

Opening the classic folder browser dialog with a specific folder preselected

 devblogs.microsoft.com/oldnewthing/20150406-00

April 6, 2015



Raymond Chen

Today's Little Program shows how to set the initial selection in the `SHBrowseForFolder` dialog.

The design of the `SHBrowseForFolder` function had a defect: The `BROWSEINFO` structure doesn't have a `cbSize` member at the start. This means that the structure cannot ever change because the function would have no way of knowing whether you are calling with the old structure or the new one. If it weren't for this defect, setting the initial selection would have been easy: Add a `pidlInitialSelection` member to the structure and have people fill it in.

Alas, any new functionality in the `SHBrowseForFolder` function therefore requires that the new functionality be expressed within the constraints of the existing structure.

Fortunately, there's a callback that takes a message number.

The workaround, therefore, is to express any new functionality in the form of new callback messages.

And that's how the ability to set the initial selection in the folder browser dialog came about. A new message `BFFM_INITIALIZED` was created, and in handling that message, the callback can specify what it wants the selection to be.

```

#define UNICODE
#define _UNICODE
#define STRICT_TYPED_ITEMIDS
#include <windows.h>
#include <ole2.h>
#include <oleauto.h>
#include <shlobj.h>
#include <stdio.h> // horrors! Mixing C and C++!

int CALLBACK Callback(
    HWND hwnd, UINT uMsg, LPARAM lParam, LPARAM lpData)
{
    switch (uMsg) {
    case BFFM_INITIALIZED:
        SendMessage(hwnd, BFFM_SETSELECTION, TRUE,
            reinterpret_cast<LPARAM>(L"C:\\Windows"));
        break;
    }
    return 0;
}

int __cdecl wmain(int, wchar_t **)
{
    CCoInitialize init;
    TCHAR szDisplayName[MAX_PATH];
    BROWSEINFO info = { };
    info.pszDisplayName = szDisplayName;
    info.lpszTitle = TEXT("Pick a folder");
    info.ulFlags = BIF_RETURNONLYFSDIRS;
    info.lpfncb = Callback;
    PIDLIST_ABSOLUTE pidl = SHBrowseForFolder(&info);
    if (pidl) {
        SHGetPathFromIDList(pidl, szDisplayName);
        wprintf(L"You chose %ls\n", szDisplayName);
        CoTaskMemFree(pidl);
    }
    return 0;
}

```

We initialize COM and then call the `SHBrowseForFolder` function with information that includes a callback. The callback responds to the `BFFM_INITIALIZED` message by changing the selection.

It's not hard, but it's a bit cumbersome.

Sorry.

Bonus chatter: The presence of the callback means that the function cannot shunt the work to a new thread when called from an MTA thread because you are now stuck with the problem of which thread the callback should run on.

- The callback may want to access resources that belong to the original thread, so that argues for the callback being run on the original thread.
- The callback may also want to access resources inside the dialog box, say in order to customize it. That argues for the callback being run on the new thread.

You can't have it both ways, so you're stuck.

But it doesn't really matter, because you shouldn't be performing UI from a multi-threaded apartment anyway. There's not much point in making it easier for people to do the wrong thing.

Raymond Chen

Follow

