# The Old New Thing

## Generally speaking, yanking the power plug unexpectedly should not be part of your business process

13 Aug 2013 7:00 AM | 25

A customer had a complex process for setting up their computers, and the process recorded information in the registry so that applications could record their state across reboots. They then noticed that if they yanked the power cord instead of going through the normal Shutdown process, that the registry keys were not reliably updated. They were wondering if there was a function they can call to force the registry to be flushed to disk even if the system doesn't go through a normal shutdown.

Patient: "Doctor, it hurts when I do this."

Doctor: "Don't do that."

You could call the `RegFlushKey` function each time you update the registry key, but you'll be flushing your performance down the drain.

And calling `RegFlushKey` doesn't solve the "unexpected power loss" problem entirely. If power is lost *while the key is being flushed*, then you can end up with internal registry corruption. Not to mention that cutting power will corrupt the hard drive due to unflushed data both in the operating system disk cache as well as the hard drive's on-board disk cache.

It's like somebody who says, "We never save our documents; we just let Excel AutoSave find the document each time we startup Excel. We found that if we yank the power cord to turn off the computer, sometimes when we boot the computer back up, the Excel document we were working on doesn't show up in the AutoSave recovery dialog. How can we force Excel to AutoSave our document before we yank the power cord?"

Dude, your problem isn't a configuration problem with AutoSave. Your problem is that *you're yanking the power cord as part of your business process*.

The customer reported back that, thankfully, killing power was not part of their normal procedures. Rather, the issue with unexpected power loss was something they discovered by accident.

Sigh of relief.

**Blog - Comment List MSDN TechNet**

### Comments

**Someone**
13 Aug 2013 7:19 AM
#

"If power is lost while the key is being flushed, then you can end up with internal registry corruption."

I really hope that this is not true. The registry must behave like a real transactional

database by being absolute robust against loss of power at any time. After reboot, the OS must recover the registry to a consistent state.

**davep**
13 Aug 2013 7:33 AM
#

"They then noticed that if they yanked the power cord instead of going through the normal Shutdown process, that the registry keys were not reliably updated."

Not an example of a "good hire".

**alegr1**
13 Aug 2013 7:45 AM
#

@Someone:

The registry consistency is kept by a transactional log, starting I guess with Vista/Longhorn.

**jader3rd**
13 Aug 2013 8:08 AM
#

Yanking the power cord is part of my regular business process in testing our High Availability and recovery solution. If you're not willing to pull the power cord on every piece in your system (not necessarily all that once), you don't have a good solution.

**thomas**
13 Aug 2013 8:42 AM
#

Any reason they did not use battery backed UPSes to keep the computers up long enough to shutdown gracefully?

**Jessica**
13 Aug 2013 8:46 AM
#

@jader3rd,

I write software for Windows-based always-on industrial controllers that are only shut down by removing power, so the same concept applies. However, the customer Raymond wrote about was using their software for setting up their computer systems. There's no

reason that a one-time setup procedure on a system they control should have to be able to withstand a power-cycle at any arbitrary part of the process.

**Parrotlover77**
13 Aug 2013 10:13 AM
#

"The customer reported back that, thankfully, killing power was not part of their normal procedures. Rather, the issue with unexpected power loss was something they discovered by accident."

Never let facts get in the way of a good rant against a customer of yours. Also, if the registry cannot tolerate a power outage during flush, Windows is in serious trouble (not everybody understands why that might be bad and even less have a UPS for their desktop). Thankfully it is actually quite tolerant.

And 1000 times yes to what jader3rd said.

**Danny**
13 Aug 2013 10:15 AM
#

If you have a watchdog in place to avoid hang-ups then that is a very close case to the power loss scenario. Think embedded systems, their base OS always have watch dog and if something goes wrong and your application hangs the watch dog will restart without waiting for the underlying OS to do any flushing. So part of the robustness test for these is also to make sure you always have 3 steps. One is memory cache, two is disk cache, three is marked as OK. So you mark the "file" as disk cache and start to put from memory to disk. Then last step is to mark your disk cache as OK. Anything happens before marking as OK, you dismiss it after restart even the data is just fine inside but the restart came in during OK marking. Real life example? Your car, unless you still drive a ancient one, all created after 20 years ago got embedded systems in them. Oh, you talk Windows world only? Fine, then do you talk on a Lumia?

**V-SHorn**
13 Aug 2013 10:24 AM
#

@Jessica: I can see how Raymond's description can be read to mean that the only time the customer's registry entries were changed was during system setup, but I suspect that if this became a problem for them due to an unexpected power outage, then the customer was updating the registry continually. If true, the timing of registry flushes and the volatility of the registry would be highly relevant pieces of information. I would agree with anyone who thinks that sounds like registry abuse, and I don't know why they would need to solve any problem that way.

**Nicholas**

13 Aug 2013 11:54 AM
#

> Windows-based always-on industrial controllers that are only shut down by removing power

I don't understand this. Unless you have some kind of special persistent storage device that's designed with "pulling the plug" in mind, how can such a system be even remotely plausible? As Raymond describes (and this is just one example), Windows is absolutely not created with unexpected power losses in mind. Is there some configuration you can come up with that makes such a system work, or does it just work "well enough" that nobody has bothered to find something else?

**Jessica**
13 Aug 2013 12:18 PM
#

@Nicholas

In the software, writes are all done using Transactional NTFS, and state information is saved out in scheduled batches. It's okay if a power failure wipes out data since the last batch, as long as everything is consistent when it boots back up. The power supply circuitry brings down the hard drive power more slowly than the motherboard, allowing us to disable the write-cache buffer that Raymond has written about in the past.

When validating hardware changes we run the system through a very aggressive power-cycling test. It usually makes any system unbootable eventually, but if they survive long enough then we know that failures due to file system corruption will fall far enough down on the list of returns that it isn't an issue. Since switching to SSDs we haven't been able to kill a system at all with the power cycler, but it hasn't been long enough to see how that translates into real-world failures.

**Matteo Italia**
13 Aug 2013 3:03 PM
#

@Jessica: uhm, you may not kill SSDs with power cycling, but from what I hear they have quite a tendency to commit suicide without doing anything particular to them.

Probably the last generations got better (and fortunately, after a year my Vertex 3 is still up and running), but consumer-grade SSDs do show high failure rates (see e.g. here: hexus.net/.../44937-new-reliability-poll-shows-surprising-ssd-failure-rate); to quote Jeff Atwood, «Solid state hard drives fail. A lot. And not just any fail. I'm talking about catastrophic, oh-my-God-what-just-happened-to-all-my-data instant gigafail. It's not pretty.» (www.codinghorror.com/.../the-hot-crazy-solid-state-drive-scale.html)

**Nicholas**
13 Aug 2013 4:13 PM
#

@Jessica

Interesting. I suppose that you might be able to make an application robust against power loss, but I'm still wondering about the operating system. Windows may be in the middle of writing to the Registry, or performing some group of operations and when suddenly interrupted things don't go very well. In fact, since you say that your power-cycling tests eventually render a system unbootable, it looks like you're really just playing a game of chance every time you turn off your systems. Eventually the dice come up snake eyes and, hopefully, it doesn't happen at the wrong time. Even if your application survives okay you'll still be rebuilding/reimaging the systems.

Regarding disabling of the write-cache, do you have to use special or specific hard drives in these systems? So many (read: all) of them do their own magic buffering and aren't built to survive power losses. And then you hope you don't have any driver I/O buffering getting in the way... Even using TxF won't help with this.

I'm not disparaging you or your systems :) I'm just always curious when I see Windows being used in places it just doesn't seem well-suited for.

**Maurits [MSFT]**
13 Aug 2013 5:03 PM
#

> yanking the power plug unexpectedly should not be part of your business process

Of course not. Since it's part of the process, it is by definition expected.

**Roger**
13 Aug 2013 6:26 PM
#

You mean every time we do an office move here on campus (like 3 times a year) those guys log into my system and do a clean shutdown!? I'm not sure what I'm more afraid of...

**Jon**
13 Aug 2013 6:52 PM
#

@Nicholas

Nobody said that she was using regular Windows. Windows Embedded Standard (which is the "normal" Windows, not CE) has a number of features for exactly their use case. One is the Write Filter. This allows Windows to run on a base, write-protected image. Another is Hibernate Once Resume Many. This means every time Windows starts up, it is effectively reimaged.

**hagenp**

14 Aug 2013 1:19 AM
#

Still it is time for Windows to incorporate a "journaling file system".

These have been around for decades by now, and with larger HDD sizes it is overdue to have some robust kind of mass storage system.

**Someone**
14 Aug 2013 3:16 AM
#

@hagenp: Are you serious? It's there since the first version of Windows NT. Its called NTFS.

**Jessica**
14 Aug 2013 4:46 AM
#

Actually, we are using WES7, but HORM doesn't work with our application. We're aware that there is a risk every time it is shut off, but that's why we do the testing. Like I said before, if it makes it long enough on the test then we know that the actualy failures in the field will be very rare - physical hard drive failures and other hardware problems (or a field service person replacing a perfectly good unit as part of a shotgun approach to troubleshooting) are much more likely based on actual return data. I've talked to the disk manufacturer about our use case, and they claim the cache should successfully flush.

It will still be a while till we start seeing how SSDs do in the field, but so far we haven't had any come back for drive problems. We're expecting them to perform more reliably in this environment - there is generally a decent amount of low-level vibration that can only be damped so much.

These things have been running Windows since before I started working on them, and there isn't much I can do to change that any time soon, but at least I've been able to make them significantly more reliable with the changes I've been discussing. The biggest argument for Windows is that a customer can plug in a USB printer and it just works. If not, they can run Windows Update or the Add Printer wizard. Printer support is there under Linux, but making it user-friendly in a way people who don't use a computer at home (and don't really understand that the controller is a computer) can handle is a much bigger task.

**Brian_EE**
14 Aug 2013 6:26 AM
#

@Jessica: I seem to get the impression (from reading this blog the last couple years) that most of the readers/commenters don't understand embedded systems very well, which is fine. We also have design verification groups that submit our equipment to a wide range of tests. Our systems typically run on QNX, and we find the highest failure rate of our products in the field is due to enemy gunfire spraying shrapnel and bullets

through our circuit boards.

**Brad Westness**
14 Aug 2013 6:28 AM
#

@Roger: Or the just push the physical power button?

@jader3rd: However, this sounds like it's a one-time set up process that basically doesn't complete correctly if the power is yanked in the middle. It's a bit like complaining that Windows Setup doesn't complete correctly if your power goes out midway through. Or that a firmware update for your phone doesn't complete correctly if you unplug it and take out the battery midway through.

**Jessica**
14 Aug 2013 7:25 AM
#

@Brian_EE,

Sounds like the sort of problem they would try to get me to write a software patch to fix. :-) Luckily our equipment actually tends to destroy guns (destructive testing) rather than the other way around.

**Marcel**
14 Aug 2013 7:32 AM
#

We use Windows in PC based QA systems and I've been told that they are all just shut down hard by the mains switch. IT apparently disabled the hard drive write caches of the machines, but apart from that it's standard Windows XP fare (soon Windows 7). Never heard of any problems, but it might get interesting again now that out industrial PCs come equipped with SSDs by default...

**Killer{R}**
16 Aug 2013 2:08 PM
#

its a bit offtopic, but while reading your imaginary excel 'saving' I remembered bash's joke about smbd's mother saved documents by Power button. But unlike article's example this way works perfectly. Excel asks to save document before computer shutdowns (thanks ATX, yep).

**Joshua**
17 Aug 2013 7:51 AM

#

The question has another important case; that is calling RegFlushKey will get you out of trouble if you are using the registry as part of a distributed transactional system (distributed does not necessarily mean across computers here but only across different API sets that have different definitions of transaction).