Generally speaking, if your function fails, you should return a failure code

devblogs.microsoft.com/oldnewthing/20110610-00

June 10, 2011

Raymond Chen

A customer requested assistance with their shell namespace extension, and the request worked its way to me for resolution.

```
Unhandled exception at 0x76fab89c (shell32.dll) in explorer.exe: 0xC0000005:
Access violation reading location 0x00000000.
shell32.dll!CShellItem::_GetPropertyStoreWorker()  + 0x44 bytes
shell32.dll!CShellItem::GetPropertyStoreForKeys()  + 0x38 bytes
thumbcache.dll!CThumbnailCache::_GetMonikerDataFromShellItem()  + 0x8b bytes
thumbcache.dll!CThumbnailCache::GetThumbnail()  + 0x11c bytes
shell32.dll!CSetOperationCallback::_LookupThumbnail()  + 0x8d bytes
shell32.dll!CSetOperationCallback::_PrefetchCachedThumbnails()  + 0xb6 bytes
shell32.dll!CSetOperationCallback::OnNextBatch()  + 0x4f bytes
shell32.dll!CEnumTask::_PushBatchToView()  + 0x68 bytes
shell32.dll!CEnumTask::_IncrFillEnumToView()  + 0x2ca5 bytes
shell32.dll!CEnumTask::_IncrEnumFolder()  + 0x8da5a bytes
shell32.dll!CEnumTask::InternalResumeRT()  + 0xa6 bytes
shell32.dll!CRunnableTask::Run()  + 0x92 bytes
browseui.dll!CShellTask::TT_Run()  + 0x2d bytes
browseui.dll!CShellTaskThread::ThreadProc()  + 0x87 bytes
browseui.dll!CShellTaskThread::s_ThreadProc()  + 0x21 bytes
shlwapi.dll!_ExecuteWorkItemThreadProc@4()  + 0xe bytes
ntdll.dll!_RtlpTpWorkCallback@8()  + 0xaa bytes
ntdll.dll!_TppWorkerThread@4()  + 0x274 bytes
kernel32.dll!@BaseThreadInitThunk@12()  + 0x12 bytes
ntdll.dll!__RtlUserThreadStart@8()  + 0x27 bytes
```

The customer was at a loss because the customer's code was nowhere on the stack. What is wrong?

The customer didn't provide a dump file or any other information beyond the stack trace. (Hint: When reporting a problem with a shell namespace extension, at least mention the last few method calls your namespace extension received before the crash.) I was forced to use my psychic powers to solve the problem. But you can, too. All the information you need is right there in front of you.

The shell faulted on a null pointer in the function `CShellItem::_GetPropertyStoreWorker`, which from its name is clearly a worker function which obtains the property store from a shell item.

At this point, you put on your thinking cap. Why is the shell taking a null pointer fault trying to retrieve the property store from a shell item? Remember that the problem is tied to a custom namespace extension.

> My psychic powers tell me that the namespace extension returned `S_OK` from `GetUIObjectOf(IPropertyStoreFactory)` but set the output pointer to `NULL`.

(It turns out my psychic powers were weak without coffee, because the initial psychic diagnosis was `GetUIObjecttOf(IPropertyStore)` instead of `IPropertyStoreFactory`.)

As a general rule, if your function fails, then you should return a failure code, not a success code. There are exceptions to this rule, particular when OLE automation is involved, but it's a good rule to start with.

The customer reported that fixing their `IShellFolder::BindToObject` to return an error code when it failed fixed the problem. The customer then followed up with another crash, again providing startling little information.

```
Unhandled exception at 0x763cf7e7 (shell32.dll) in explorer.exe: 0xC0000005:
Access violation reading location 0x000a0d70.
Call Stack:
shell32.dll!CInfotipTask::InternalResumeRT() + 0x2e bytes
shell32.dll!CRunnableTask::Run() + 0x92 bytes
browseui.dll!CShellTask::TT_Run() + 0x2d bytes
browseui.dll!CShellTaskThread::ThreadProc() + 0x87 bytes
browseui.dll!CShellTaskThread::s_ThreadProc() + 0x21 bytes
shlwapi.dll!_ExecuteWorkItemThreadProc@4() + 0xe bytes
ntdll.dll!_RtlpTpWorkCallback@8() + 0xaa bytes
ntdll.dll!_TppWorkerThread@4() + 0x274 bytes
kernel32.dll!@BaseThreadInitThunk@12() + 0x12 bytes
ntdll.dll!__RtlUserThreadStart@8() + 0x27 bytes
```

The customer reported that `IQueryInfo::SetInfoTip` is getting called. The customer liaison added, "Raymond, I'm looking forward to your psychic powers again."

Apparently, some people don't understand that psychic powers are not something you ask for. It's my way of scolding you for not providing enough information to make a quality diagnosis possible. You don't come back saying, "Hey, thanks for answering my question even though I did a crappy job of asking it. Here's another crappy question!"

I reported back that my psychic powers were growing weary from overuse, and that the customer might want to expend a little more time investigating the problem themselves. Especially since it has the same root cause as their previous problem.

[Raymond is currently away; this message was pre-recorded.]

Raymond Chen

Follow