Seamless Update
–
Generating Update Capsule

April 30, 2020

1. Quick overview

A seamless update is defined as a firmware ingredient update that

- Stages firmware OOB with minimal performance impact to running workload,
- Activates the firmware with less than 10s blackout for VMs running on the host.

When reset/re-init is unavoidable for activation,

- Minimize the re-init domain to the absolute minimum possible (e.g. do not reset the system to re-initialize OS)
- Minimize latency of such reset/re-init

Seamless update may include:

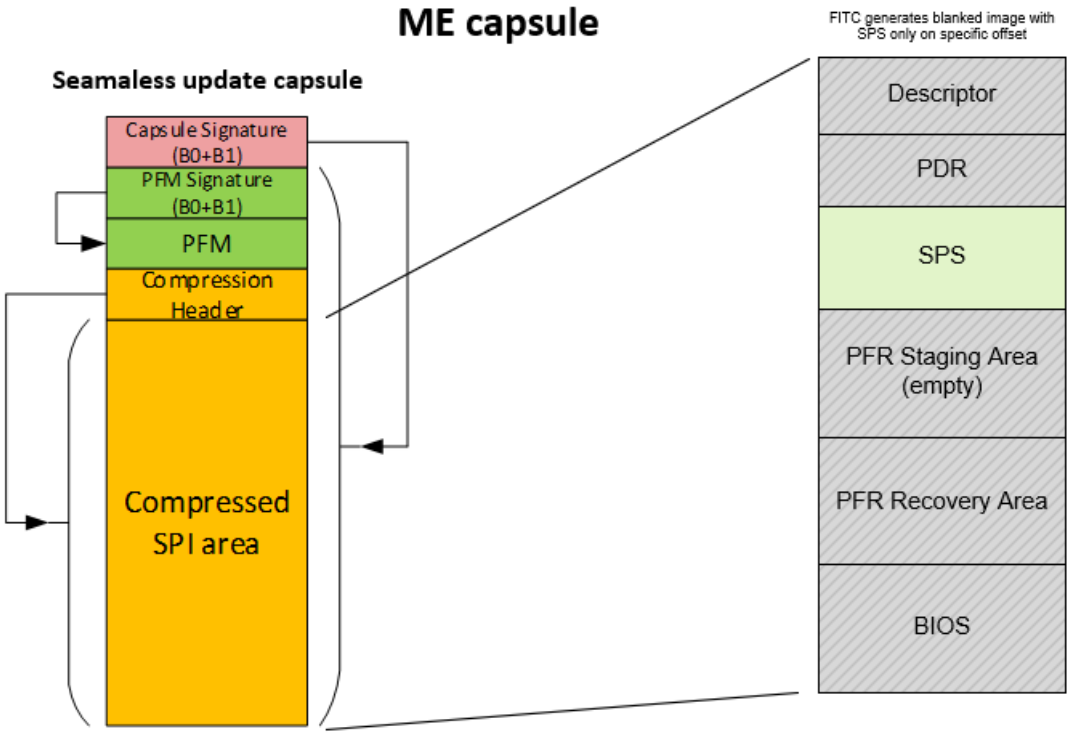
- ME-SPS FW Update
- uCode FV Update
- BIOS FW Update
- DCPMM FW Update
- SSD FW Update
- Etc.

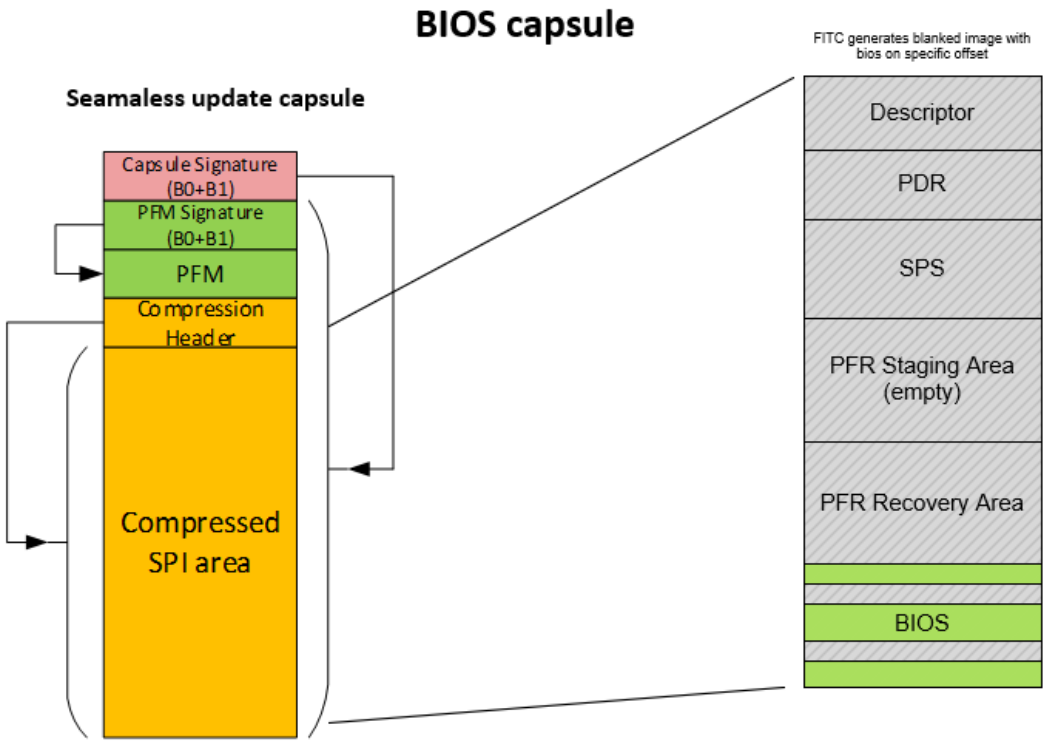
In order to perform seamless update, special update capsule needs to be generated. On Intel® Whitley CRB Wilson City platform, Intel® reference implementation of seamless update capsule is using PFR (Platform Firmware Resilience) capsule format with extensions. The FITc and IBST tools released with Intel® Whitley Server Platform Services (SPS) FW release package are used for seamless update capsule generation in this reference implementation.

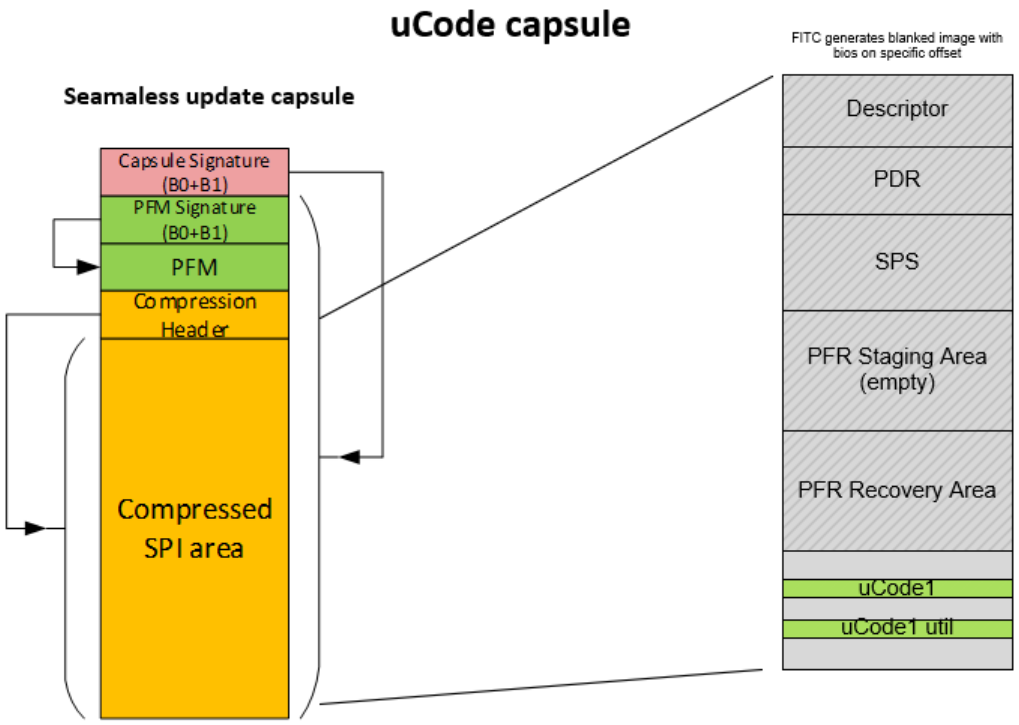
This user guide provides step-by-step instructions as well as a batch script on Intel® Whitley CRB Wilson City platform to generate seamless update capsules for ME-SPS FW, uCode FV, and BIOS FW.

Note: uCode FV is assumed to be generated by customer in-house or 3rd party tools.

2. Capsule Structure



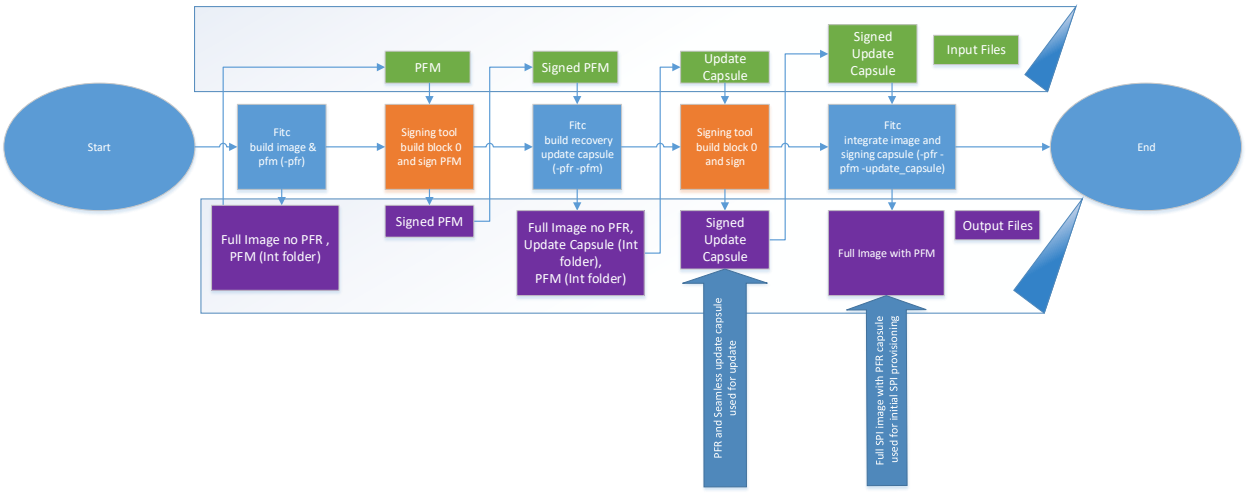




3. Capsule Generation

3.1 Flow

To create the seamless update capsule, a combination of a stitching and a signing tool is required. In Intel® reference implementation, SPS FITc and IBST are used. IBST can also be replace by an in-house block signing tool. The whole process is given in the diagram below.



3.2 Capsule Configuration XML File

To create seamless update capsule, a capsule configuration XML file need to pass to FITc. The configuration XML file contains PFM entry(ies) that specifies the region name, region start and end offset in the SPI flash for the update capsule. The following shows the PFM entry definition in sample capsule configuration XML files for seamless ME-SPS, BIOS, and uCode FV update on Intel® Whitley CRB Wilson City platform.

ME-SPS FW update capsule:

```
<PfmCfg>
    <PfmEntries>
        <EntryPFM>
            <RegionName value="ME" />
            <StartOffset value="0x0" />
            <EndOffset value="0x0" />
        </EntryPFM>
    </PfmEntries>
</PfmCfg>
```

Bios update capsule:

```
    <PfmEntries>
        <EntryPFM>
```

```
            <RegionName value="BIOS" />
            <StartOffset value="0x0" />
            <EndOffset value="0x880000" />
        </EntryPFM>
        <EntryPFM>
            <RegionName value="BIOS" />
            <StartOffset value="0x900000" />
            <EndOffset value="0xA40000" />
        </EntryPFM>
        <EntryPFM>
            <RegionName value="BIOS" />
            <StartOffset value="0xCE0000" />
            <EndOffset value="0x0" />
        </EntryPFM>
    </PfmEntries>
```

uCode1 FV update capsule:

```
    <PfmEntries>
        <EntryPFM>
            <RegionName value="BIOS" />
            <StartOffset value="0xA40000" />
            <EndOffset value="0xB90000" />
        </EntryPFM>
    </PfmEntries>
```

uCode2 FV update capsule:

```
    <PfmEntries>
        <EntryPFM>
            <RegionName value="BIOS" />
            <StartOffset value="0xB90000" />
            <EndOffset value="0xCE0000" />
        </EntryPFM>
    </PfmEntries>
```

3.3 Step by Step via CLI

1. First step is to generate PFM with proper configuration. The capsule configuration xml with PFM entries node like listed in section 3.2 needs to be passed to the FITc tool.

   CLI command should look like:

   ```
   spsFITc.exe -b -pfr -rcv <path to spsRecovery> -o <output path to spi_image> -bios <path to bios.rom> -f <path to capsule config xml file> -gbe <path to gbe> -pdr <path to pdr> -der1 <path to der1>
   ```

   Note: It is important to have exactly the same image map as the image flashed on platform, since the offsets of regions must be the same.

   PFM will be created in the ***Int*** folder next to the *spsFITc.exe* with name: ***PFM.bin***.

2. As a second step PFM has to be signed. IBST tool is used for this purpose on Intel® reference implementation. IBST tool needs two keys for signature and PAC.xml that contains the signature structure.

   CLI command should look like:

   ```
   python ibst.py PAC.xml –skip_valid -s root_key=<path to root key> csk_key=<path to csk key> binary=<path to generated PFM> -o <output path to signed PFM>
   ```

3. In third step unsigned Update Capsule will be generated. You should use exactly the same command as in first step with one more parameter:

   ```
   spsFITc.exe -b -pfr -pfm <path to signed PFM from 2 step> -rcv <path to spsRecovery> -o <output path to spi_image> -bios <path to bios.rom> -f <path to capsule config xml file> -gbe <path to gbe> -pdr <path to pdr> -der1 <path to der1>
   ```

   Update Capsule will be created in the ***Int*** folder next to the *spsFITc.exe* with name: ***UpdateCapsule.bin***.

4. The last step is to sign the created capsule with the IBST tool:

   ```
   python ibst.py PAC.xml –skip_valid -s root_key=<path to root key> csk_key=<path to csk key> binary=<path to generated update capsule> pc_type=0x5 -o <output path to signed update capsule>
   ```

   Signed update capsule is ready to update ME/Bios/uCode seamlessly. You find it in *<output path to signed update capsule>*!

3.3 Capsule Generation with Batch Scripts

Step1: Prepare capsule generation workspace for the sample batch script in the release package

   a. unzip CapsuleGenerationEnv.zip to your workspace.

b. From SPS FW release package, copy "**FlashImageTool**" folder to CapsuleGenerationEnv\
c. From SPS FW release package, copy "**IbstTool**" folder to CapsuleGenerationEnv\, and follow IBST tool user guide (IBSTtool_UG_Whitley.pdf) to install the tool, and generate root key (**key_root_prv.pem**) and csk key (**key_csk_prv.pem**) - refer to Intel® PFR Specification for details
d. Copy CapsulegenerationEnv**PAC.xml** to \CapsuleGenerationEnv\IbstTool\PAC.xml
e. From SPS FW release package, copy "spsRecovery.bin" to \CapsuleGenerationEnv\
f. From BIOS or IFWI release package, copy the **16MB BIOS rom file** (for example, WLYDCRB.SYS.WR.64.2019.49.5.04.1731_0013.D33_P80002_LBG_SPS_SMLS.rom) to \CapusleGenerationEnv\
g. Copy **pdr.bin**, **gbe.bin**, **der.bin** (if those regions are enabled in SPI flash image) to \CapsuleGeneration\Env

Step2: Capsule Generation

-- Generate BIOS update capsule

a. In powershell, run `.\create_SU_capsule.cmd .\BiosUpdate.xml <path to 16MB bios rom>`
b. BiosUpdateCapsule.bin will be generated under current folder.

-- Generate ME-SPS FW (full image) update capsule

a. In powershell, run `.\create_SU_capsule.cmd .\MeUpdate.xml <path to 16MB Bios rom>`
b. MeUpdateCapsule.bin will be generated under current folder.

-- Generate uCode FV update capsule

a. Copy uCode FV binaries under CapsuleGenerationEnv\
b. To generate uCode update capsule for FV1, n powershell, run `.\create_SU_capsule.cmd .\uCode1Update.xml <path to 16MB uCode FV binary>`
c. uCode1UpdateCapsule.bin will be generated under current folder.
d. To generate uCode update capsule for FV2, run `.\create_SU_capsule.cmd .\uCode2Update.xml <path to 16MB uCode FV binary>`