

Finding a printer, and then creating a shortcut to that printer

devblogs.microsoft.com/oldnewthing/20130128-00

January 28, 2013



Raymond Chen

Today’s “Little Program” does two things: It looks for a printer in the Printers folder, and then once it finds it, it creates a shortcut to that printer.

As is common with “Little Programs”, I don’t bother with error checking. I’ll leave you to do that.

Second part first, since it is handy on its own: Creating a shortcut to an arbitrary item in the shell namespace, provided either in the form of an ID list or a shell item. (The ID list is the thing that identifies an item in the shell namespace.)

```
void CreateShortcutToIDList(PCWSTR pszName, PCUIDLIST_ABSOLUTE pidl)
{
    CComPtr<IShellLink> spsl;
    spsl.CoCreateInstance(CLSID_ShellLink);
    spsl->SetIDList(pidl);
    CComQIPtr<IPersistFile>(spsl)->Save(pszName, TRUE);
}
```

```
void CreateShortcutToItem(PCWSTR pszName, IShellItem *pitem)
{
    CComHeapPtr<ITEMIDLIST_ABSOLUTE> spidl;
    CComQIPtr<IPersistIDList>(pitem)->GetIDList(&spidl);
    CreateShortcutToIDList(pszName, spidl);
}
```

Neither of these is particular complicated. To create a shortcut given an ID list:

- Create a brand new `ShellLink` object.
- Tell that shell link object to point to our desired ID list.
- Save the shell link.

To create a shortcut given a shell item:

- Ask the `IShellItem` for its ID list.
- Create a shortcut to that ID list.

Okay, now the first half: Finding the printer. That is a matter of binding to the Printers folder and enumerating its contents. When we find the one whose name matches the target printer, we declare victory.

```
int __cdecl wmain(int argc, wchar_t **argv)
{
    CCoInitialize init;
    CComPtr<IShellItem> spPrinters;
    SHGetKnownFolderItem(FOLDERID_PrintersFolder, KF_FLAG_DEFAULT,
                        nullptr, IID_PPV_ARGS(&spPrinters));
    CComPtr<IEnumShellItems> spEnum;
    spPrinters->BindToHandler(nullptr, BHID_EnumItems,
                            IID_PPV_ARGS(&spEnum));
    for (CComPtr<IShellItem> spPrinter;
         spEnum->Next(1, &spPrinter, nullptr) == S_OK;
         spPrinter.Release()) {
        CComHeapPtr<wchar_t> spszName;
        spPrinter->GetDisplayName(SIGDN_NORMALDISPLAY, &spszName);
        wprintf(L"Found printer \"%ls\"\n", spszName);
        if (lstrcpw(spszName, argv[1]) == 0) {
            wprintf(L"Creating shortcut as \"%ls\"\n", argv[2]);
            CreateShortcutToItem(argv[2], spPrinter);
        }
    }
    return 0;
}
```

This is a little trickier, but not by much.

- Initialize COM.
- Get the `IShellItem` for the Printers folder.
- Get the enumerator for the Printers folder.
- For each item in the Printers folder:
 - Get its name and print it just for diagnostic purposes.
 - If the name matches the one we're looking for, then create the shortcut.

Here are the header files I used:

```
#define STRICT_TYPED_ITEMIDS // enable stricter type checking on ITEMIDS
#include <windows.h>
#include <atlbase.h> // for CComPtr, CComQIPtr
#include <atlalloc.h> // for CComHeapPtr
#include <shlobj.h> // for shell interfaces
#include <knownfolders.h> // for FOLDERID_PrintersFolder
#include <stdio.h> // for wprintf
```

For those of you who prefer to work with CSIDLs, the change is relatively minor. Replace the `SHGetKnownFolderItem` call with

```
BindToCsidlItem(CSIDL_PRINTERS, &spPrinters);
```

You may have noticed that I used the for-if anti-pattern. I could've gone for the item directly by using `SHCreateItemFromRelativeName : <!-- IShellFolder::ParseDisplayName : ->`

```
int __cdecl wmain(int argc, wchar_t **argv)
{
    CCoInitialize init;
    CComPtr<IShellItem> spPrinters;
    SHGetKnownFolderItem(FOLDERID_PrintersFolder, KF_FLAG_DEFAULT,
                        nullptr, IID_PPV_ARGS(&spPrinters));
    CComPtr<IShellItem> spPrinter;
    SHCreateItemFromRelativeName(spPrinters, argv[1], nullptr,
                                IID_PPV_ARGS(&spPrinter));
    CreateShortcutToItem(argv[2], spPrinter);
    return 0;
}
```

As before, if you're the sort of person who prefers to do things old-school, you can parse the name yourself, at which point you may as well give up on shell items, hike up your pants, and do it with an onion on your belt:

```

HRESULT BindToIDList(PCUIDLIST_ABSOLUTE pidl,
                    REFIID riid, void **ppv)
{
    *ppv = nullptr;
    CComPtr<IShellFolder> spsfDesktop;
    HRESULT hr = SHGetDesktopFolder(&spsfDesktop);
    if (SUCCEEDED(hr)) {
        if (pidl->mkid.cb) {
            hr = spsfDesktop->BindToObject(pidl, nullptr, riid, ppv);
        } else {
            hr = spsfDesktop->QueryInterface(riid, ppv);
        }
    }
    return hr;
}

```

```

int __cdecl wmain(int argc, wchar_t **argv)
{
    CCoInitialize init;
    CComHeapPtr<ITEMIDLIST_ABSOLUTE> spidlPrinters;
    SHGetSpecialFolderLocation(nullptr,
                              CSIDL_PRINTERS, &spidlPrinters);
    CComPtr<IShellFolder> spsfPrinters;
    BindToIDList(spidlPrinters, IID_PPV_ARGS(&spsfPrinters));

    ULONG cchEaten;
    DWORD dwAttributes = 0;
    CComHeapPtr<ITEMIDLIST_RELATIVE> spidl;
    spsfPrinters->ParseDisplayName(nullptr, nullptr, argv[1],
                                  &cchEaten, &spidl, &dwAttributes);

    CComHeapPtr<ITEMIDLIST_ABSOLUTE> spidlPrinter;
    spidlPrinter.Attach(ILCombine(spidlPrinters, spidl));
    CreateShortcutToIDList(argv[2], spidlPrinter);
    return 0;
}

```

The `BindToIDList` function is nothing special; we already saw the guts of it when we wrote [BindToCsidlItem](#).

The main program proceeds in three steps:

- Get the ID list for the Printers folder and bind to it.
- Parse the printer name, producing a printer ID list.
- Create a shortcut to the ID list for the printers folder combined with the printer ID list.



Raymond Chen

Follow