SHSTDAPI\_(DWORD\_PTR) SHGetFileInfoA(LPCSTR pszPath, DWORD dwFileAttributes, SHFILEINFOA \*psfi, UINT cbFileInfo, UINT uFlags);

typedef struct \_SHFILEINFOA

{

HICON hIcon; // out: icon

int iIcon; // out: icon index

DWORD dwAttributes; // out: SFGAO\_ flags

CHAR szDisplayName[MAX\_PATH]; // out: display name (or path)

CHAR szTypeName[80]; // out: type name

} SHFILEINFOA;

DWORD dwAttributes = 0;

UINT nFlag = 0x00000000;

CString szFile;

GetDlgItemText(IDC\_EDIT1, szFile);

if(IsDlgButtonChecked(IDC\_CHECKDisplayName))

{

nFlag |= SHGFI\_DISPLAYNAME; //顯示名

}

if(IsDlgButtonChecked(IDC\_CHECKTypeName))

{

nFlag |= SHGFI\_TYPENAME; //類型名

}

if(IsDlgButtonChecked(IDC\_CHECKICON))

{

nFlag |= SHGFI\_ICON; //ICON

}

if(IsDlgButtonChecked(IDC\_CHECKSFGAO\_flags))

{

nFlag |= SHGFI\_ATTRIBUTES; //att

}

if(IsDlgButtonChecked(IDC\_CHECKExeTYpe))

{

nFlag |= SHGFI\_EXETYPE; //exetype

}

SHFILEINFO fileInfo;

memset(&fileInfo, 0x00 ,sizeof(fileInfo));

DWORD dwRet = SHGetFileInfo(

szFile,

dwAttributes,

&fileInfo,

sizeof(fileInfo),

nFlag);

CString str;

str.Format("%d",dwRet);

SetDlgItemText(IDC\_EDIT\_RETURNCODE,str); //返回

SetDlgItemText(IDC\_EDIT\_DISNAME,fileInfo.szDisplayName); //顯示名

SetDlgItemText(IDC\_EDIT\_TYPENAME,fileInfo.szTypeName); //類型名

str.Empty();

if(fileInfo.dwAttributes & SFGAO\_CANCOPY)

{

str += "SFGAO\_CANCOPY|";

}

if(fileInfo.dwAttributes & SFGAO\_CANMOVE)

{

str += "SFGAO\_CANMOVE|";

}

if(fileInfo.dwAttributes & SFGAO\_CANLINK)

{

str += "SFGAO\_CANLINK|";

}

if(fileInfo.dwAttributes & SFGAO\_STORAGE)

{

str += "SFGAO\_STORAGE|";

}

if(fileInfo.dwAttributes & SFGAO\_CANRENAME)

{

str += "SFGAO\_CANRENAME|";

}

if(fileInfo.dwAttributes & SFGAO\_CANDELETE)

{

str += "SFGAO\_CANDELETE|";

}

if(fileInfo.dwAttributes & SFGAO\_HASPROPSHEET)

{

str += "SFGAO\_HASPROPSHEET|";

}

if(fileInfo.dwAttributes & SFGAO\_SHARE)

{

str += "SFGAO\_SHARE|";

}

if(fileInfo.dwAttributes & SFGAO\_HASSUBFOLDER)

{

str += "SFGAO\_HASSUBFOLDER|";

}

if(fileInfo.dwAttributes & SFGAO\_REMOVABLE)

{

str += "SFGAO\_REMOVABLE|";

}

if(fileInfo.dwAttributes & SFGAO\_FOLDER)

{

str += "SFGAO\_FOLDER|";

}

SetDlgItemText(IDC\_EDIT\_ATT,str); //att

//ICON

str.Format("%d",fileInfo.iIcon);

SetDlgItemText(IDC\_EDITICONINDEX,str); //ICON索引

HDC hdc =this->GetDC()->Detach();

::DrawIcon(hdc, 206\*2, 65\*2, fileInfo.hIcon);

::ReleaseDC(m\_hWnd, hdc);

|  |  |  |
| --- | --- | --- |
| **代码** | **值** | **描述** |
| SHGFI\_ICON | 0x0100 | 将文件的HICON类型的图标Handle存储到结构SHFILEINFO的hIcon成员中。 |
| SHGFI\_DISPLAYNAME | 0x0200 | 将指向文件显示名串的指针存储到结构SHFILEINFO的szDisplayName成员中。 |
| SHGFI\_TYPENAME | 0x0400 | 将指向文件类型串的指针存储到结构SHFILEINFO的szTypeName成员中。 |
| SHGFI\_ATTRIBUTES | 0x0800 | 将DWORD类型的给定文件所有要恢的属性值存储到SHFILEINFO结构的dwAttributes成员中。 |
| SHGFI\_ICONLOCATION | 0x1000 | 将指向包含了Shell正在用于指定对象的图标的文件名串的指针存储到SHFILEINFO结构的szDisplayName中。因此，这个标志不能和SHGFI\_DISPLAYNAME标志一起使用。奇怪的是，它似乎仅在指定文件夹时才能工作，指定文件名则总是返回空。 |
| SHGFI\_EXETYPE | 0x2000 | 引起函数返回一个表示可执行文件二进制格式和它的目标平台的值。 |
| SHGFI\_SYSICONINDEX | 0x4000 | 引起函数返回一个包含图标的系统图像列表Handle。图标的索引存储在SHFILEINFO 结构的iIcon字段中。 |

|  |  |  |
| --- | --- | --- |
| **代码** | **值** | **描述** |
| SHGFI\_LARGEICON | 0x00000 | 引起函数恢复文件的大图标。 |
| SHGFI\_SMALLICON | 0x00001 | 引起函数恢复文件的小图标。 |
| SHGFI\_OPENICON | 0x00002 | 对于文件夹，这个设置引起函数恢复它打开时显示的图标。 |
| SHGFI\_SHELLICONSIZE | 0x00004 | 引起函数恢复具有一定尺寸的图标，这个图标尺寸在显示控制板的‘外观’标签中设置。 |
| SHGFI\_SELECTED | 0x10000 | 恢复的图标是一个文件选中时显示的图标(与高亮颜色混合的)。 |
| SHGFI\_LINKOVERLAY | 0x08000 | 恢复的图标是一个文件作为快捷方式显示的图标。 |

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | | | **描述** |
| SFGAO\_CANCOPY | | | 文件对象可以通过拖拽或剪裁板进行拷贝 |
| SFGAO\_CANDELETE | | | 可以通过Shell删除文件对象 |
| SFGAO\_CANLINK | | | 文件对象可以建立快捷方式 |
| SFGAO\_CANMOVE | | | 文件对象可以通过拖拽或剪裁板移动 |
| SFGAO\_CANRENAME | | | 文件对象可以通过Shell重命名 |
| SFGAO\_HASPROPSHEET | | | 文件对象至少有一张树形表单 |
| SFGAO\_GHOSTED | | | 问津对象使用精灵图标显示(一般是隐藏文件) |
| SFGAO\_LINK | | | 这个文件对象是一个快捷方式 |
| SFGAO\_READONLY | | | 这个文件对象是只读的 |
| SFGAO\_SHARE | | | 指定的文件夹是共享的 |
| SFGAO\_HASSUBFOLDER | | | 指定的文件夹至少有一个自文件夹 |
| SFGAO\_COMPRESSED | | | 文件对象驻留于压缩驱动器上 |
| SFGAO\_FILESYSTEM | | | 文件对象是文件系统得部件，不是虚拟文件夹。也就是说它是一个存在的物理对象(驱动器，目录或文件) |
| SFGAO\_FOLDER | | | 指定的对象是一个文件夹 |
| SFGAO\_REMOVABLE | | | 文件对象驻留于可移动介质上(典型地软盘) |
| **文件签名** | **Hex码** | **意义** | |
| PE | 0x4550 | Win32可执行格式，由微软所有32位操作系统采用。 | |
| NE | 0x454E | Windows 3.x新的可执行格式，典型地16位窗口程序 | |
| MZ | 0x5A4D | DOS 可执行格式，如果查询．com 或 ．bat也返回这个值。 | |