presented by

Microsoft

Engineering UEFI Firmware for Windows: Best Practices and Pitfalls to Avoid

UEFI Summerfest – July 15-19, 2013
Presented by Emily Wilson (Microsoft Corp.)

Updated 2011-06-01

Agenda

- Introduction
- Common issues we've seen
- Tests and tools to use on a regular basis
- Questions

Why it matters

Many new experiences in Windows are powered by UEFI.

This connection gives the user to have a whole experience, without visible seams between hardware, firmware and OS.

Seemingly small issues can translate into the system being:

- Not-bootable
- Un-usable
- Not easily recoverable in the field

Boot to devices or firmware

Windows “use a device” menu shows the list of items in UEFI load options.

Windows “boot to firmware” only available if OSIndications variable is used.

Issues we’ve seen

- Devices that have USB devices that come and go, based on last boot configuration
- Options that are not valid
- Have names that are not understandable to users
- USB class entry is renamed for a specific device (that is no longer present)
- Missing boot to firmware options

Expected behavior

- *Include a USB Class device entry*
  - This should be included regardless of whether or not a USB device was plugged in on the last boot, or is even currently plugged in
- Prune options that are not valid
- Ensure description strings are correct– end users are reading those!
- Ensure and test support for OSIndications

Boot graphics

The boot graphic is part of the first experience the user has with their device.

In some cases, the system is locked to the resolution given by the GOP driver at boot.

Issues we’ve seen

- Come up with an aspect ratio that doesn’t even match the screen!
- Boot with an extremely low resolution
- Lose their logo during OS re-install
- Have logos placed in the wrong location, extra logos, or text

Be sure your implementation is

- Coming up at native resolution
- Continues to follow guidance around use of the BGRT table
  - Ensure that the valid bit is being used only as an indication of whether there is currently a valid logo on the screen
- Meets the Windows guidance for POST screen layout

USB Enumeration

A frequent optimization in order to reduce POST time is to avoid full USB enumeration.

It is not a safe optimization if the system is not booting directly to Windows.

Full enumeration may be skipped except:

- If there are other options higher than Windows in the boot order
- A boot next variable is set, causing a different boot device to be used
  - That boot device might be USB, or might require the use of a USB keyboard
- Failures occur on the immediately preceding boots

Hardware Changes

Changes to hardware configurations across power transitions need to be handled with care.

The fast startup transition in Windows 8 is an S4 transition.

Recommendations

- EFI GetMemoryMap must be identical from the last cold boot to any resume from Hibernate, including fast boot
- Indications for hardware/firmware changes across a fast boot transition should be indicated by changing the Hardware Signature in the FACS table

Variable services

Has been a frequent source of bugs.

Some tests passing with less than a 50% success rate.

Issues we've seen

- Corruptions because runtime variables were allowed to be created with the same name as boot variables
- UEFI variables not being protected across a firmware update
- A spectrum of issues in Private Authenticated Variables

Recommendations

- Pick up fixes from Tianocore frequently
- Run HCK and other UEFI tests available from Microsoft early and often in your development cycle
- Swing by our testing suite here at the plugfest!

Boot Order

Chain loading done incorrectly is creates fragile systems breaking OS servicing.

Issue we've seen

- UEFI applications needing to run prior to calling the Windows Boot Manager load option. Rather than using BootOrder, the applications choose to modify files in an existing option instead.

Recommendations

- Use the load options list as specified in the UEFI spec
- Place any needed UEFI drivers or apps in the appropriate folder so that they are loaded and used appropriately
- Test and be sure to understand what happens in cases where BootOrder may be changed

Quick but important items

DHCP Protocols should be enabled where available

- No effect on POST time
  - Should not actually be enumerated unless network boot is being used on that boot
- Currently required in HCK tests for systems with wired LAN

Enumerating and reading variables

- Performance issues observed reading and enumerating UEFI variables
- Problematic especially in cases when all variables are being enumerated

Recommended tests

Try these

- HCK UEFI Tests
  - Npctest.exe Use command line “npctest.exe envvar”
  - UefiSecureBootLogo.dll in the WHCK.
    “te /name:*variable* UefiSecureBootLogo.dll”
    “te /name:*VerifyBootServices* UefiSecureBootLogo.dll”.
- Run Power test to exercise the reliability of entering and exiting applicable power transitions
- Run “bcdedit /enum all”

There are some tests that don’t map to a requirement that you may find *very* useful!

We’re happy to help you run any of these this week at the event.

Thanks for attending the UEFI Summerfest 2013

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

Microsoft