Windows 10 IoT Enterprise: Unified Write Filter and Boot Time Architecture Trade-Offs

By Sean D. Liming and John R. Malin
Annabooks – www.annabooks.com

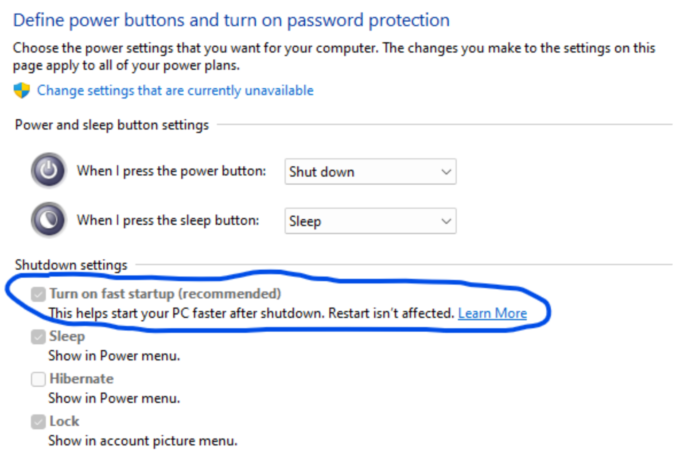
January 2024

Recently, two developers commented that Windows 10 IoT Enterprise LTSC 2021 was booting slower than previous editions of Windows 10 IoT Enterprise. They also report that when UWF is not enabled in the system, the system boots faster. Normally, if one person reports an issue, it could be an anomaly or design issue, but with two developers reporting a noticeable increase in boot time, a little investigation is needed. As stated in the book, *Starter Guide for Windows® 10 IoT Enterprise 2nd Edition*, hardware plays a big role in the performance of the system; but software has an impact on performance as well. In the case of Windows, design choices can impact the boot speed of the device. In this paper, we will discuss the impact of Unified Write Filter (UWF) on boot speed.

UWF versus Fast Startup

The write filter technology has changed since Windows NT Embedded, but the desire to protect the C partition from disk corruption or unwanted writes to the disk remains the same. Most OEMs take advantage of UWF to protect the system and create high a high-availability system. When UWF is enabled, features like a page file, disk cache, and hibernation are disabled. Disabling hibernation has introduced a subtle boot timing issue that has come to light with the release of Windows 10 IoT Enterprise LTSC 2021.

Microsoft has been working to improve the boot time of Windows. Fast startup (aka fast boot) was introduced to speed up the boot time from a cold start. Of course, there is the resume from sleep, but this is a warm boot time solution. For a cold start, the normal PC user has no idea that fast startup is already enabled. All the user sees is the system booting within seconds compared to old Windows versions of yesteryear.


Define power buttons and turn on password protection

Choose the power settings that you want for your computer. The changes you make to the settings on this page apply to all of your power plans.

Change settings that are currently unavailable

Power and sleep button settings

When I press the power button: Shut down

When I press the sleep button: Sleep

Shutdown settings

[✓] Turn on fast startup (recommended)
This helps start your PC faster after shutdown. Restart isn't affected. Learn More

[✓] Sleep
Show in Power menu.

[ ] Hibernate
Show in Power menu.

[✓] Lock
Show in account picture menu.

Fast startup requires hibernation to be enabled as it takes advantage of the hiberfil.sys file. Fast startup also requires a safe shutdown, since information is being stored in the hiberfil.sys on shutdown. With UWF enabled, hibernation is disabled, which means fast startup is also disabled. When UWF is enabled, systems go through the full Windows boot process, which can take some time compared to fast startup.

Most OEMs allow their systems to be powered off without performing a safe shutdown. For example, pull the power plug or just kill the power. UWF is there to help mitigate disk corruption for a sudden power loss. Fast startup will not work if Windows is not safely shut down, since nothing is saved to the hiberfile.sys file. Fast startup is not a possible solution for many OEMs.

We know UWF will force a longer boot-up sequence since fast startup is disabled, which is a big clue to what the clients were reporting.

Test Configuration

Now, we turn to testing the boot speed for the different Windows 10 releases. To test the boot speed, an older platform was chosen to test three Windows 10 releases and the preview of Windows 11 IoT Enterprise.

Test Platform: Up2 Board, Intel N3350, 4GB RAM, 256mSATA

Windows Releases Under Test: Windows 10 IoT Enterprise LTSC 2016, Windows 10 IoT Enterprise LTSC 2019, Windows 10 IoT Enterprise LTSC 2021, and Windows 11 IoT Enterprise LTSC 2024 (preview). All versions are 64-bit, and installed for UEFI boot. All versions were configured the same way using the same device drivers. To disable fast boot, hibernation was disabled and UWF was enabled, which is a typical scenario for many systems.

Test Measurement: The boot time from power-on to the Explorer Shell appearing was measured. Three test boots were made to address human error. The average time was calculated. Two sets of tests for each Windows Release were made. The first was with UWF enabled, and the second was UWF disabled and fast startup enabled. For fast startup to function successfully, the system was safely shut down before the boot measurement was made.

Boot Time Results

The timed results for both sets of tests are as follows:

UWF Enabled/ Fast Startup Disabled Results:

Windows IoT Release	Average Boot Time (seconds)
Windows 10 IoT Enterprise LTSC 2016	41.34
Windows 10 IoT Enterprise LTSC 2019	40.81
Windows 10 IoT Enterprise LTSC 2021	49.22
Windows 11 IoT Enterprise LTSC 2024 (Preview)	51.74

LTSC 2016 and LTSC 2019 had similar boot times. There was a big jump from LTSC 2019 and LTSC 2021.

UWF Disabled / Fast Startup Enabled Results:

Windows IoT Release	Average Boot Time (seconds)
Windows 10 IoT Enterprise LTSC 2016	23.89
Windows 10 IoT Enterprise LTSC 2019	26.09
Windows 10 IoT Enterprise LTSC 2021	28.07
Windows 11 IoT Enterprise LTSC 2024 (Preview)	31.41

Even the fast startup time has increased with each new release, but it is still much faster than a full boot.

Important: Different hardware configurations, different Windows features, and other 3rd party software can cause an increase in the boot time, which can go up exponentially. Your results may be very different, but there is no escaping the fact that LTSC 2021's boot time will be greater.

What is causing the boot time increase?

Without breaking out the low-level debug tools and digging into source code we don't have, we can only take an educated guess on what is causing the boot time to increase. The working theory is that the improvements in the secure boot process have increased the boot time. Increased security is a positive for the system, but coming at the cost of an increase in boot time is a negative, which means there is a trade-off between UWF protection versus boot speed.

What about UWF / Hibernate Once Resume Many?

Is it possible to boot faster with UWF enabled? Yes, it is. UWF has a feature called Hibernate Once Resume Many (HORM). The idea is to boot from the hiberfil.sys each and every time, which is faster than a cold boot. HORM has been a disregarded feature because of the complexity of setting up HORM. Once set up correctly, the system boots (or resumes from hibernation) within seconds.

Note: In older versions of Windows Embedded, it was observed that resuming from hibernation was slower or equal to a cold boot. Windows 10 has improved on hibernation resume speed.

Before we go further, there is a little quirk with hibernation that has been lightly documented. Let's say you are working on your laptop with a USB flash drive plugged in. You hibernate the laptop and then pull the USB flash disk out of the system. You decide to put the USB flash drive into another system and copy over some files to use on this laptop. You go back to the laptop, plug in the USB flash disk, and power on the laptop which resumes from hibernation. You go to check the new files on the USB flash disk and the files are gone. This is because the file table from the USB flash disk was stored in the hiberfile.sys, and the file table gets restored when the system resumes. The new files you added in the other system are not in the old file table, thus they are lost.

Most OEMs set up the write filter to protect drive C, but they have a second partition that is left unprotected to store data. For EWF, there was a solution to dismount the secondary partition before hibernation, and after hibernation has resumed, remount the secondary partition. This older article covers the solution: Dismounting Volumes in a Hibernate Once/Resume Many Configuration | Microsoft Learn. The dismount/mount solution got around the little quirk of the hiberfil.sys and the file tables.

Here is where HORM fails to be a solution for UWF. The old Enhanced Write Filter (EWF) was the first to support HORM. EWF protected the whole partition with no file write-throughs, which is needed for hibernation resume to work per the file table saving already discussed. UWF allows for write-throughs. For HORM to be enabled for UWF, UWF cannot have any write-throughs and all volumes must be configured for ROM mode, which is new for UWF. Notice, the word "volume" and not partition is used. UWF was designed for sector-level protection, and EWF was designed for file-level protection. The difference in write-filter architecture comes into play. For UWF, all volumes must be protected to enable HORM. Even if the secondary partition is dismounted, the volume remains with no drive letter. Uwfmgr.exe still sees the volume in the system and will not enable HORM unless all volumes are protected regardless of whether the volume is unmounted. If HORM was a little talked-about feature before, HORM is now a completely useless feature.

Note: It is possible to plug in a USB flash disk or other disk after hibernation, and have that as a data disk. In manufacturing, one would have to initiate HORM on each system and then add the disk. Again, this is not the ideal solution.

Why UWF/HORM is forcing all volumes to be protected is unknown. If the partition/volume is dismounted, that partition's file table is not present when hibernation is initiated. It might be that the

UEFI boot partitions have to be protected, thus someone was lazy and made it so all volumes have to be protected. Only Microsoft can investigate further.

Summary

Many OEMs prefer to support their platforms as long as they can, but each Windows release keeps advancing security and new technology that impacts the performance of older systems. In this case, the boot time has increased increased over the previous versions. The investigation results confirmed the boot time increase when moving to Windows10 IoT Enterprise LTSC 2021. UWF and boot time bring another architecture point that needs to be considered when creating a custom Windows IoT image.