

Redirecting the Favorites folder is technically legal, but you wouldn't like it

 devblogs.microsoft.com/oldnewthing/20140702-00

July 2, 2014



Raymond Chen

A customer liaison asked for assistance in debugging why Internet Explorer frequently stops responding at their customer's site. "We have tried a number of things like clearing the temporary Internet files, disabling add-ons, and resetting Internet Explorer settings. Please let me know if you can provide guidance based on the dump files provided below to indicate what could be causing Internet Explorer to hang here."

The dump file showed 23 threads, and all of them seemed to be normal, except for one which looked like this:

ntdll!KiFastSystemCallRet
ntdll!ZwOpenFile+0xc
kernel32!BaseDllOpenIniFileOnDisk+0x1ec
kernel32!BaseDllReadWriteIniFileOnDisk+0x22
kernel32!BaseDllReadWriteIniFile+0x154
kernel32!GetPrivateProfileStringW+0x35
kernel32!GetPrivateProfileSectionNamesW+0x18
shell32!CPrivateProfile::_GetPrivateProfileAlloc+0x9e
shell32!CPrivateProfile::_GetSectionNames+0xa0
ieframe!CINIPropSetStg::_Load+0x74
ieframe!CInternetShortcutPropertyStore::_CreateStoreFromFile+0x4e
ieframe!CInternetShortcutPropertyStore::_Load+0x22
ieframe!CInternetShortcut::_LoadFromFileW+0x39
ieframe!CInternetShortcut::_Initialize+0x17
shell32!InitializeFileHandlerWithFile+0x2d
shell32!CFileSysItemString::_HandlerCreateInstance+0x29a
shell32!CFileSysItemString::_LoadHandler+0x91
shell32!CFSFolder::_CreatePerInstanceDefExtIcon+0x7d
shell32!CFSFolder::_CreateDefExtIcon+0xe9
shell32!CFSFolder::_s_GetExtractIcon+0x1b
shell32!CFSFolder::_BindHandler+0x209
shell32!CFSFolder::_GetUIObjectOf+0x21
shell32!GetExtractIconW+0x31
shell32!_GetILIndexFromItem+0x52
shell32!SHMapPIDLToSystemImageListIndex+0x37
ieframe!OrderItem_GetSystemImageListIndex+0x187
ieframe!CSFToolbar::_GetBitmap+0x2f
ieframe!CSFToolbar::_OnGetDispInfo+0x34
ieframe!CSFToolbar::_OnNotify+0x8e
ieframe!CISFBand::_OnNotify+0x2c
ieframe!CSFToolbar::_OnWinEvent+0x89
ieframe!_FwdWinEvent+0x1d
ieframe!CBandSite::_SendToToolband+0x44
ieframe!CInternetToolbar::_OnNotify+0x2e
ieframe!CInternetToolbar::_SizableWndProc+0x223
user32!InternalCallWinProc+0x23
user32!UserCallWinProcCheckWow+0x14b
user32!SendMessageWorker+0x4b7
user32!SendMessageW+0x7c
comctl32!CCSendNotify+0xbfb
comctl32!SendNotifyEx+0x63
comctl32!CReBar::_WndProc+0x24f
comctl32!CReBar::_s_WndProc+0x2c
user32!InternalCallWinProc+0x23
user32!UserCallWinProcCheckWow+0x14b
user32!CallWindowProcAorW+0x97
user32!CallWindowProcW+0x1b
comctl32!CallOriginalWndProc+0x1a
comctl32!CallNextSubclassProc+0x3d
comctl32!DefSubclassProc+0x46
ieframe!CInternetToolbar::CITBandSite::_s_RebarSubclassWndProc+0x5a
comctl32!CallNextSubclassProc+0x3d

```
comctl32!MasterSubclassProc+0x54
user32!InternalCallWinProc+0x23
user32!UserCallWinProcCheckWow+0x14b
user32!SendMessageWorker+0x4b7
user32!SendMessageW+0x7c
comctl32!CCSendNotify+0xbfb
comctl32!CToolBar::TBGetItem+0x2c
comctl32!CToolBar::DrawButton+0x5e9
comctl32!CToolBar::DrawToolBarH+0x1ad
comctl32!CToolBar::TBPaintImpl+0xd5
comctl32!CToolBar::TBPaint+0x18c
comctl32!CToolBar::ToolBarWndProc+0xd2e
comctl32!CToolBar::s_ToolbarWndProc+0x9b
user32!InternalCallWinProc+0x23
user32!UserCallWinProcCheckWow+0x14b
user32!CallWindowProcAorW+0x97
user32!CallWindowProcW+0x1b
comctl32!CallOriginalWndProc+0x1a
comctl32!CallNextSubclassProc+0x3d
comctl32!DefSubclassProc+0x46
ieframe!CSFToolBar::_DefWindowProc+0xb8
ieframe!CISFBand::_DefWindowProc+0x75
ieframe!CNotifySubclassWndProc::s_SubclassWndProc+0xb4
comctl32!CallNextSubclassProc+0x3d
comctl32!MasterSubclassProc+0x54
user32!InternalCallWinProc+0x23
user32!UserCallWinProcCheckWow+0x14b
user32!DispatchClientMessage+0xda
user32!__fnDWORD+0x24
ntdll!KiUserCallbackDispatcher+0x2e
user32!NtUserDispatchMessage+0xc
user32!DispatchMessageWorker+0x38c
user32!DispatchMessageW+0xf
ieframe!CTabWindow::_TabWindowThreadProc+0x280
kernel32!BaseThreadInitThunk+0xe
ntdll!__RtlUserThreadStart+0x23
ntdll!_RtlUserThreadStart+0x1b
```

(Remember, you can get symbols for operating system binaries.)

General debugging tip: If you see a really huge stack, that's a good sign that something interesting is going on. Boring stacks tend to be small.

Furthermore, frames near the bottom of the stack tend to describe what the purpose of the thread is, whereas frames near the top of the stack tend to describe what the thread is actually doing right now. (Exercise: Why?)

In this case, we see a stack that was probably created to manage a tab window (`CTab-Window::_TabWindowThreadProc`) and it's currently stuck in an I/O operation. You can then look at the file name to see what file is stuck.

```
0:001> du 04cd6aac
04cd6aac "\\server\share\abcdefg\Favorites\Mail.url"
```

It looks like this user stored their Favorites on a network share that is not responding.

The customer liaison replied,

Thanks a lot for this information. Can you help me understand how do we tell that the dump indicates this file I/O is causing IE to hang? Having this information would help me better explain this to the customer.

I wasn't sure how to respond to this. If you see a function with the words File and one of Open, Read, or Write in its name, there's a good chance that it opens, reads, or writes a file. You probably want to look to see what file is being opened, read from, or written to, because that may give a clue why the I/O operation is stuck.

It turns out that this customer redirected the user's Favorites to a network location. The Internet Explorer folks tell me that this is not an explicitly supported scenario in the sense that they did not do any tuning to make this scenario work well, and frequent hangs are consequently not unexpected. If you redirect the Favorites to a network location, then you get what you get. And if that server frequently becomes unavailable, then what you get often sucks.

Raymond Chen

Follow

