

Re: Safely SHUTDOWN Device from Application

Source:

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2008-09/msg00510.html>

- *From:* "Chris Tacke, eMVP" <ctacke.at.opennetcf.dot.com>
 - *Date:* Mon, 22 Sep 2008 10:58:48 -0500
-

I wouldn't even bother doing any of this on the emulator – the hardware will implement it all differently (at least the kernel protions) so there's really little point in doing it. You could do the IPC stuff for your apps, and yes queues would probably be the route I'd use. If all of this is for a suspend, however, why all these machinations? The OS and FSD will handle all of this for you.

The only time I've ever seen something like this being remotely necessary is for power failure situations where you have some form of backup power like supercaps and you want to cleanly shut off everything before all power is lost. Even in those cases, just using a transactional FAT file system (not Microsoft's TFAT BTW – it tends to corrupt flash in this scenario, or at least it did in just about every test we ran) is usually a lot simpler.

Chris Tacke, Embedded MVP
OpenNETCF Consulting
Giving back to the embedded community
<http://community.OpenNETCF.com>

"Neo" <prideray@xxxxxxxx> wrote in message
<news:ef67e4e5-2c0b-4c54-b262-8e8f019ef749@xx>
On Sep 22, 6:03 pm, "Chris Tacke, eMVP" <ctacke.at.opennetcf.dot.com>
wrote:

If you have only specific applications then you could designate one to be the app that actually invokes sleep, but you'd still need IPC to have all of the other apps tell you when they're ready to shut down, and again, you'll likely need some form of timeout for them as well in case they don't report in for any reason.

How you handle suspend is very much hardware dependent, so I can't really

Re: Safely SHUTDOWN Device from Application

tell you where it is. If you have a power button, it likely generates some form of interrupt at the processor or maybe a CPLD or whatever. If you have suspend currently working from that button then you need to look at the handler for that interrupt and *that* is where you need to intercept things.

--

Chris Tacke, Embedded MVP
OpenNETCF Consulting
Giving back to the embedded community <http://community.OpenNETCF.com>

"Neo" <pride...@xxxxxxxx> wrote in message

news:848a4bfe-0159-489d-bd58-2e6324ed8edb@xx
On Sep 19, 10:26 pm, "Chris Tacke, eMVP" <ctacke.at.opennetcf.dot.com> wrote:

That's going to be very device-specific. The handler for device suspend needs to be modified to send out the notification. An example of how this could be done is the way Eurotech does it (not saying it's the best way, just *a* way). You can set a registry option to have the power button interrupt just set a system event. It's then up to the app to take whatever actions it needs and then call the API to actually suspend.

The problem in your case is the arbitrary desire for "any" app to be notified and given time. How will they be notified (an even is simple enough)? More important is how will the OS/kernel know when all of those apps are "done" and are in a safe state? Would they send an event back? How would the OS know who all should reply back? What if the app is hung and not responding? There are a *lot* of variables to this – and I can't see any general solution.

--

Chris Tacke, Embedded MVP
OpenNETCF Consulting
Giving back to the embedded community <http://community.OpenNETCF.com>

Re: Safely SHUTDOWN Device from Application

Thanks,
Neo

Thanks for the reply Chris. What if there are only specific applications of mine that might deal with the flash and no other user applications apart from these are going to run on the device? In that case, is there a mechanism by which we can notify these applications about the system going for a shutdown? Let's not consider the case where one of the applications is not responding. By "The handler for device suspend" – you mean the routine OEMPowerOff(). Am I correct? If yes, what kind of changes I might require to do?

Warm Regards,
Neo

You mean, the app that we'll write to carry out the shutdown process will invoke sleep after notifying all the desired (known) applications about the shutdown using an IPC? (Hence giving them a chance to safely exit). I did not know prior to this discussion what were the means to do IPC on WinCE devices. After few searches I find that: Message Queues, Services.exe and Shared Memory via File Mapping are amongst the better ways to do that. (Ref: <http://blogs.msdn.com/cenet/archive/2005/07/13/438424.aspx>). What mechanism might be appropriate in our case? I think Message Queues would be both appropriate and relatively easy for a newbie like me to use? Any comments?

As of now, I don't have the hardware with me. I think I'll need to find the IOCTL for the Suspend (of the emulator) and find a mapping of that IOCTL to a routine in the Source, which I can modify. If this approach is correct, can I use the Call Profiler to know these things? Or is there a better way to find out this info?

Thanks for all the help,
Neo

.