

Re: HOWTO: Notify owner from a DLL?

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2004-02/1060.html>

From: Joseph M. Newcomer (*newcomer_at_flounder.com*)

Date: 02/18/04

Date: Wed, 18 Feb 2004 01:37:23 -0500

I think it was Doug Harrison who pointed out a book on C++ design patterns that discusses a safe way to do this, but I'm not allowed to quote him any longer, so you'll have to check the archives.

There are some issues about program construction that simply require programs be well-behaved. Garbage-collected systems like C# and Java eliminate these problems entirely. Otherwise you have to rely on the user of your library to do the right thing. But frankly, if they don't, that's tough. You give them the specs; if they don't follow them, the phrase in common use is Life Is Hard.

joe

On Wed, 18 Feb 2004 02:37:29 GMT, "RickL" <zlinge@nospam.com> wrote:

>Thanks for the tips..
>
>I'm still a little confused about passing a string
>from the DLL to the client application.
>
>My DLL will have a serial port thread that can
>receive characters from a com port.
>Suppose I receive a 16 byte message. Now I want
>to notify the client application
>and also pass the received message to the client.
>
>Normally, my receive thread would allocate memory
>using 'new' and pass this
>pointer to the parent window as the LPARAM in
>::PostMessage(). The parent
>could do whatever it needs to with the string and
>then call 'delete' to delete the
>heap allocated string buffer.
>
>But it's not clear if I should implement a similar
>mechanism if the receive thread
>is now in a DLL. One problem I see is that I'd be
>relying on the client application
>to do the right thing and delete the heap
>allocated buffer.

>
>*Joe, you mentioned providing a function like*
>*ReleaseBuffer inside the DLL. I guess*
>*it's function would be to delete the buffer. But*
>*the client application will still have*
>*to call that function. In other words, they'll*
>*have to read my documentation!*
>
>
>
>*Thanks for your suggestions,*
>*RickL*
>
>
>
>
>*"Joseph M. Newcomer" <newcomer@flounder.com> wrote*
>*in message*
>*news:j1m23016vnaamgl8bgnhfue1gkqg7d9nuj@4ax.com...*
>*| David said everything I would. Note that the*
>*common method of using PostMessage to pass*
>*| data back is to allocate data on the stack using*
>*new or malloc and when the receiver has*
>*| processed it, it releases it using delete or*
>*free. If you choose this method, what you*
>*| will have to do is provide a function with some*
>*name like ReleaseBuffer which takes the*
>*| pointer you passed back via PostMessage, and*
>*calls free or delete inside the DLL. This*
>*| would be necessary if you chose to statically*
>*link the MFC library in, for the reasons*
>*| David mentions. But note that if you decide to*
>*use the MFC runtime DLL, this solution*
>*| still works correctly!*
>
>*| Note that you cannot use a statically-allocated*
>*buffer or stack-allocated buffer in the*
>*PostMessage, because you are effectively in a*
>*"multithreaded" situation (even if you*
>*aren't actually using threads; think of the*
>*invocation of the handler as appearing as if*
>*| it were a separate thread of control).*
>*Therefore, you must do heap allocation (actually,*
>*| if you don't mind a fixed upper bound on the*
>*total buffer space, you could write your own*
>*| mini-allocator to allocate your own static*
>*buffer, but all the other issues remain)*
>*| joe*
>
>*| On Mon, 16 Feb 2004 17:06:14 +0000, David*
>*Lowndes <davidl@example.invalid> wrote:*
>
>
>

microsoft.public.vc.mfc: Re: HOWTO: Notify owner from a DLL?

>|>>1. If I want the DLL to be useable by clients
>that
>|>>are non-MFC, is it correct that I should
>create a
>|>>regular DLL (as opposed to an extension DLL)?
>|>
>|>Yes, a regular DLL is all you need. MFC
>extension DLLs are special
>|>things for... extending MFC :)
>|>
>|>>2. I would prefer to statically link the DLL
>so
>|>>that I don't have to include the MFC DLL with
>the
>|>>distribution. Is static linking OK?
>|>
>|>It depends! If your interface to the calling
>application doesn't
>|>involve passing heap allocated objects (such as
>CString), then there's
>|>no problem. If you're passing allocated objects
>that need allocating,
>|>reallocating, freeing across the DLL/EXE
>boundary, then you have to
>|>ensure that both the EXE and DLL are built with
>the shared
>|>run-time/MFC (so that they use a common heap).
>|>
>|>>3. How do I notify a client using the DLL,
>when a
>|>>serial port event occurs? If I can still use
>|>>::PostMessage, then I guess I would need to
>get
>|>>the handle of the client app with some kind of
>|>>Init() function?
>|>
>|>It's best if you design your DLL interface such
>that the user has to
>|>supply you with the information you need to
>notify them of the event.
>|>
>|>Dave
>|
>| Joseph M. Newcomer [MVP]
>| email: newcomer@flounder.com
>| Web: <http://www.flounder.com>
>| MVP Tips: http://www.flounder.com/mvp_tips.htm
>

Joseph M. Newcomer [MVP]
email: newcomer@flounder.com

Re: HOWTO: Notify owner from a DLL?

microsoft.public.vc.mfc: Re: HOWTO: Notify owner from a DLL?

Web: <http://www.flounder.com>

MVP Tips: http://www.flounder.com/mvp_tips.htm