

Re: Thread Deadlock

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2004-09/1088.html>

From: Mihajlo Cvetanovic (mac_at_RnEeMtOsVeEt.co.yu)

Date: 09/14/04

Date: Tue, 14 Sep 2004 11:18:18 +0200

rob wrote:

> *Thanks for your input. I have a question, though. How can an app update
> itself when an event handler of this app creates a modal dialog? The app
> does not get out of the event handler until the dialog is closed. So how
> does the message loop in the modal dialog help the app to be updated? In my
> thread I can't directly call a function in the main app (I am not sending
> messages but do a direct call). So how can a function in the main app be
> executed when the app is stuck in the event handler that popped up the modal
> dialog?*

1. Message handling

Every window has a function that gets called when a message needs to be dispatched to it. This function is registered to a window, and this is hidden from you in MFC, but create non-MFC project and see what wizard gave you. This function is called from a Message Pump. All message pumps (including those in both MFC and non-MFC projects, dialog boxes, even in message boxes (created with [Afx]MessageBox) look pretty much like this:

```
while( (bRet = GetMessage( &msg, NULL, 0, 0 )) > 0)
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

So, message handler for one window (dialog) gets called from message handler of another window (main window). While in dialog, message pump can call main window's message handler just as well. Only thing to remember is that message pump must always be "alive", which means that you can't wait too long in one message handler and not do something that creates another message pump.

2. Call Stack of a Thread

When debugging observe the Call Stack. You'll see a number of functions one on top of another. It doesn't matter what are those functions, or whose classes are they belonging to. What matters is that every thread has its own call stack, totally independent of stacks of other threads.

microsoft.public.vc.mfc: Re: Thread Deadlock

To put it another way: threads have context, classes don't have context.
It doesn't matter if that class is derived from CWinApp, or whatever.

HTH