

Re: Something like "ping" for windows messages.

Re: Something like "ping" for windows messages.

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2006-12/msg01788.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Wed, 13 Dec 2006 17:11:06 -0500
-

The third-party socket libraries I know of come with source (in a couple cases it's an extra-price feature). And they run on a large number of platforms; I forget which ones run on AS/400, but I know one of them does.

joe

On 13 Dec 2006 08:51:59 -0800, "Peter Smithson" <Peter_Smithson@xxxxxxxxxxxxx> wrote:

Joseph M. Newcomer wrote:

Recently a colleague was creating a mutex with CMutex. He wanted to do something and it was obvious to me from the Win32 documentation how to do it. But the CMutex layer didn't allow you to change that feature (forget what it was now - something to do with the creation of the mutex). Thin layers that simply remove functionality don't see that great to me.

How did we get from sockets to mutexes? Most experienced MFC programmers know that the whole set of wrapper classes for synchronization primitives just suck, and should be avoided. And what does a defective design of synchronization primitives relate to the socket design? They almost certainly were done by different people (and the person doing the synchronization classes seems to have been somewhat clueless).

I thought it was another example of a very simple wrapper class that adds very little but (presumably) hides features documented in the

Re: Something like "ping" for windows messages.

Win32 API. In those cases, I can't see why using the Win32 API is a bad idea.

When I teach networking, I tell my students that if they want portbilty, the correct solution is to buy a third-party socket library that is designed to be portable.

I think there were some of those on that faq website I put in my last reply. I didn't see much advantage in that though as having 3rd party code in your product is always a nightmare. Usually you don't get the source so if there's a problem you're in trouble. Some of our code could run on about 10 different versions of UNIX, Windows, AS/400 and MPE systems. Getting hold of the correct binary (64 bit/32 bit, Linux kernel versions etc.) or porting a 3rd party library was something we often had to do (ISAM libraries for example) and was often the cause a hold up beyond our control when getting a product out.

Sure you have to port your own code but that's going to be much smaller and simpler then a generic library that has many more features than you need.

Cheers.

Peter.

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