System Firmware and Device Firmware Updates using Unified Extensible Firmware Interface (UEFI) Capsules

Fall 2018 UEFI Plugfest
October 15 – 19, 2018
Presented by Brian Richardson (Intel)

Materials by Michael Kinney (Intel)

www.uefi.org
Building and Distributing UEFI Capsules for Firmware Update

1. Generate UEFI Capsule

2. Publish UEFI Capsule

3. Distribute UEFI Capsule

4. Process UEFI Capsule

Microsoft* Windows Update or Linux* Vendor Firmware Service (LVFS) https://fwupd.org

* Other names and brands may be claimed as property of others
Platform Initialization (PI) Architecture

Firmware Phases

- Pre EFI Initialization (PEI)
- Driver Execution Environment (DXE)
- Boot Device Selection (BDS)
- Transient System Load (TSL)
- Exposed Platform Interface

Security (SEC)
- Pre EFI Initialization (PEI)
- Driver Execution Environment (DXE)
- Boot Device Selection (BDS)
- Transient System Load (TSL)
- Exposed Platform Interface

Power on ➔ [ . . Platform initialization . . ] ➔ [ . . . OS boot . . . ] ➔ Shutdown

www.uefi.org
**PI Architecture Firmware Phases**

**Example: UEFI Capsule Processing**

- **Pre EFI**
  - Initialization (PEI)
- **Driver Execution Environment (DXE)**
- **Boot Device Select (BDS)**
- **Transient System Load (TSL)**
- **Run Time (RT)**
- **After Life (AL)**

- **Receive UEFI Capsules (LVFS)** based on matching ESRT
- **Load UEFI Capsule**
- **Call UpdateCapsule()**
- **Pass UEFI Capsule to matching Firmware Management Protocol**
- **Verify Update from ESRT**

**Normal Boot**
- **ResetSystem()**
- **Final OS Boot Loader**
- **Save UEFI Capsule**

**Boot on FLASH Update**
- **Coalesce UEFI Capsules**
- **Process UEFI Capsules**
- **ResetSystem()**

**Power on**
- [.. Platform initialization ..]
- [.. OS boot ..]
- Shutdown

ESRT = EFI System Resource Table

[www.uefi.org](http://www.uefi.org)
Process UEFI Capsule

**FMP =** UEFI Firmware Management Protocol

**GUID =** Globally Unique Identifier

---

**UEFI Capsule**

- **UEFI Capsule Header**
  - FMP Header
  - Auth Info
  - Payload Header (Extensible)
  - Payload

**System Firmware**

SetImage()

**Authenticate**

1. **FMP Driver**
   - ImageTypeId
   - GUID A
   - Public Key(s)

2. **SetImage()**

3. **Publish**

4. **Update**

**ESRT Table**

- GUID A

---

www.uefi.org
# EDK II UEFI Capsule Features

EFI Development Kit II ([https://www.tianocore.org](https://www.tianocore.org))

<table>
<thead>
<tr>
<th>Feature</th>
<th>UDK2017 / UDK2018</th>
<th>edk2-stable201808</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate UEFI Capsule</td>
<td>Integrated EDK II Build</td>
<td>Standalone Python* Script</td>
</tr>
<tr>
<td>Update Granularity</td>
<td>Focused on Monolithic</td>
<td>Designed to support Multiple Components</td>
</tr>
<tr>
<td>Authentication</td>
<td>PKCS7 Single Key</td>
<td>PKCS7 Multiple Keys</td>
</tr>
<tr>
<td>Pre Check</td>
<td>N/A</td>
<td>Power/Battery, Thermal, System</td>
</tr>
<tr>
<td>Update Indicator</td>
<td>Requires platform code</td>
<td>Built-in with Consistent UX and Progress Bar</td>
</tr>
<tr>
<td>Firmware Management Protocol (FMP)</td>
<td>Requires full implementation</td>
<td>Produced by FmpDxe module customized using configuration data and small libraries</td>
</tr>
<tr>
<td>Test Key Detection</td>
<td>Requires platform code</td>
<td>Built-in</td>
</tr>
<tr>
<td>Watchdog</td>
<td>Requires platform code</td>
<td>Built-in</td>
</tr>
<tr>
<td>ESRT Driver</td>
<td>Legacy + FMP</td>
<td>Smaller/Simpler FMP only version</td>
</tr>
</tbody>
</table>

* Other names and brands may be claimed as property of others
Firmware Update Indicators

UEFI Graphics Console

- EFI_GRAPHICS_OUTPUT_PROTOCOL

- System Logo

- User Experience (UX) Capsule
  - Bitmap Message

UEFI Text Console

- EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL

- Update Progress - 100%
- Update Progress - 100%
- Update Progress - 100%
- Update Progress - 32%

Customize with a new DisplayUpdateProgressLib instance

www.uefi.org
FmpDxe Module Overview

FMP DXE Module
Configured through PCDs
Produces UEFI Firmware
Management Protocol

- FmpAuthenticationLib
- BaseCryptLib
- OpenSslLib
- FmpPayloadHeaderLib
- FmpDeviceLib
- CapsuleUpdatePolicyLib

PCD = Platform Configuration Database
### FmpDxe Module Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FILE_GUID</strong></td>
<td>ESRT GUID Value</td>
</tr>
<tr>
<td><strong>PcdFmpDeviceImageIdName</strong></td>
<td>FMP Image Descriptor - Unicode string</td>
</tr>
<tr>
<td><strong>PcdFmpDeviceBuildTimeLowestSupportedVersion</strong></td>
<td>Build time FMP/ESRT default value</td>
</tr>
<tr>
<td><strong>PcdFmpDeviceLockEventGuid</strong></td>
<td>Event GUID to lock FW storage device. Default is End of DXE.</td>
</tr>
<tr>
<td><strong>PcdFmpDeviceProgressWatchdogTimeInSeconds</strong></td>
<td>Watchdog armed on each progress update</td>
</tr>
<tr>
<td><strong>PcdFmpDeviceProgressColor</strong></td>
<td>24-bit Progress Bar Color (0x00rrggbb)</td>
</tr>
<tr>
<td><strong>PcdFmpDevicePkcs7CertBufferXdr</strong></td>
<td>One or more PKCS7 Certs in XDR format. Encode w/ <a href="www.uefi.org">BaseTools/Scripts/BinToPcd</a></td>
</tr>
<tr>
<td><strong>PcdFmpDeviceTestKeySha256Digest</strong></td>
<td>Set to {0} to disable test key detection</td>
</tr>
</tbody>
</table>

XDR = External Data Representation using Variable-Length Opaque Data format from RFC 4506

[www.uefi.org](http://www.uefi.org)
## CapsuleUpdatePolicyLib APIs
### Platform Specific Library

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckSystemPower()</td>
<td>Is system power/battery ok for FW update?</td>
</tr>
<tr>
<td>CheckSystemThermal()</td>
<td>Is system temperature ok for FW update?</td>
</tr>
<tr>
<td>CheckSystemEnvironment()</td>
<td>Is the system environment ok for FW update?</td>
</tr>
<tr>
<td>IsLowestSupportedVersionCheckRequired()</td>
<td>Skip lowest supported version check? (e.g. Service Mode)</td>
</tr>
<tr>
<td>IsLockFmpDeviceAtLockEventGuidRequired()</td>
<td>Skip firmware storage device lock action? (e.g. Manufacturing Mode)</td>
</tr>
</tbody>
</table>
# FmpDeviceLib APIs

## Device Specific Library

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegisterFmpInstaller()</td>
<td>Future expansion for add-in controllers.</td>
</tr>
<tr>
<td>FmpDeviceGetSize()</td>
<td>Size of <em>currently stored FW image</em>.</td>
</tr>
<tr>
<td>FmpDeviceGetImageTypeIdGuidPtr()</td>
<td>ESRT/FMP GUID. Overrides FILE_GUID value.</td>
</tr>
<tr>
<td>FmpDeviceGetAttributes()</td>
<td>FMP Attributes Supported/Settings.</td>
</tr>
<tr>
<td>FmpDeviceGetLowestSupportedVersion()</td>
<td>LSV from <em>currently stored FW image</em>.</td>
</tr>
<tr>
<td>FmpDeviceGetVersionString()</td>
<td>Unicode version string from <em>currently stored FW image</em>.</td>
</tr>
<tr>
<td>FmpDeviceGetVersion()</td>
<td>32-bit version value from <em>currently stored FW image</em>.</td>
</tr>
<tr>
<td>FmpDeviceGetImage()</td>
<td>Retrieve copy of <em>currently stored FW image</em>.</td>
</tr>
<tr>
<td>FmpDeviceCheckImage()</td>
<td>Check if a new FW image is valid for this device.</td>
</tr>
<tr>
<td>FmpDeviceSetImage()</td>
<td>Update FW storage with a new FW image.</td>
</tr>
<tr>
<td>FmpDeviceLock()</td>
<td>Lock FW storage to prevent any further changes.</td>
</tr>
</tbody>
</table>
ESRT GUIDs and Keys
(Multiple Components)

System Firmware

FMP Driver
- ImageTypeId GUID A
- Vital Product Data (VPD)
- Public Key(s)

FMP Driver
- ImageTypeId GUID B
- Public Key(s)

FMP Driver
- ImageTypeId GUID C
- Public Key(s)

FMP Driver
- ImageTypeId GUID D
- Public Key(s)

ESRT
- GUID A
- GUID B
- GUID C
- GUID D

www.uefi.org
ESRT GUIDs and Keys

3rd Party FMP Driver

3rd Party UEFI Capsules must be re-signed with System Key

www.uefi.org
ESRT GUIDs and Keys

3rd Party FMP Driver

System allows UEFI Capsules from 3rd Party to be installed

www.uefi.org
Add FMP to Existing Device Driver

FMP Library
- ImageTypeId: GUID A
- Public Key(s)

FMP Driver
- ImageTypeId: GUID B
- Public Key(s)

FMP Driver
- ImageTypeId: GUID C
- Public Key(s)

System Firmware

ESRT
- GUID A
- GUID B
- GUID C

www.uefi.org
Summary

• New UEFI Capsule Update Features in EDK II
  – Platform firmware and device firmware (ESRT/FMP)
  – Multiple authentication keys & test key detection
  – Improved UX and system update pre-checks
• Simplified capsule generation (Python script)
• Supports OS-based firmware update workflow
  – Model Based Servicing via Microsoft Windows Update
  – Linux Vendor Firmware Service (LVFS) via fwupd.org
Call to Action

• Add UEFI Capsule Support to platforms
• Implement UEFI Capsule Update for devices
• Take advantage of EDK II FmpDevicePkg features
• Use Windows Update & LVFS to simplify distribution of firmware updates
• Provide feedback and contribute!
  – TianoCore - https://www.tianocore.org/
  – LVFS - https://fwupd.org/
Thanks for attending the Fall 2018 UEFI Seminar and Plugfest

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

Intel, the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© Intel Corporation.