A question about how to detect whether Windows Update needs the system to be restarted turns out to be the wrong question

devblogs.microsoft.com/oldnewthing/20151203-00

December 3, 2015

Raymond Chen

A customer wanted to know whether there was a way to detect that Windows Update has recently installed an update that requires a reboot. They explained that their computers are running dedicated software, and the customers are not technically-savvy, so they want to display the "You need to reboot" message more prominently, and let the user choose the best time to reboot. "We have been checking a variety of registry keys, and we have found that the set of keys to check is quite extensive and varies depending on the nature of the update that was installed."

There is a fairly straightforward way to check whether Windows Update is waiting for the system to reboot, but we asked for more information about the customer's scenario. Fortunately, the customer was willing to share.

> These computers are in a hospital running critical medical software. The software runs 24/7 and must not reboot without user consent.

If these are critical machines, then asking whether Windows Update is requesting a reboot is the wrong question. By the time you get the answer "Yes", you are already in trouble: If you reboot the computer as soon as a reboot is required, you interrupt the machine's critical functioning. If you postpone the reboot, then you leave the computer in a nonstandard configuration wherein an update is partially-installed, and that nonstandard configuration may compromise the machine's critical function.

In this specific case, the idea would be to change the design from "Install the update, and then postpone the reboot until a convenient time" to "Wait for a convenient time, then install the update and reboot immediately." Here's a sketch of how it could work. (Note: This is a sketch, not a fully-analyzed scenario. Do not implement this design on your critical medical systems before independently validating its efficacy to your satisfaction. In fact, as a general rule, you should not use blog posts as the sole basis for software design decisions for critical medical systems.)

- Set Automatic Updates so it does not install updates immediately. Either disable it, or set it to download updates without installing them.
- Ensure that users do not have permission to alter these settings or to initiate the installation of updates independently.
- Periodically check whether there are new updates available. The Searching, Downloading, and Installing Updates script on MSDN is an example of how you can use the Windows Update Agent API to answer questions like this.
- If there are updates available that do not require a reboot, you may choose to install them without interrupting service. You can use the `InstallationBehavior.RebootBehavior` property to determine whether a particular update guarantees that it can be installed without a reboot.
- If there are updates available that may require a reboot, inform the user that the computer will be unavailable while the updates are installed and let the user choose when to install those updates.
- When installing the updates, display a message indicating that the computer is unavailable.
- When installation is complete, reboot immediately.

The important things are that (1) you treat the installation of the update and the reboot as a unit, and (2) you don't start the install+reboot process until the user confirms that they are okay with the system being temporarily unavailable.

Thanks to my colleague Mark Phaedrus for providing the raw materials from which this article was constructed.

Raymond Chen

Follow