

Re: Cross Process Callbacks

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2006-08/msg00268.html>

- *From:* "Paul G. Tobey [eMVP]" <p space tobey no spam AT no instrument no spam DOT com>
 - *Date:* Fri, 21 Jul 2006 15:19:45 -0700
-

If you have the help of a wrapper API in the application space, you can use synchronization objects, like events, to handle notification of a 'callback' that it's time to run. Getting data to it is still a problem, but you could handle that with a point-to-point message queue, shared memory (memory-mapped file, without the file), etc. The wrapper API, when the application calls a function, CallMeBackWithData(Callback fcn, DWORD extraparams), say, would launch a thread:

```
thread()  
{  
  event = CreateEvent()  
  
  DeviceIoControl( driver, set up callback, event )  
  
  WaitForSingleObject( event )  
  
  // call the application's specified call-back function, grabbing data  
  // sent by the driver from wherever it belongs  
}
```

Paul T.

"Silver" <thesilverring@hotmail.com> wrote in message
news:uBJRZ3QrGHA.2052@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hi all,
I want to make a callback from a driver in the context of device.exe to an application in userland. Actually, that's a lie, I want to do it differently but I am getting pushback for other reasons.

My scenario is like this:

Application -> registers callback via stream driver IOCTL
Driver -> allocates buffers and handles reading data from device

..
..

Re: Cross Process Callbacks

Driver -> Data arrives at driver and is put in a buffer in created in device.exe context.

Driver -> Driver makes callback to userland application with the data mapped to application user space.

Application -> processes data and returns

Driver -> handles buffer cleanup and continues in device.exe context

The first problem is mapping the data buffer, MapPtrToProcess is deprecated and MapCallerPtr would not work as it is not a PSL call and I want to avoid a buffer copy. The best way I could come up with was ReadProcessMemory, this is a trusted API though so I may have problems there too.

Next, how do I make a function call in the context of a different process space? I could use a thread in the application space that simply waits for an event and do it that way (like an IST) but I am racking my brains to figure out another way – does anybody have any ideas?

BTW: This is under CE5.0, to be used on WM5.

Geoff

--