COMDIR\\ERRORDIR\\目录中
}
::FreeLibrary((HMODULE)hLib);
::CoUninitialize();
return;
```

## 2 COM 智能注册系统实例演示

一个完美的COM智能注册系统最好是基于NT SERVICE的后台服务程序; 但是为了演示系统运行结果, 故采用了基于对话框的应用程序。

启动该系统后, 当有COM组件文件(\*.dll)装入ADDNEWDIR目录中时, 系统可以非常敏感地执行注册。见图1示例。

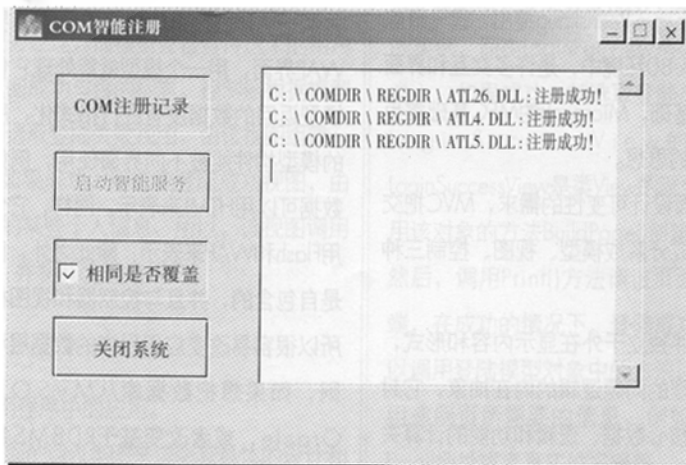


图1 系统运行示意图