Re: custom shell starts before all services loaded

Source:
http://www.tech−archive.net/Archive/WinXP/microsoft.public.windowsxp.embedded/2004−03/1239.html

---

From: Greg Brown (*gbrown_at_iristech.com*)
Date: 03/25/04

```
Date: Thu, 25 Mar 2004 08:55:57 −0500
```

Thanks for the replies. Let me try to reiterate my problem. I apologize if this is wordy.

I have a 'basically' headless device. I say basically, because I _do_ have a user interface, but it occurs on a custom hardware display, not on a vga desktop, so it is headless in the sense that the user will never see the window's desktop. For testing, I do have a standard vga monitor connected.

I created a small app for my custom shell which then starts my main app, which is displayed on the custom display. Here's the bootup sequence. On the vga monitor, I see the empty desktop come up, then the screen switches to the 'loading person settings' screen, then my shell app apparently starts, because my main app comes up on the custom display. At that point, from the user's perspective, the system looks to be running, but he has no input available yet (through standard keyboard interface). Back on the desktop, the personal settings screen stays visible for what seems like an eternity (not sure why this is so slow), and then finally my shell app becomes visible on the desktop, and the keyboard input becomes available to the user.

So, as you can see, my goal is to delay starting the main app until keyboard input is available. Here's what I've tried so far:

I tried Slobodan's suggestion of using CMP_WaitNoPendingInstallEvents(), but it did not work. I noticed that the 'HidServ' service was not running when my shell initially started, so I tried Marco's suggestion of using QueryServiceStatus() to wait for that service to be running, but that also did not work. As far as WaitForInputIdle(), I understand that to mean that I would be waiting until my main app _can_ accept input, not when input is actually available from the system, maybe I'm wrong.

Perhaps my whole approach is misguided. Any other suggestions are welcome.

Thanks,

Greg

"KM" <konstmor@nospam.yahoo.com> wrote in message
news:uTP4GbeDEHA.712@tk2msftngp13.phx.gbl...
> *Greg,*
>
> *I did not quite get what your requirement is. Do you just want to launch*
your shell app and know when user input is available in it?
> *Then you should look at WaitForInputIdle API. You may end up launching a*
very small app as your shell. The app will run your heavy
> *shell through CreateProcess and during the launching time it may show a*
splash screen. If you call to WaitForInputIdle with the
> *handle from CreateProcess, then you know for sure when user is able to do*
anything within the shell UI.
> *More detail, " ... the calling thread can use the WaitForInputIdle*
function to wait until the new process has finished its
> *initialization and is waiting for user input with no input pending. This*
can be useful for synchronization between parent and child
> *processes, because CreateProcess returns without waiting for the new*
process to finish its initialization. For example, the creating
> *process would use WaitForInputIdle before trying to find a window*
associated with the new process.".
>
> *– –*
> *KM,*
> *BSquare Corporation*
>
> > *I found other posts similar to this on google, but no responses...*
> >
> > *My custom shell starts, but no user input is available for several*
seconds.
> > *I guess I could use EnumProcesses to wait until something starts, but*
not
> > *sure what to wait for. The only thing I saw resembling a reply was a*
> > *suggestion that the startup app just wait 'a bit' until it started the*
> > *actual gui. I was hoping for something a little more deterministic.*
> >
> > *Thanks,*
> >
> > *Greg*
>
>