



**Forum: WD 7.x**

**Topic: Lien .lnk**

**Subject: Re: Lien .lnk**

Posté par: payalba

Contribution le : 10/6/2003 11:52:39

Voici un exemple tiré de MSDN. Je ne pense pas que cela soit adaptable pour WinDev :

PYT

//----

An application may need to access and manipulate a shortcut that was created previously. This operation is referred to as "resolving" the shortcut.

The application-defined `ResolveIt` function in the following example resolves a shortcut. Its parameters include a window handle, a pointer to the path of the shortcut, and the address of a buffer that receives the new path to the object. The window handle identifies the parent window for any message boxes that the shell may need to display. For example, the shell can display a message box if the link is on unshared media, if network problems occur, if the user needs to insert a floppy disk, and so on.

The `ResolveIt` function calls the `CoCreateInstance` function and assumes that the `CoInitialize` function has already been called. Note that `ResolveIt` needs to use the `IPersistFile` interface to store the link information. `IPersistFile` is implemented by the `IShellLink` object. The link information must be loaded before the path information is retrieved, which happens later in the example. Failing to load the link information causes the calls to the `IShellLink::GetPath` and `IShellLink::GetDescription` member functions to fail.

```
HRESULT ResolveIt(HWND hwnd, LPCSTR lpszLinkFile, LPSTR lpszPath)
```

```
{
```

```
    HRESULT hres;
```

```
    IShellLink* psl;
```

```
    char szGotPath[MAX_PATH];
```

```
    char szDescription[MAX_PATH];
```

```
    WIN32_FIND_DATA wfd;
```

```
    *lpszPath = 0; // assume failure
```

```
    // Get a pointer to the IShellLink interface.
```

```
    hres = CoCreateInstance(&CLSID_ShellLink, NULL,  
        CLSCTX_INPROC_SERVER, &IID_IShellLink, &psl);
```

```
    if (SUCCEEDED(hres)) {
```

```
        IPersistFile* ppf;
```

```
        // Get a pointer to the IPersistFile interface.
```

```
        hres = psl->lpVtbl->QueryInterface(psl, &IID_IPersistFile,  
            &ppf);
```

```

if (SUCCEEDED(hres)) {
    WORD wsz[MAX_PATH];

    // Ensure that the string is Unicode.
    MultiByteToWideChar(CP_ACP, 0, lpszLinkFile, -1, wsz,
        MAX_PATH);

    // Load the shortcut.
    hres = ppf->lpVtbl->Load(ppf, wsz, STGM_READ);
    if (SUCCEEDED(hres)) {

        // Resolve the link.
        hres = psl->lpVtbl->Resolve(psl, hwnd, SLR_ANY_MATCH);
        if (SUCCEEDED(hres)) {

            // Get the path to the link target.
            hres = psl->lpVtbl->GetPath(psl, szGotPath,
                MAX_PATH, (WIN32_FIND_DATA *)&wfd,
                SLGP_SHORTPATH );
            if (!SUCCEEDED(hres))
                HandleErr(hres); // application-defined function

            // Get the description of the target.
            hres = psl->lpVtbl->GetDescription(psl,
                szDescription, MAX_PATH);
            if (!SUCCEEDED(hres))
                HandleErr(hres);
            lstrcpy(lpszPath, szGotPath);
        }
    }
    // Release the pointer to the IPersistFile interface.
    ppf->lpVtbl->Release(ppf);
}
// Release the pointer to the IShellLink interface.

psl->lpVtbl->Release(psl);
}
return hres;
}

```