Why does Internet Explorer not call DLL_PROCESS_DETACH on my DLL when I call ExitProcess?

devblogs.microsoft.com/oldnewthing/20111118-00

November 18, 2011

Raymond Chen

A customer asked a question, but as is often the case, the question was much more telling than the answer.

> We have an Internet Explorer plug-in which calls `ExitProcess` to force Internet Explorer to exit. We found that when we do this, our plug-in does not receive a `DLL_PROCESS_DETACH` notification. What could be preventing our plug-in from receiving the `DLL_PROCESS_DETACH` notification?

As we saw some time ago when we looked at the way processes shut down (plus an important follow-up or two), all a process has to do to thwart proper delivery of `DLL_PROCESS_DETACH` notifications is to do something untoward during shutdown, at which point the kernel just gives up and calls `TerminateProcess`.

But like I said, the answer is much less interesting than the question. What if the user had an unsaved email message at the time you decided to exit Internet Explorer? Recall that plug-ins are a guest in the host process; don't go changing the carpet. When we asked the customer why they were exiting Internet Explorer from their plug-in, we received the explanation, "The reason I am calling `ExitProcess` is that I do not know another good way to exit Internet Explorer from a plug-in."

In this case, the guest is doing far more than just changing the carpet. The guest called in a demolition company!

"Why did you call the demolition company to destroy my house?"
"I couldn't think of a good way to destroy your house."

The point isn't that it's bad to use a telephone call to hire a demolition company to destroy somebody's house and that you should use some other method to contact them (like, say, a text message). The point is that *it's bad to destroy somebody else's house in the first place*.

Upon further investigation, the customer was writing a test for their plug-in. They open Internet Explorer and navigate to a page that uses the plug-in. When they are satisfied that the plug-in operated correctly, they want to exit the copy of Internet Explorer in order to conclude the test.

If you want to destroy a house, then destroy your own house. Call `CoCreate-Instance(CLSID_InternetExplorer)` to build a house, navigate to your test page with `IWebBrowser2::Navigate`, and when you're done, you can destroy the house with `IWeb-Browser2::Quit()`. There is sample code to do exactly this in the documentation for the `IWebBrowser2` interface.

Bonus chatter: The `IWebBrowser2` interface is scriptable.

```
var ie = new ActiveXObject("InternetExplorer.Application");
ie.Visible = true;
ie.Navigate("http://www.microsoft.com/");
WScript.Sleep(5000); // five seconds, say
ie.Quit();
```

Raymond Chen

Follow