Hasn’t the problem of updates being partially installed until the next reboot already been solved by changes in Windows?

devblogs.microsoft.com/oldnewthing/20151211-00

December 11, 2015

Raymond Chen

Last week, I discussed how A question about how to detect whether Windows Update needs the system to be restarted turns out to be the wrong question. One of the issues I noted was the risk of the partially-installed updated. A number of commenters wanted to know if that problem was still true. My colleague Mark Phaedrus discussed the matter in comments, but I’m going to promote his responses to a full posting, thereby extending his Internet fame from 15 minutes to 30 minutes.

Let me answer a question that several folks have raised (both in the comments and offline): Hasn't the problem of updates being partially installed until the next reboot already been solved by changes in Windows?

This is, to a large extent, true. Modern versions of Windows use Component-Based Servicing (CBS). This technology makes sure that new Windows components, and new versions of existing components, are installed atomically. In other words, if it is possible to install or update a component without a reboot, CBS does so. If it is not possible (because one or more files are in use, or because the component requires more complicated setup), then the entire installation of the component is automatically suspended until the next reboot.

So this means that the problem described in this blog post is gone, right? Absolutely not, for at least two reasons.

First, not all updates distributed through Windows Update/Microsoft Update are purely CBS-based. There are a variety of different types of updates (drivers, Office updates, etc.), each of which may have different installation behaviors. For example, there are still a few troublesome drivers that do not behave normally until the next reboot. And from the Windows Update perspective, there is a class of updates called "command-line updates" — updates that have unusual needs, and so cannot be published in the usual standardized formats. Command-line updates can still work in whatever way they want, just like the good old days of UPDATE.EXE. And that means that command-line updates may still be subject to the problem.

To summarize: Most updates no longer create an unusual machine state that requires a reboot to resolve. There are still a few that do. In an ordinary consumer environment, the remaining problem is small enough to be ignored (or at least small enough that there are lots of other things to concentrate on fixing first). But in an environment where The Machine Simply Must Work, it's still an unacceptable risk. And so the best practice for these environments is still assuming that any update that requires a reboot to complete should have that reboot performed as soon as possible.

Second, even setting the "Does the update do the right thing before the reboot?" problem aside, CBS itself creates another problem in this scenario. Since many Windows updates wind up getting their processing delayed until the reboot, that reboot itself can take longer (since all those pending operations then get performed). And in an environment where The Machine Simply Must Work, this means that the consequences of an accidental/unplanned reboot can be even worse. So again, in this environment, it's important to ensure that Windows Update never initiates a reboot on its own. And since Windows Update will sometimes initiate reboots on its own when it's set to install updates automatically, this means that the best practice for these environments is still the practice described in the blog: Set Automatic Updates to not install updates automatically, and use your own code to install updates and reboot at the correct times and with the proper user notification.

> To reiterate earlier caveats, when I talk about situations where The Machine Simply Must Work, I naturally presume you're not talking about life-critical medical applications, because Windows is not for life-critical applications, as the esteemed attorneys who hand-crafted the Windows EULA from artisanal Unicode characters will happily point out.
>
> To give my own favorite example of a non-medical situation where The Machine Simply Must Work, as well as one of the more uncomfortable moments of my Microsoft career: There was an internal developer conference going on at the Microsoft Executive Briefing Center, a very nice building filled with conference rooms normally used by the grand poohbahs of the Microsoft organization, and occasionally used for gatherings like this one. Throughout the building there are large displays helpfully pointing out which meetings are going on in which rooms at which times. And on this occasion, with numerous grand poohbahs on hand as well as large portions of the Windows Update team, all those displays were showing the old "Your machine needs to reboot" prompt we all knew and loved from Windows 7, with the pre-reboot countdown timer inexorably rolling down towards zero.
>
> If the role of an internal developer conference is to encourage discussions among teams, then that one certainly succeeded. Because discussions most definitely ensued.

Raymond Chen

Follow