A brief discussion on how best to respond to the end-session messages

blogs.msdn.microsoft.com/oldnewthing/20170329-00

Raymond Chen - MSFT March 29, 2017

A customer discovered that their application's shutdown code sometimes deadlocked. To address the problem, they moved the bulk of their shutdown code to the `WM_END SESSION` message handler. The customer found my earlier discussion of the `WM_END SESSION` message and wondered if they were doing the right thing.

Yes, it's okay to do shutdown activities in response to the `WM_END SESSION` message, provided that the `wParam` is nonzero, indicating that the session really is ending. If the `wParam` is zero, then it means that the session is *not* ending, so you had better not destroy anything you still need.

Recall the shutdown sequence: First, the application receives a `WM_QUERY END SESSION` message. Here is the traditional point at which you can display a prompt to ask the user whether they want to save their unsaved changes.[1] Normally, you return `TRUE`, but if the user hits *Cancel* or otherwise indicates that they don't want to shut down after all, then you return `FALSE`.

If you returned `TRUE`, then you will eventually receive a `WM_END SESSION` message whose `wParam` indicates whether the session really is ending. (The session might not actually be ending if another application returned `FALSE` to the `WM_QUERY END SESSION` message, or if the user canceled shutdown from the UI.)

The customer shared some of their code, and I noticed that they were destroying a window in their `WM_END SESSION` message handler, which is suspicious for two reasons:

1. If the `wParam` is `FALSE`, the application will continue to run, but it lost one of its windows!
2. If the `wParam` is `TRUE`, then it's okay to destroy things, but remember that you are running under a time constraint, and the building is being demolished, so you probably shouldn't be wasting time sweeping the floor and emptying the trash cans.

What you could do is to kick off a background thread to prepare for shutdown when you receive the `WM_QUERY END SESSION` message. For example, you might start an autosave operation. Whatever you do, make sure that it's okay for the operation to occur even if the shutdown is subsequently canceled.

When you get the `WM_END SESSION` message, you wait until that background operation completes before telling the system, "I'm good; you can shut down now."

Opportunistically starting the operation when you get the `WM_QUERY END SESSION` message means that you can respond more quickly to the `WM_END SESSION` message.

[1] In practice, displaying a prompt is usually not a good idea because if you don't respond to the message after a few seconds, the system will shut down without you.