

Because programmers were trusted to do the right thing, part 2

devblogs.microsoft.com/oldnewthing/20060216-06

February 16, 2006



Raymond Chen

Some time ago, I discussed briefly the philosophy of API design that prevailed in the early days. One of the places this manifested itself was in the area of power management. As originally designed, power management was a cooperative affair (as was nearly everything in Windows in the early days). When the user attempted to put the computer into a low power state, the system sent a `WM_POWER` message to each program, allowing it to prepare for the suspend operation, say, closing network resources or flushing caches, or even reject the suspend attempt outright. Programmers were trusted to do the right thing. Rejecting the suspend attempt should be done based on user confirmation. For example, a program might put up the message, “You have unsaved changes to the database record. Save the changes before suspending?” with the replies Yes, No or Cancel. Selecting “Yes” would save the changes, selecting “No” would leave the changes unsaved, and selecting “Cancel” would cancel the suspend operation. Programs were discouraged from rejecting the suspend operation without receiving confirmation from the user, although this was not enforced because a program might have a legitimate need to fail the suspend silently. What would that legitimate need be? Who knows, but maybe there is one out there, and Windows was being flexible enough to allow for that possibility. As Huat Chye Lim pointed out over on the Windows Mobile PC Team Blog, the result of this was that everybody considered themselves to be one of these “exceptional” cases that didn’t need to get user confirmation before rejecting a suspend request. Ultimately, the result was that for many users, the suspend command simply never worked. Programmers were trusted to do the right thing, and they abused that trust.

Those who were at the 2005 PDC and attended FUN319 Windows Vista: Developing Power-Aware Applications learned that Windows Vista is addressing the issue by simply not trusting programs any more. Programs will no longer be able to reject suspend requests or even to delay the suspend for more than a few seconds. That’s what happens when you don’t play friendly: The person with the ball simply doesn’t let you play any more.

Raymond Chen

Follow

