Re: What time is it? Issues with Local time, system time, DST

Source:
http://www.tech−archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2009−04/msg00206.html

- *From*: "Paul G. Tobey [eMVP]" <p space tobey no spam AT no instrument no spam DOT com>
- *Date*: Wed, 15 Apr 2009 10:45:48 −0700

> Sorry, I mistyped. I meant that OEMGetRealTime returns a 32 or 64 bit value. That wasn't clear either, since it's a SYSTEMTIME, which can be converted to a flat value (using functions available in kernel or applications).

No, OEMGetRealTime does NOT return anything time−related. It returns TRUE or FALSE.

> We have a headless device that runs from some utility closet and needs to know what time it is when it reboots without user intervention. It works with a server that could be in a different time zone, and so many time configuration and calculation are done in GMT. Also, local time sometimes has certain limitations. If you want to subtract two time values, you cannot do it directly in local time, unless you know there was no DST transition between them.

I would still suggest that you NOT try to shoehorn UTC into what Windows CE is expecting to be local time. If you want your device to run with local time = UTC, that's fine; just set the time zone appropriately. Otherwise, use a service, like the time service that Microsoft provides, to sync with your external server or simply convert your local time into UTC before using SystemTimeToFileTime() and doing your math on it.

> It seems like determining current time on startup would be much less of a headache, and would require reference to one piece of hardware (RTC) not the coordination of two (RTC, flash). Our overall board was implemented years ago and we had trouble getting registry persistence to work well. I we were doing it over we probably would persist registry directly in flash and save some of the headache.

It's no easier to read time referenced to one time zone from the RTC than it is to read time referenced to any other time zone. It sounds as though

you're trying to make a design that is poorly implemented somehow "better" by misusing the API in Windows CE. That's not a sensible thing to do. You still have to have the same information, regardless of whether the hardware clock is keeping local or UTC time: DST state and time zone. No reason to reverse how Windows CE expects it to work that I can see. If you need to sync time with a server on startup, you can do that with the Time Service that Microsoft provides.

> It's really a question of how you figure out what time it as the moment you boot. It sounds like generally you should persist timezone and DST state (either through registry or otherwise), and use those to adjust RTC if necessary on startup.

Yes, since that's how the entire API works. You can do anything, of course, given enough time. I've rewritten the code in several versions of Windows CE to properly handle the DST transition occurring when power is off, for example. It works and, for time clock applications, it matters. However, in your case, I see no compelling reason to do anything other than persist the registry and use the battery−backed RTC to keep time while the unit reboots.

Paul T.

.