

# How can I get the canonical name for a known folder?

 [devblogs.microsoft.com/oldnewthing/20160208-00](http://devblogs.microsoft.com/oldnewthing/20160208-00)

February 8, 2016



Raymond Chen

A customer had a question about the `IKnownFolder::GetFolderByName` method: “What is the canonical name for the Documents folder? We tried “Documents” but that didn’t work.”

One question that comes to mind is why you are using this method to begin with. If you already know that you want the Documents folder, then use `IKnownFolder::GetFolder` and pass `FOLDERID_Documents`. Just go straight for the thing you want; don’t play around with canonical names.

But okay, let’s answer the question anyway. The way to get the canonical name for a folder is to ask it!

Today’s smart pointer library is (rolls dice) ATL!

```
#include <windows.h>
#include <shlobj.h>
#include <stdio.h> // horrors! Mixing stdio and C++!
#include <atlbase.h>
#include <atlalloc.h>

int __cdecl main()
{
    CCoInitialize init;

    CComPtr<IKnownFolderManager> mgr;
    CoCreateInstance(CLSID_KnownFolderManager, 0,
                    CLSCTX_ALL, IID_PPV_ARGS(&mgr));

    CComPtr<IKnownFolder> kf;
    mgr->GetFolder(FOLDERID_Documents, &kf);
    KNOWNFOLDER_DEFINITION def;
    kf->GetFolderDefinition(&def);
    printf("%ls\n", def.pszName);
    FreeKnownFolderDefinitionFields(&def);
    return 0;
}
```

Run this program and it tells you that the canonical name is `Personal`.

Let's go one step further: Let's print the canonical names for *all* of the known folders.

```
#include <windows.h>
#include <shlobj.h>
#include <stdio.h> // horrors! Mixing stdio and C++!
#include <atlbase.h>
#include <atlalloc.h>

int __cdecl main()
{
    CCoInitialize init;

    CComPtr<IKnownFolderManager> mgr;
    CoCreateInstance(CLSID_KnownFolderManager, 0,
                    CLSCTX_ALL, IID_PPV_ARGS(&mgr));

    UINT count;
    CComHeapPtr<KNOWNFOLDERID> kfids;
    mgr->GetFolderIds(&kfids, &count);
    for (UINT index = 0; index < count; index++) {
        CComPtr<IKnownFolder> kf;
        mgr->GetFolder(kfids[index], &kf);
        KNOWNFOLDER_DEFINITION def;
        kf->GetFolderDefinition(&def);
        printf("%ls\n", def.pszName);
        FreeKnownFolderDefinitionFields(&def);
    }
    return 0;
}
```

Raymond Chen

**Follow**

