

LNK.TXT

Date: Mon, 11 Aug 1997 19:03:42 -0700

To: submit@wotsit.demon.co.uk

From: Elvis <elvis@iccom.com>

Subject: I see you are looking for info on windows .LNK file format....

It's been a while since I've messed with that, but I dug up the following source code. It has been so long that I can't remember if this was the file I got working, so someone should test it out (I don't have time). However, it basically decodes the windows .LNK file format to find out what it is a shortcut to, and determines if that is a directory (I was going to write a unix type ln command using .LNK file formats....). There are 2 files here. One is a C version, the other is a C++ version.... Hope this helps....

-Donald Murray

```
=====
C version
=====
```

```
#include <windows.h>
#include <windowsx.h>
#include <objbase.h>
#include <shlobj.h>
#include <stdio.h>
#include <initguid.h>
#include <string.h>

main(int ac, char *av[])
{
    IShellLink *psl;
    HRESULT hres;
    WIN32_FIND_DATA wfd;
    char szGotPath[MAX_PATH];
    IPersistFile *ppf;

    if (ac != 2)
    {
        printf("Syntax: ln <pathname>\n");
        return 0;
    }

    hres = CoInitialize(NULL);
    if (!SUCCEEDED(hres))
        printf("Could not open the COM library\n");

    hres = CoCreateInstance(&CLSID_ShellLink, NULL, CLSCTX_INPROC_SERVER,
                           &IID_IShellLink, (LPVOID *)&psl);
    if (SUCCEEDED(hres))
    {
        hres = psl->IPvtbl->QueryInterface(psl, &IID_IPersistFile, &ppf);
        if (SUCCEEDED(hres))
        {
            WORD wsz[MAX_PATH];

            MultiByteToWideChar(CP_ACP, 0, av[1], -1, wsz, MAX_PATH);

            hres = ppf->IPvtbl->Load(ppf, wsz, STGM_READ);
            if (SUCCEEDED(hres))
            {
                hres = psl->IPvtbl->Resolve(psl, 0, SLR_ANY_MATCH);
                if (SUCCEEDED(hres))
```

```

                                LNK.TXT
        {
            strcpy(szGotPath, av[1]);

            hres = psl->IPVtbl->GetPath(psl, szGotPath, MAX_PATH,
                                      (WIN32_FIND_DATA *)&wfd, SLGP_SHORTPATH);
            if (!SUCCEEDED(hres))
                printf("GetPath failed!\n");

            printf("This points to %s\n", wfd.cFileName);
            if (wfd.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)
                printf("This is a directory\n");
        }
    else
        printf("IPersistFile Load Error\n");
    ppf->IPVtbl->Release(ppf);
}
else
    printf("QueryInterface Error\n");
    psl->IPVtbl->Release(psl);
}
else
    printf("CoCreateInstance Error - hres = %08x\n", hres);
return 0;
}

```

```

=====
C++ version
=====

```

```

#include <windowsx.h>
#include <objbase.h>
#include <shlobj.h>
#include <stdio.h>
#include <initguid.h>
#include <stdlib.h>
#include <iostream>
#include <string.h>

```

```

// This program should print out whether the file is a link and where it
// points to and whether it is a directory or not.
//

```

```

main(int ac, char *av[])
{
    if (ac != 2)
    {
        printf("Syntax: In <pathname>\n");
        return 0;
    }

    IShellLink *psl;                                // pointer to IShellLink i/f
    HRESULT hres;
    WIN32_FIND_DATA wfd;
    char szGotPath[MAX_PATH];

    // Get pointer to the IShellLink interface.

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

```

LNK.TXT

```
if (SUCCEEDED(hres))
{
    // Get pointer to the IPersistFile interface.

    IPersistFile *ppf;
    hres = psl->QueryInterface(IID_IPersistFile, (LPVOID *)&ppf);

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

        // Ensure string is Uni code.
        MultiByteToWideChar(CP_ACP, 0, av[1], -1, wsz, MAX_PATH);

        // Load the shell link
        hres = ppf->Load(wsz, STGM_READ);
        if (SUCCEEDED(hres))
        {
            // Resolve the link.

            hres = psl->Resolve(0, SLR_ANY_MATCH);
            //
            // Using 0 instead of hWnd, as hWnd is only used if
            // interface needs to prompt for more information. Should use
            // hWnd from current console in the long run.

            if (SUCCEEDED(hres))
            {
                strcpy(szGotPath, av[1]);

                hres = psl->GetPath(szGotPath, MAX_PATH,
                                   (WIN32_FIND_DATA *)&wfd, SLGP_SHORTPATH );
                if (!SUCCEEDED(hres))
                    printf("GetPath failed!\n");

                printf("This points to %s\n", wfd.cFileName);
                if (wfd.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)
                    printf("This is a directory\n");
            }
        }
        else
            printf("IPersistFile Load Error\n");
        ppf->Release();
    }
    else
        printf("QueryInterface Error\n");
    psl->Release();
}
else
    printf("CoCreateInstance Error - hres = %08x\n", hres);
return 0;
}
```