Don't be helpless: I don't know anything about MFC modal loops, but unlike some people, I'm not afraid to find out

devblogs.microsoft.com/oldnewthing/20070521-00

May 21, 2007

Raymond Chen

Commenter Tom Grelinger asks via the Suggestion Box:

> If I have a modal CDialog that is visible and usable to the user. Let's say I receive an event somewhere else in the program and I call DestroyWindow on the modal CDialog from within the event. I notice that the OnDestroy is called on the CDialog, but DoModal never exits until a WM_QUIT is posted to the modal's message pump. What are the pitfalls to this? Unfortunately, there is really no way to avoid this situation.

I'm not sure what the question is, actually. The question as stated is "What are the pitfalls to this?" but he answered that in his own question: The pitfall is that "DoModal never exits until a `WM_QUIT` is posted to the modal dialog's message pump."

I'm going to assume that the question really is, "Why doesn't destroying the window work?" with the follow-up question, "What is the correct way to dismiss a modal dialog?"

The first problem with this question is that it assumes that I know what a `CDialog` is. From its name, I'm going to assume that this is an MFC class for managing a dialog box. But you don't even have to know that to answer the first reformulated question operating only from Win32 principles: `DestroyWindow` is not how you exit a modal dialog. You exit a modal dialog with `EndDialog`. The `DestroyWindow` technique is for *modeless* dialogs.

But let's look at the question another way, which is my point for today: You have the MFC source code. Don't be afraid to read it. Especially since I don't use MFC personally; I don't even know the basic principles of application design with MFC. I work in straight Win32. As a result, I don't know the answer off the top of my head, but fifteen minutes reading the MFC source code quickly reveals the reason why destroying the window doesn't work.

Watch me as I go and find out the answer. It's nothing you can't already do yourself.

The `CDialog::DoModal` method calls `CWnd::RunModalLoop` to run the dialog loop. If you look at `CWnd::RunModalLoop`, you can see the conditions under which it will exit the modal loop. Here's the code with irrelevant details deleted. (They're irrelevant because they have nothing to do with how the modal loop exits.)

```
int CWnd::RunModalLoop(DWORD dwFlags)
{
    ... preparatory work ...
    // acquire and dispatch messages until the modal state is done
    for (;;)
    {
        ... code that doesn't break out of the loop ...
        // phase2: pump messages while available
        do
        {
            // pump message, but quit on WM_QUIT
            if (!AfxGetThread()->PumpMessage())
            {
                AfxPostQuitMessage(0);
                return -1;
            }
            ... other code that doesn't break out of the loop ...
            if (!ContinueModal())
                goto ExitModal;
            ... other code that doesn't break the loop ...
        }  while (::PeekMessage(pMsg, NULL, NULL, NULL, PM_NOREMOVE))
    }
ExitModal:
    m_nFlags &= ~(WF_MODALLOOP|WF_CONTINUEMODAL);
    return m_nModalResult;
}
```

There are only two ways out of this loop. The first is the receipt of a `WM_QUIT` message. The second is if `CWnd::ContinueModal` decides that the modal loop is finished. The commenter already mentioned the quit message aspect to the modal loop, so that just leaves `CWnd::ContinueModal`.

The `CWnd::ContinueModal` method is very simple:

```
BOOL CWnd::ContinueModal()
{
    return m_nFlags & WF_CONTINUEMODAL;
}
```

Therefore, the only other way the loop can exit is if somebody clears the `WF_CONTINUEMODAL` flag. A little grepping shows that there are only three places where this flag is cleared. One is in `CPropertyPage`, which is a derived class of `CDialog` and therefore isn't relevant here. (I'll ignore `CPropertyPage` in future searches.) The second is in the line above right after the label `ExitModal`. And the third is this method:

```
void CWnd::EndModalLoop(int nResult)
{
    // this result will be returned from CWnd::RunModalLoop
    m_nModalResult = nResult;
    // make sure a message goes through to exit the modal loop
    if (m_nFlags & WF_CONTINUEMODAL)
    {
        m_nFlags &= ~WF_CONTINUEMODAL;
        PostMessage(WM_NULL);
    }
}
```

This method is called in only one place:

```
void CDialog::EndDialog(int nResult)
{
    if (m_nFlags & (WF_MODALLOOP|WF_CONTINUEMODAL))
        EndModalLoop(nResult);
    ::EndDialog(m_hWnd, nResult);
}
```

Following the money one last step, the `CDialog::EndDialog` method is called from four places in `CDialog`. It’s called from `CDialog::HandleInitDialog` and `CDialog::InitDialog` if some catastrophic error occurs during dialog initialization. And it's called from `CDialog::OnOK` and `CDialog::OnCancel` in response to the user clicking the OK or Cancel buttons.

Notice that the `CDialog::EndDialog` method is not called when somebody forcibly destroys the dialog from the outside.

That's why destroying the dialog window doesn't break the modal loop. If you want to break out of the modal loop, your only choices are to post a quit message or call `CWnd::EndModalLoop`, either directly or indirectly (via `CDialog::EndDialog`, for example).

Notice that the MFC modal loop obeys the convention on quit messages by re-posting the quit message when it breaks out of the modal loop. (Though it really should have posted the `wParam` from the quit message rather than just posting zero.)

The workaround therefore is not to destroy the dialog with `DestroyWindow` (something you should have known not to do *a priori* since that's not how you exit modal dialog boxes) but rather by calling `CDialog::EndDialog`, passing a result code that lets the caller of `CDialog::DoModal` know that the dialog box exited under unusual circumstances.

This took me fifteen minutes to research and a little over an hour to write up. All this work to answer a question that you should have been able to answer yourself with a little elbow grease. You're a smart person. Have confidence in yourself. You can do it. I know you can.

Raymond Chen

Follow