Leveraging Windows Update to Distribute Firmware Updates

Model Based Servicing (MBS)

Presented by David Edfeldt,
Senior Program Manager
Windows PCE O3
(OEMs, ODMs, Operators)
Agenda

Model Based Servicing (MBS) Overview
  What is MBS?
  Why it matters
  How it works

Targeting + UEFI + PNP = MBS
  Targeting
  PNP interaction with UEFI
    ESRT, Capsule
  Driver details to consider

Wrap up

Questions
What is MBS?

A system centric method to service shipping systems.
Blends a flexible system targeting with ability to update both firmware and drivers.
Installing dependent firmware and driver sets possible.
Number of firmware/driver version mixes found in the field minimized, simplifying support.
Telemetry data on installation success.
Why MBS matters – Case study

- First round of Windows 8.1 AOAC systems shared bugs with firmware and sleep states, resulting in user initiated hard reboots.
- Solution found – needed 2 firmware changes (UEFI BIOS and a Device) and drivers
- One OEM was able to use MBS to update
- Other OEMs used their proprietary services to make the change available
User Interactive Hours Between Unexpected Shutdowns
MBS serviced & Other AOAC Devices
Nov 2013 to Mar 2014 - Median Trimmed Mean 99% with 5%/95% Bootstrap Confidence Intervals

The UEFI Forum
www.uefi.org
MBS set publishing workflow

1. Driver certification via Portal
2. Driver/FW signed
3. Create OEM manifest
4. Open bug on Portal
5. Publish OEM manifest
6. Publish OEM manifest
7. Collection + SMBIOS + Targeting Data
8. Users

Targeting + UEFI + PNP = MBS
Targeting + UEFI + PNP = MBS

- OEMs determine targeting (via SMBIOS fields)
- Use UEFI to
  1. use EFI System Resource Table (ESRT) to enable FW update path
  2. Install FW through Capsule updating
  3. Determine if new drivers should be activated
- Use expanded PNP rules to
  1. Identify updatable FW
  2. Handle updating decisions with Windows Update
  3. Prep capsule for FW update
  4. Allow new driver to install beside old driver and delay activation
Targeting + UEFI + PNP = MBS

Details of MBS install

- System contacts WU with targeting information and current driver/FW configuration.
- WU looks for MBS match. If MBS match found, WU collects updated drivers/FW.
- System downloads the set of drivers and/or firmware.
- If firmware is part of the set, store the package in UEFI capsule (prep next boot).

- System runs with MBS defined drivers/FW.
- If FW successful, Windows activates new drivers.
- All drivers are installed but those marked for reboot are not activated.
- Reboot! UEFI executes the FW installations found in capsule.
- If not, old drivers stay in use.

For details: windows-uefi-firmware-update-platform.docx.
Targeting + UEFI + PNP = MBS

Role of Computer Hardware IDs

- Computer Hardware IDs introduced in Windows 7 allows systems in device and printers view
- IDs are GUIDs generated from hashes of various model related BIOS fields
- Flexibility in which GUID to use, based on OEM’s business plan for populating these fields

OEMs create with Computer HWID
Targeting + UEFI + PNP = MBS

Options leverage common fields

- Manufacturer + Family + ProductName + SKUNumber + BIOS Vendor + BIOS Version + BIOS Major Release + BIOS Minor Release
- Manufacturer + Family + ProductName + BIOS Vendor + BIOS Version + BIOS Major Release + BIOS Minor Release
- Manufacturer + ProductName + BIOS Vendor + BIOS Version + BIOS Major Release + BIOS Minor Release
- Manufacturer + Family + ProductName + SKUNumber
- Manufacturer + Family + ProductName
- Manufacturer + SKUNumber
- Manufacturer + ProductName
- Manufacturer + Family
- Manufacturer + Enclosure Type
- Manufacturer

These combinations are all usable for targeting. Which to use is OEM’s decision.
Targeting + **UEFI** + PNP = MBS

*Starts with EFI System Resource Table (ESRT)*
ESRT adds Firmware in Device Manager

Unique GUID in ESRT identifies the devnode

ESRT values bubble up in Devnode properties

www.uefi.org
Targeting + UEFI + PNP = MBS

PNP preps FW installation using Capsule

PNP mechanism sets up both driver and firmware installs

- PNPid identifies
- FW update packages in an INF, handled like a driver
- Must be signed by MS or an authority locally authenticated
- PNP places the Firmware in capsule

UEFI does the firmware install

- UpdateCapsule(), CapsuleHeaderArray, and QueryCapsuleCapabilities() used for setup, install
- A single firmware.bin for each capsule update, no dependency possible
- Multiple capsules possible
- Ordering no possible – if order important use a single package.
- Installation results recorded into ESRT so OS can determine if new driver should activate

windows-uefi-firmware-update-platform.docx

The UEFI Forum
www.uefi.org
Targeting + UEFI + PNP = MBS

Driver changes for updating with firmware

NO firmware dependencies for a driver?
• Driver does not need modification.
• Driver will install and be used immediately

Firmware dependencies for a driver?
• Driver activation waits on new firmware installed, ie, at reboot.
• Driver changes needed:
  1. Run drivers from the driver store (allows multiple driver version to be installed and kernel to pick correct version).
  2. Adjust references to HAL extensions since can no longer be assumed to be in same location.
  3. Include the reboot directive on any drivers that should not install until after the firmware is installed.

Documentation on writing drivers with activation on reboot is available.
Keys to success:

1. Thoughtful use of SMBIOS fields for targeting
2. ESRT completed so firmware can be targeted
3. Firmware updates designed to leverage capsule installations
4. Firmware updates packaged in INFs so they can use PNP to install.
5. Driver INFs configured so installation can occur with firmware
Status of MBS

• Limited to participating SoC systems H1/2014
• 8 OEMs either using or testing currently
• Looking to enroll up to 3 OEMs to expand across product lines during 2014
• Further targeted training in planning stages

Interested? Follow-up with david.edfeldt@microsoft.com
For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org