

microsoft.public.vc.mfc: Re: MultiThread and socket problem....(I'm making a simple EchoServer.)

Re: MultiThread and socket problem....(I'm making a simple EchoServer.)

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2004-06/1159.html>

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

Date: 06/11/04

Date: Thu, 10 Jun 2004 21:40:00 -0400

WM_COPYDATA is usually risky; I've had situations where apps we never heard of were doing SendMessage(WM_BROADCAST, WM_COPYDATA, ...) and these apps would crash our programs. See my essay on safe use of WM_COPYDATA to see how to make sure this can't happen to you.

If you have threads, you can't avoid multithreading issues.

If you want to send messages between threads, then you MUST use UI threads, not worker threads.

If the thread is not the main GUI thread, and itself has no user-visible windows, you can use PostThreadMessage to post messages to it. You can't use PostThreadMessage to the main GUI thread for reasons explained in the PostThreadMessage documentation.

If you use PostMessage, you must have a window (usually a top-level invisible window) which handles the message.

WM_COPYDATA must not be posted. So it is probably a poor choice of message to use. It can only be sent, and sending messages between threads is extremely risky and should not ever be done.

I see no reason to use WM_COPYDATA when an ordinary message or thread message will suffice. It only adds complexity without benefit.

You should not use WM_USER messages, and I don't even like WM_APP messages. I much prefer using Registered Window Messages, because they are less subject to misinterpretation. Again, if some dimwit does a SendMessage(HWND_BROADCAST, WM_APP+17, ...) message (using of course a symbolic value for that message ID), you get it and crash. You know, with a registered window message, that the sender is legitimate, unless you have some sociopathic program like a virus that uses registered window codes for "fun".

Read my essay on worker threads on my MVP Tips site. It explains how to send data between threads. In particular, it sends a CString * across the thread boundary.

joe

On 10 Jun 2004 18:12:11 -0700, bajy@intizen.com (Ryan Park) wrote:

Re: MultiThread and socket problem....(I'm making a simple EchoServer.)

microsoft.public.vc.mfc: Re: MultiThread and socket problem....(I'm making a simple EchoServer.)

>Thank you for your response.
>
>AFAP, I want to dodge from thread synch or multithreading stuffs.
>
>So I decided to use WM_COPYDATA between threads to communicate.
>
>When I use SendMessage API this works good.
>
>But when I use PostThreadMessage returns false.
>
>//posting codes here//
>BOOL bo = ::PostThreadMessage (threadid, WM_COPYDATA, NULL,
>(LPARAM)&tempStruct);
>////////////////////////////////////
>
>GetLastError show me DWORD value 1159 meaning "The message can be used
>only with synchronous operations".
>
>I've found out that when I post WM_APP or WM_USER, PostThreadMessage
>returns 0 (means success.)
>
>What is the matter with this WM_COPYDATA??
>
>Second question ...
>
>Now I'm willing to use WM_APP with LPARAM, RPARAM.
>
>But how can I load data to this message?
>
>I just want load string.
>
>Thank you in advance.
>-Ryan

Joseph M. Newcomer [MVP]
email: newcomer@flounder.com
Web: <http://www.flounder.com>
MVP Tips: http://www.flounder.com/mvp_tips.htm