Customizing item enumeration with IShellItem

devblogs.microsoft.com/oldnewthing/20150126-00

January 26, 2015

Raymond Chen

If you are using the original `IShellFolder` interface, then you can use `SHCONTF` values to customize how child items are enumerated. But what if you're using `IShellItem`?

Let's take it one step at a time. First, the basic program. (Remember, Little Programs do little to no error checking.)

```
#define UNICODE
#define _UNICODE
#define STRICT
#define STRICT_TYPED_ITEMIDS
#include <windows.h>
#include <shlobj.h>
#include <atlbase.h>
#include <atlalloc.h>
#include <propsys.h>
#include <stdio.h>

int __cdecl wmain(int argc, wchar_t **argv)
{
 CCoInitialize init;

 if (argc < 2) return 0;
 CComPtr<IShellItem> spsiFolder;
 SHCreateItemFromParsingName(argv[1], nullptr,
                             IID_PPV_ARGS(&spsiFolder));

 CComPtr<IEnumShellItems> spesi;
 spsiFolder->BindToHandler(nullptr, BHID_EnumItems,
                              IID_PPV_ARGS(&spesi));

 for (CComPtr<IShellItem> spsi;
      spesi->Next(1, &spsi, nullptr) == S_OK;
      spsi.Release()) {
  PrintDisplayName(spsi, SIGDN_NORMALDISPLAY, L"Display Name");
  PrintDisplayName(spsi, SIGDN_FILESYSPATH, L"File system path");
  wprintf(L"\n");
 }
 return 0;
}
```

Run this program with the fully-qualified path to a directory as the command line argument, and it enumerates all the items in the folder. This uses the default enumeration, which is “include both folders and non-folders, and include hidden items, but not super-hidden items.”

But what if we want to customize the enumeration?

We saw that the `IBindCtx` parameter acts as a catch-all *options* parameter. In this case, we need to look at the options available for `BHID_EnumItems` and see if any of them help us.

Fortunately, we have `STR_ENUM_ITEMS_FLAGS` which lets us override the default enumeration mode. Let’s say that we want only folders, and we want to respect the user’s preferences for hidden items (which means that we omit `SHCONTF_HIDDEN`).

I’m goint to do this two ways. First the flat version:

```
...
CComPtr<IBindCtx> spbcEnum;
CreateDwordBindCtx(STR_ENUM_ITEMS_FLAGS, SHCONTF_FOLDERS,
                   &spbcEnum);

CComPtr<IEnumShellItems> spesi;
spsiFolder->BindToHandler(spbcEnum, BHID_EnumItems,
                             IID_PPV_ARGS(&spesi));
```

Now the fluent version:

```
...
CBindCtxBuilder builder;
builder.SetVariantDword(STR_ENUM_ITEMS_FLAGS, SHCONTF_FOLDERS);

CComPtr<IEnumShellItems> spesi;
spsiFolder->BindToHandler(builder.GetBindCtx(), BHID_EnumItems,
                             IID_PPV_ARGS(&spesi));
```

(Don’t forget that error checking has been elided for expository purposes.)

The `STR_ENUM_ITEMS_FLAGS` bind context string was added in Windows 8, so it has no effect on older versions of Windows. We’ll address this next week.

Note that the `IEnumShellItems` interface is incorrectly-named. The convention for enumeration interfaces is to name them `IEnumXXX`, where XXX is *singular*.

Raymond Chen

Follow