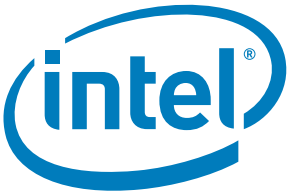


LakeField Platform Intel® Converged Security Engine 13.30 and Intel® Sensor Hub Firmware

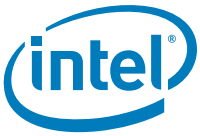
Compliance and Testing Guide

Revision 1.0

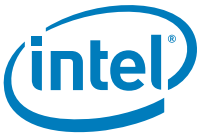
March 2020

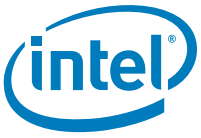
Intel Confidential

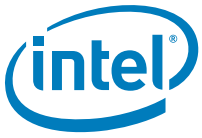
Document Number: 575826



Contents







Figures

Tables

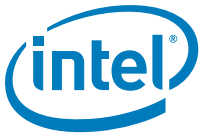
Revision History

Document Number	Revision Number	Description	Revision Date
575826	0.5	• **Initial Release**	December 2017
	0.6	• **Added the following chapters** — Intel® Integrated Clock Controller Compliancy. — EC Lite FW Compliancy. • **Removed the following chapters** — Intel® IOC . — Intel® IPT . — FW Capsule Update. — Intel® Trusted Execution Engine. • **Intel® CSE 13.30 Power Management and Stress Testing:** — Changed all S3 test cases to S0ix. — Removed all S4 test cases. — All stress test cases moved to DC only testing. • **Intel® Trace Hub** — Updated required probes for testing LakeField platform. • **Intel® Boot Guard 2.1:** — Updated test cases according to Boot Guard 2.1 architecture. • **PCH Soft Strap Configuration:** — Updated to align with LakeField. • **Protected Media Playback:** — Updated screen captures of Intel® FIT according to LakeField tool.	April 2018
	0.7	• **Added new chapters:** — Dekel PHY Compliancy. — Universal Flash Storage (UFS) Compliancy. • **Updated OS support across all chapters.** • **Intel® CSE 13.3 BIOS Copmliancy:** — Removed test case BIOS_05. • **Intel® CSE 13.30 Power Management and Stress Testing:** — Added information on LKF power state change. • **ISH FW and Platform Sensors Compliancy** — Removed S3 and DeepSx test steps from test case ISS_TST_14. • **Download and Execute (DnX)** — Updated test environment setup and tools. — Re-defined existing test cases and added new ones. • **Intel® Boot Guard 2.1 Compliancy:** — Updated Information for supported TXT BtgInfo tool. — Updated the objective and pass criteria for tests BTG_003 and BTG_005. • **Signing and Manifesting and Secure Tokens Compliancy:** — Updated Debug Tokens test case SIGN_SECTOK_03 procedure. • **Intel® Trace Hub Compliancy** — Removed tests cases for Capturing ITH CSE trace from S0-S3 and from S3-30. — Renumbered test cases. • **Protected Media Playback:** — Revised the introduction and updated supported boot media.. • **Intel® Integrated Clock Controller Compliancy:** — Added a note in ICC_TST_10 to explain EOP restriction.	September 2018

Document Number	Revision Number	Description	Revision Date
575826	0.8	• **LakeField Intel® CSE BIOS Compliance** — Updated test BIOS_02 to include the transition from S5 to S0. • **Intel® CSE 13.30 FW - Power Management and Stress Testing** — added test ME_PM_9 for testing the flow S0 to S4 to S0. — Updated every power button long press test step to specify the required pressing time. — Updated test ME_PM_3 to include returning the machine to S0 and check for the last boot reason. • **Universal Flash Storage (UFS) Compliancy** — Updated note regarding config file location for partitioning. — Added notes to DnX User Guide for some test cases. • **ISH FW and Platform Sensors Compliancy** — Replaced AWS term with PETS. — Adjusted test case numbers. • **Intel® Platform Trust Technology (Intel® PTT) Compliancy** — Removed S4 from test PTT_009. — Updated Bitlocker auth loading percentage to 3% in test PTT_009. • **Download and Execute (DnX) Compliancy** — Added tests to Write / Read / Erase Unlock Token through DnX. — Updated note regarding config file location for partitioning. — Fixed cross reference. • **Intel® Boot Guard 2.1:** — Added a note to test coverage summary to clarify the reference for "Verified Only". • **Signing and Manifesting and Secure Tokens Compliancy:** — Removed test case SIGN_SECTOK_01. — Updated test case SIGN_SECTOK_02. • **Intel® Trace Hub** — Updated test environment setup. — Merged tests together. • **Platform Controller Hub (PCH) SoftStrap Configuration** — Corrected PCIe Controller 1 and 2 lane configurations. • **Dekel PHY Compliancy:** — Updated test IDs. — Updated test summary table. — Updated Dekel PHY version scheme.	March 2019
	0.85	• **Intel® CSE 13.30 FW - Power Management and Stress Testing** — Corrected typo in test ME_PMST_4 test name. — Updated global reset steps. • **Universal Flash Storage (UFS) Compliancy** — Updated tests UFS_01 and UFS_05. — Updated tests execution status from Manual to Interactive with Intel® PETS. • **Download and Execute (DnX) Compliancy** — Updated tests DnX_01, DnX_04, and DnX_06. — Updated tests execution status from Manual to Interactive with Intel® PETS. • **Intel® Boot Guard 2.1:** — Updated the description and procedure of test BTG_001. • **Intel® Integrated Clock Controller Compliancy:** — Updated test cases to align with the updated Intel® CCT tool commands. • **Platform Controller Hub (PCH) SoftStrap Configuration** — Corrected FDO jumper location. — Updated PCIE root port mapping.	May 2019
	0.9	• **Added Intel® DAL Compliancy Chapter** • **ISH FW and Platform Sensors Compliancy** — Removed test ISS_TST_8. • **Download and Execute (DnX) Compliancy** — Updated Write_OEMUnlockToken test DnX_07. • **Intel® Boot Guard 2.1:** — Updated the testing tool to be BPMGen2 instead of Intel® MEU.	August 2019

Document Number	Revision Number	Description	Revision Date
575826	0.95	• **Intel® CSE 13.30 FW - Power Management and Stress Testing** — Added an additional step in the stress flow test cases to check if there is any flash log in system. • **Universal Flash Storage (UFS) Compliancy** — Updated use cases for UFS. — Updated Test Environment and Tools section. — renamed tests UFS_04,05 • **Intel® Dynamic Application Loader Compliancy:** — Added a note indicating that Intel® DAL applets should be signed with RSA 3K. • **Intel® Integrated Clock Controller Compliancy:** — Updated test cases to align with the removal of Intel® CCT tool.	October 2019

§ §



1 Introduction

1.1 Purpose and Scope

Intel® CSE Compliance Guide for LakeField platforms is designed to provide original equipment and device manufacturers with the compliancy requirements for Intel® CSE 13.30 based platforms. It provides implementation, methodology and tools to verify compliance for the various Intel® CSE and Intel® ISH FW core components and technologies. It also provides the test environment setup information, the procedure for each test, and the expected results for the purpose of validating compliancy.

Requirements contained in this document target the system BIOS, Intel® CSE, Intel® ISH and other aspects of overall platform implementation.

Note: All tests can be run without using Intel® APS device except for the Power Management tests. Without an Intel® APS device, configure Intel® PETS console SUT power settings (under the Intel® APS tab) to be Manual rather than controlled by Intel® APS device. Doing so, Intel® PETS relies on the user to verify, if the system is in S0, ACDC Power supplied, so on via user prompt questions. When doing the power management tests, it is best to do them with APS device for reliable signal measurements and results. The APS device and software qualifies, if the platform transitions into the respective power state appropriately based on the signal level voltages of the board. Refer to the APS connection guide for more details on each power state signal status expectations. The board power state can be manually measured as each compliance test mentions “verify” or “observe” using a multi-meter.

1.2 Acronyms and Definitions

1.2.1 General

Acronym or Terminology	Definition
DnX	Download and Execute
FDV	Flash Descriptor Valid
FPF	Field Programmable Fuses
fTPM	Firmware Trusted Platform Module- Intel® implementation of TCG TPM 2.0 in firmware.
Intel® APS	Intel® Automatic Power Switch (Intel® APS) System State Test Device
Intel® FIT	Intel® Flash Image Tool
Intel® FPT	Intel® Flash Programming Tool
Intel® CSE	Intel® Converged Security Engine
Intel® PETS	Intel® Platform Enablement Test Suite
ISH	Integrated Sensor Hub
LKF	LakeField Platform
PM	Power Management
PTT	Platform Trust Technology, also known as fTPM
SPI	Serial Peripheral Interface
SUT	System Under Test
TCG	Trusted Computing Group
TPM	Trusted Platform Module
UFS	Universal Flash Storage

1.2.2 System States and Power Management

Acronym or Terminology	Definition
S0	A system state where power is applied to all Hardware devices and system is running normally (refer to latest industry ACPI specification).
S0i3	"Sleep", "Deep Sleep". Used when the user is not actively using the device. CPU in C6 (retained in shared SRAM). Sleep mode, always connected Able to wake from user or platform Screen off
S4	Lowest Power, longest wake latency sleeping state. OS context in Memory is saved into disk. All devices are powered off. Hibernate mode
S5	"Soft off". All devices are powered off. System Shutdown
M0	Intel® CSE FW power state where all HW power planes are activated and the host power state is S0.
M-Off	An Intel® CSE FW power state, where no power is applied to the CSE subsystem. (Intel® CSE FW is shut down).
OS Hibernate	When the OS saves state information to the hard disk

Acronym or Terminology	Definition
Standby	When the OS state is saved to memory and resumed from the memory when mouse, keyboard, or other activity occurs is configured as a wake event.
Shut Down	A state where the system power is off and the power cord is still connected.
Warm/Cold reset	The platform shall support restart (warm / cold) from System active state (S0) by closing the applications, initiate OS reboot sequence to bring the platform to S0 active state
Global reset	A full platform reset that includes the CSE sub system and host sub system

1.3 Intel® Platform Enablement Test Suite (Intel® PETS) Testing Guidelines

Intel recommends that customers run Intel® PETS tests, whenever there are any changes in:

- BIOS
- Intel® CSE Firmware
- EC Firmware
- Board/Silicon stepping changes

The following tests should be executed in the specified order:

1. Run Intel® PETS Setup Environment Test
2. Run Integrated Clock Control test Package
3. Run SPI test package
4. Run BIOS test package
5. Run Power Test packages
6. Run Feature tests depending on the SKU

1.4 Boot Guard—Discrete TPM and Intel® Platform Trust Technology (Intel® PTT)

The following table shows the configuration information for the Boot Guard—Discrete TPM and Intel® Platform Trust Technology (Intel® PTT) with respect to how they work with different operating systems and firmware combinations. Refer to Boot Guard and Intel® PTT chapter for actual compliance tests.

Definitions:

- Supported—Intel validates this combination
- Not Supported—Intel not validates this combination
- N/A—Not a valid combination from a validation standpoint

Table 1-1. Boot Guard—Discrete TPM and Intel® Platform Trust Technology (Intel® PTT)

Platform[1]	Intel® Active Client Manager (Intel® ACM)	Intel® CSE Firmware	Intel® PTT	TPM 1.2	TPM 2.0
LakeField Based	Intel® ACM 3.x	Intel® CSE 13.30	Yes	Yes	Yes

Note:
1. Refer to platform dashboard for POR configurations.

§ §

(This page is intentionally left blank)

2 LakeField Intel® CSE BIOS Compliance

Intel® CSE BIOS Compliance section serves as a checklist for the environment setup for the host BIOS and Intel® CSE interface testing and validation.

2.1 Intel® BIOS Compliance Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name[1]	OS Supported	Platform	How?
BIOS_01	End of POST	Compliance_BIOS.xml	W	LKF	A
BIOS_02	DRAM Init Done	Compliance_BIOS.xml	W	LKF	A

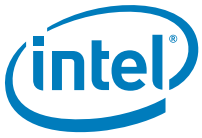
Note:
1. Tests BIOS_01, BIOS_02 and BIOS_05 can be done in one iteration after reset (no need for 3 separate resets). Tester may remove the resets from PETS SW between the tests.

2.2 End of POST

Test ID:	BIOS_01
Test Case Title:	End of POST
Objective:	Verify that the BIOS sends the END_OF_POST message when the platform is transitioning from S5 and before the BIOS boot process is done and the OS starts.
Platform:	LKF
Mandatory/Optional:	Mandatory
Test Pass Criteria:	Test passes if the BIOS Mode displays a status of POST Boot.
Description:	At the end of BIOS POST, system BIOS must send an "END OF POST" HECI message to CSE declaring end of POST and start of OS load. BIOS must also wait for an "END of POST" response message from CSE before proceeding.
Procedure:	Verify if END_OF_POST message was sent by BIOS: • EFI Procedure: — Boot to EFI Shell — Run: "MEInfo.efi" — Check that the value of BIOS boot state is "Post boot" • Windows* Procedure: — Boot to Windows* — From elevated CMD run: "MEInfo.exe" — Check that the value of BIOS boot state is "Post boot"

2.3 DRAM INIT DONE

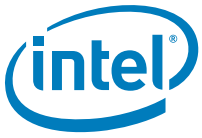
Test ID:	BIOS_02
Test Case Title:	DRAM Init Done
Objective:	Verify that the BIOS set the DRAM INIT Done bit
Platform:	LKF
Mandatory/Optional:	Mandatory
Test Pass Criteria:	The BIOS is required to send the DRAM INIT Done message to Intel® CSE to indicate that BIOS has initialized DRAM memory.
Description:	This message is sent by the IA FW to indicate to Intel® CSE firmware that DRAM initialization was completed and CSE UMA is ready for use.



Test ID:	BIOS_02
Procedure:	1. For each of the following system transitions: a. G3 -> S0 (M-Off->M0) b. S5 -> S0 (M3->M0) EFI Procedure: • Boot to EFI shell • Run the below command to read FWSTS#1 register • Run "PCI 0 22 0" command (add –I for verbose information) Check for offset 40h • ME Current Operating State Bit [8:6] should set to "001" - "M0 with UMA" **ME Current Operation State**: This field describes the state that CSE is currently functioning in at this moment. It is the combination of CSE power state and UMA. The "Current Operation State" is set only upon entering the true hardware state, Example: Set it to M0 only after PLL's are locked to M0 frequency and controller is set to SD. 000 – Preboot (Default) 001 – M0 with UMA 010 – M0 Power Gated 011 – Reserved 100 – M3 without UMA 101 – M0 without UMA 110 – Bring up 111 – M0 without UMA but with error

§ §

(This page is intentionally left blank)



3 Intel® Converged Security Engine (Intel® CSE) Manufacturing Mode Compliance

The Intel® Converged Security Engine Manufacturing Mode compliance chapter serves as a checklist for the environment setup for the host BIOS and Intel® CSE interface testing and validation when the Intel® CSE is in Manufacturing Mode.

The tests in this section verify that certain BIOS operations are *not* performed when the Intel® CSE is in Manufacturing Mode.

Test Environment for Intel® CSE BIOS Compliance section:

The system under test is to be configured with the Intel® CSE in manufacturing mode and Deep S4/S5 disabled.

Tools for Testing:

- Intel® Platform Enablement Test Suite—Latest version of the tool from the Intel® CSE Compliance kit release. Refer to the *Intel® Platform Enablement Test Suite User Guide* available in the Intel Compliance kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.
- Compliance_MeBios_ManufacturingMode.xml—Package should be loaded to Intel Platform Enablement Test Suite in order to complete this section.

3.1 Intel® Manufacturing Mode Compliance Test Coverage Summary

OS Support', and 'How?' Columns describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Form Factor: D = Desktop, M = Mobile, A = All in one, W = Workstation

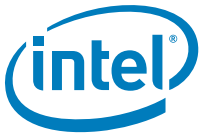
Network Factor: LAN = systems with LAN interface and test is performed using LAN interface, WLAN = systems with WLAN interface and test is performed using the WLAN interface.

Test ID	Test Case Title	How?	Intel® PETS Package Name	OS Support	Form Factor	Network Factor
BIOS_04	CF9GR locking/ unlocking - Manufacturing Mode	A	Compliance_MeBios_ManufacturingMode.xml	W	D M A	LAN+WLAN; WLAN only

3.2 CF9GR Locking/Unlocking

Test ID:	**BIOS_04**
Test Case Title:	CF9GR locking/unlocking - Manufacturing Mode
Mandatory/Optional:	Mandatory
Description:	When the system is in the Intel® CSE manufacturing mode, BIOS must set the CF9GR register (Memory-mapped address at PWRMBASE register offset 1048h [bit 20]) to '0' to allow host only resets before handing control to the OS. For the Intel® FPT tool to perform a global reset with parameter/GRESET, the BIOS must keep the CF9GR setting unlocked (by setting PWRMBASE register offset 1048h [bit 31] of the same register to '0').
Objective:	For security reasons, the BIOS must ensure that CF9GR is cleared and locked before handing control to the OS in the shipping machine. But for the usage of Intel® FPT tool with /GRESET parameter in the manufacturing environment, the BIOS must ensure that CF9GR reset mode can be changed by the Intel® FPT tool. ***Note:*** The recommended allocation of PWRMBASE is 0xFE000000 in PCH BIOS Specification.
Procedure:	1. Boot the system under test to OS. 2. Intel Platform Enablement Test Suite will perform the following: a. Manually read the PCI address space at PCH B0:D22:F0 register offset 40h [bit 4] to verify the Intel® CSE Manufacturing Mode bit is equal to '1'. b. Manually read the memory-mapped address at PWRMBASE register offset 1048h [bit 20] to verify the bit is set to '0'. c. Manually read the memory-mapped address at PWRMBASE register offset 1048h [bit 31] to verify the bit is set to '0.
Test Pass/Fail Criteria:	Test passes if the PWRMBASE register offset 1048h [bit 20] = '0' and [bit 31] of the same register is '0' when the system is in the Intel® CSE manufacturing mode.

§ §



(This page is intentionally left blank)

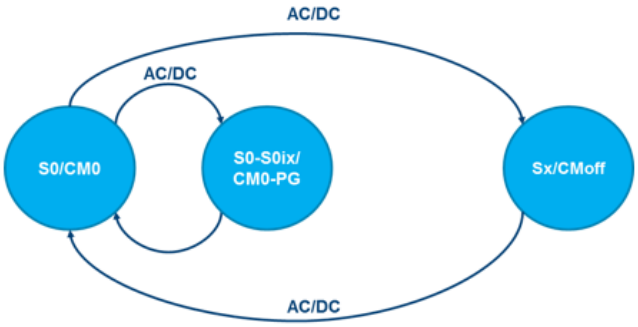
4 Intel® CSE 13.30 FW - Power Management and Stress Testing

This chapter covers system power flow transitions which involve the Intel® CSE Firmware. There are also tests in this chapter that are specifically intended to cover topics related to stress testing of the System under Test (SUT).

4.1 Introduction

Theres certain changes in LKF differ from ICL.

- LKF does not have S3 state.
- Deep-Sx is not supported for LKF. As the user can see in the picture, LKF has S0



state as an active state, Sx(S4, S5) state as a sleep state and does not support Deep-Sx state.

- LKF will support S4 and S5 from OS prospective : From OS perspective, hibernate(S4) and off(S5) are supported. Transition into Sx states(S4/S5) is supported, however when getting there, the SoC is placed in G3-like state. Therefore no Deep Sx support needed.
- From HW prospective S4/S5 will map to G3 with PMIC being only wake source and SoC having only RTC rail alive.
- PMIC will support power button, rtc alarm and charger insertion as wake sources during S4/S5.

4.2 Test Environment Setup

- System under Test (SUT) can be configured in either manual configuration mode or using enterprise provisioning mode.
- Install all platform drivers (Chipset, Graphics, WLAN).
- If there is a Global reset test to pass, then the system under test should be in manufacturing mode.

4.3 Test Step Execution and Verification

The tests described in this chapter contain test steps which are executed by Intel® PETS. While Intel® PETS brings a certain level of convenience and speed to the testing process, there are times where manual verification of steps in critical towards issue triage and debug. Review the Test Step Execution and Verification Section found for Intel® CSE Power Management tests before starting any stress tests in the chapter.

Stress tests in this chapter are designed to be run individually through a large number of iterations. Some of them require changing the system configuration before being run. When performing very large numbers of iterations, the tests may each take many hours, and is some cases several days.

Intel validation runs each of these tests the number of iterations indicated. Each OEM should decide on the tolerance level required for their boards, and choose an appropriate number of iterations.

Stress tests in this chapter are not designed to be run automatically one after the other; the test operator must place the SUT into an appropriate starting state, and then run the test cycle. However, each test individually ends with the SUT in the same state as when it started, allowing for easy iteration.

When running long iterations, ensure that the management console is set not to go to sleep, as this will pause the test.

Ensure that the SUT can boot to designated Host OS without prompting the test operator for any actions (such as scanning drivers and so forth); as this will effect stress tests which boot the SUT to the Host OS.

4.4 Tools for Testing

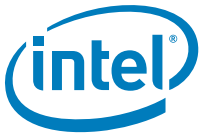
The following tools, as provided by Intel, may be used to execute automated tests listed herein:

- **Intel® Platform Enablement Test Suite** - Latest version of the tool is available in the Intel® CSE compliancy kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.
- **Intel® Automated Power Switch** – The SUT should be connected to an Intel® APS 3 unit. In case Intel® APS 3 is not available, select the Manual configuration in the Intel® PETS SUT profile configuration menu.
- **Intel® PETS Local Agent**: The agent must be installed on the SUT.
- It is recommended that power management tests (ME_PM) in this chapter be run on no less than 30% battery charge.
- It is recommended that stress tests (ME_PMST) in this chapter be run on no less than 90% battery charge.

4.5 Power Management Compliancy Test Coverage Summary

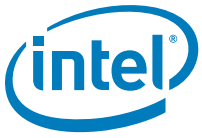
Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows *



How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
ME_PM_1	S0 to S0ix	ME_Compliance_PM.xml	W	A
ME_PM_2	S0ix to S0	ME_Compliance_PM.xml	W	A
ME_PM_3	S0 to S5 to S0	ME_Compliance_PM.xml	W	A
ME_PM_4	S5 to S0	ME_Compliance_PM.xml	W	A
ME_PM_5	Cold Reset	ME_Compliance_RST.xml	W	A
ME_PM_6	Global Reset	ME_Compliance_RST.xml	W	A
ME_PM_7	Warm Reset	ME_Compliance_RST.xml	W	A
ME_PM_8	S0 to G3	ME_Compliance_PM.xml	W	A
ME_PM_9	S0 to S4 to S0	ME_Compliance_PM.xml	W	A
ME_PMST_1	Host Reset from S0	Compliance_Power_Stress.xml	W	A
ME_PMST_2	S0 to S5 to S0 via Power Button Override	Compliance_Power_Stress.xml	W	A
ME_PMST_3	S0 to S0ix to S0 via Power Button Press	Compliance_Power_Stress.xml	W	A
ME_PMST_4	S0 to S5 to S0 by means of Shutdown and Power Button Press	Compliance_Power_Stress.xml	W	A



4.6 ME_PM_1: S0 to S0ix

Test ID:	ME_PM_1.1
Test Case Title:	S0 to S0ix by means of Power Button Suspend (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0ix
Test Pass Criteria:	The test passes if the SUT moves to S0ix and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify that a DC Battery is connected to the SUT, and that it is charged 4. Set the SUT power source to DC Only. 5. Verify CSE is working properly. 6. Suspend the SUT via the Power Button 7. Verify the SUT is in S0ix. 8. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

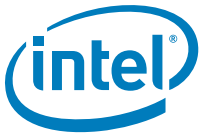
4.7 ME_PM_2: S0ix to S0

Test ID:	ME_PM_1.2
Test Case Title:	S0 to S0ix by means of Power Button Suspend (AC+DC, AConly)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0ix
Test Pass Criteria:	The test passes if the SUT moves to S0ix and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC, otherwise AC only 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Suspend the SUT via the Power Button 5. Verify the SUT is in S0ix. 6. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_2.1
Test Case Title:	S0ix to S0 by means of Power Button Press (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0ix to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash log found.

Test ID:	ME_PM_2.1
Procedure:	1. Set the SUT power source to AC+DC only. 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Verify that DC battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC-only. 6. Move the SUT to S0Ix via the Power Button 7. Verify the SUT is in S0ix 8. Press the Power Button on the SUT at least 5 seconds 9. Verify that the SUT is in S0 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

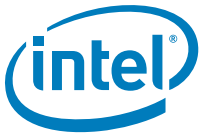
Test ID:	ME_PM_2.2
Test Case Title:	S0ix to S0 by means of Power Button Press (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0ix to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC. 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Move the SUT to S0ix via the Power Button 5. Verify the SUT is in S0ix 6. Press the Power Button on the SUT at least 5 seconds 7. Verify that the SUT is in S0 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")



4.8 ME_PM_3: S0 to S5 to S0

Test ID:	ME_PM_3.1
Test Case Title:	S0 to S5 to S0 by means of Host OS Shutdown (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S5 and S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC only 6. Shutdown the SUT via the Host OS 7. Verify the SUT is in S5. 8. Press the Power Button on the SUT at least 5 seconds 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 12. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 13. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_3.2
Test Case Title:	S0 to S5 to S0 by means of Host OS Shutdown (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S5 and S0 and the Intel® CSE is working properly with no flash log found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Shutdown the SUT via the Host OS 5. Verify the SUT is in S5. 6. Press the Power Button on the SUT at least 5 seconds 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 10. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")



4.9 ME_PM_4: S5 to S0

Test ID:	ME_PM_4.1
Test Case Title:	S5 to S0 by means of Power Button Press (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC Only 6. Shutdown the SUT via the Host OS 7. Verify the SUT is in S5. 8. Press the Power Button on the SUT at least 5 seconds 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_4.2
Test Case Title:	S5 to S0 by means of Power Button Press (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Shutdown the SUT via the Host OS 5. Verify the SUT is in S5. 6. Press the Power Button on the SUT at least 5 seconds 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

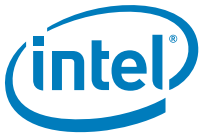
4.10 ME_PM_5: Cold Reset

Test ID:	**ME_PM_5.1**
Test Case Title:	S0 to S0 by means of Cf9 Cold Reset (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Record the Host OS last boot time on the SUT (to verify reset execution) 5. Ensure the Cf9h Global Reset is cleared to 0b 6. Perform a cold reset to the SUT by writing Eh to IO register CF9h. PCI configuration space for CF9h (Bus:Device:Function) is (0,F,0) and the offset is 0x40 7. Verify the SUT is in S0. 8. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset or that HOST OS is unavailable. 9. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.11 ME_PM_6: Global Reset

Test ID:	**ME_PM_6.1**
Test Case Title:	S0 to S0 by means of Cf9 Global Reset (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Record the Host OS last boot time on the SUT (to verify reset execution) 5. Perform a warm reset to the SUT by writing 6h or Eh to IO register CF9h. PCI configuration space for CF9h (Bus:Device:Function) is (0,F,0) and the offset is 0x40. ***Note:*** Perform a warm reset after sending HECI command mentioned in test step 5 will perform Global Reset. 6. Verify the SUT is in S0. 7. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset or that HOST OS is unavailable. 8. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**2**xxxxxx or 0x6**B**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 9. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_6.2
Test Case Title:	S0 to S0 by means of Cf9 Global Reset (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Verify that the DC Battery is connected, and that it is charged. 5. Set the SUT power source to DC-only. 6. Record the Host OS last boot time on the SUT (to verify reset execution) 7. Perform a warm reset to the SUT by writing 6h or Eh to IO register CF9h.PCI configuration space for CF9h (Bus:Device:Function) is (0,F,0) and the offset is 0x40. Note: Perform a warm reset after sending HECI command mentioned in test step 7 will perform Global Reset. 8. Verify the SUT is in S0. 9. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset or that HOST OS is unavailable. 10. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**2**xxxxxx or 0x6**B**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

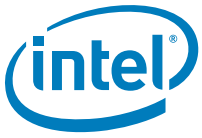


4.12 ME_PM_7: Warm Reset

Test ID:	ME_PM_7.1
Test Case Title:	S0 to S0 by means of Host OS Restart (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Record the Host OS last boot time on the SUT (to verify the reset execution). 5. Perform a warm reset of the SUT via Host OS graceful restart. 6. Verify the SUT is in S0. 7. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset, or that the Host OS is unavailable. 8. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**9**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 9. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_7.4
Test Case Title:	S0 to S0 by means of Host OS Restart (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Record the Host OS last boot time on the SUT (to verify the reset execution). 5. Verify that DC battery is connected, and that it is charged 6. Connect the SUT power source to DC-only 7. Perform a warm reset of the SUT via Host OS graceful restart. 8. Verify the SUT is in S0. 9. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset, or that the Host OS is unavailable. 10. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**9**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

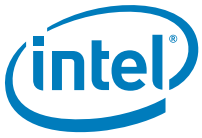
Test ID:	ME_PM_7.5
Test Case Title:	S0 to S0 by means of CF9 Warm Reset (AC+DC, AC only)



Test ID:	ME_PM_7.5
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager 4. Record the Host OS last boot time on the SUT (to verify the reset execution). 5. Perform a warm reset of the SUT by writing 6h to IO register CF9h. 6. Verify the SUT is in S0. 7. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset, or that the Host OS is unavailable. 8. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x69xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 9. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.13 ME_PM_8: S0 to G3

Test ID:	ME_PM_8.1
Test Case Title:	S0 to G3 by means of Power Loss (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT moves from S0 to G3 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly 4. Remove power from the SUT via AC-detach, and if necessary also via DC-detach. Wait for 10 seconds before continuing to allow full power drain from the SUT. 5. Verify the SUT is in G3. 6. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")



4.14 ME_PM_9: S0 to S4 to S0

Test ID:	ME_PM_9.1
Test Case Title:	S0 to S4 to S0 by means of Host OS Hibernate (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S4
Test Pass Criteria:	The test passes if the SUT moves to S5 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC only 6. Hibernate the SUT via the Host OS 7. Verify the SUT is in S4. 8. Press the Power Button on the SUT at least 5 seconds 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 12. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 13. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_9.2
Test Case Title:	S0 to S4 to S0 by means of Host OS Hibernate (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S4
Test Pass Criteria:	The test passes if the SUT moves to S4 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Hibernate the SUT via the Host OS 5. Verify the SUT is in S4. 6. Press the Power Button on the SUT at least 5 seconds 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 10. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.15 ME_PMST_1: Host Reset from S0

Test ID:	ME_PMST_1
Test Case Title:	Host Reset from S0 (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0 by means of Host Reset
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash log found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available. 3. Verify CSE is working properly 4. Record the Host OS last boot time on the SUT (to verify reset execution) 5. Perform a warm reset of the SUT by performing a host reset. 6. Verify the SUT is in S0. 7. Verify that Host OS on the SUT is available. 8. Verify CSE is working properly 9. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset. 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

4.16 ME_PMST_2: S0 to S5 to S0 via Power Button Override

Test ID:	ME_PMST_2
Test Case Title:	S0 to S5 to S0 by means of Power Button override cycle (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0 by means of Power Button Override
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash log found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available 3. Verify CSE is working properly 4. Shutdown the SUT via Power Button Press for more than 5 seconds. 5. Verify that the SUT is in S5 6. Press the Power Button on the SUT at least 5 seconds 7. Verify the SUT is in S0. 8. Verify that Host OS on the SUT is available. 9. Verify CSE is working properly 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

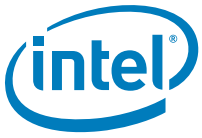
4.17 ME_PMST_3: S0 to S0ix to S0 via Power Button Press

Test ID:	ME_PMST_3
Test Case Title:	S0 to S0ix to S0 by means of Suspend and Power Button Press (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0Ix to S0 by means of Suspend and Power Button Press
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash log found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available 3. Verify CSE is working properly 4. Suspend the SUT via the Power Button 5. Verify that the SUT is in S0ix 6. Press the Power Button on the SUT at least 5 seconds 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

4.18 ME_PMST_4: S0 to S5 to S0 via Shutdown and Power Button Press

Test ID:	ME_PMST_4
Test Case Title:	S0 to S5 to S0 by means of Shutdown and Power Button Press (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0 by means of Shutdown and Power Button Press
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash log found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available 3. Verify CSE is working properly 4. Shutdown the SUT via the Host OS 5. Verify that the SUT is in S5 6. Press the Power Button on the SUT at least 5 seconds 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

§ §



(This page is intentionally left blank)

5 Serial Peripheral Interface (SPI) Configuration

The test cases in this chapter are created to verify the correct configuration of the LakeField SoC SPI Host Controller. Test cases in this section verify implementation of SPI Dual and Quad I/O Fast Read, SPI Flash Descriptor mode, and ensure compliance with Intel® CSE requirements.

5.1 Test Environment Setup

The System Under Test (SUT) is to be configured in manual configuration mode a with wired LAN or wireless LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured to ensure that the wired LAN and wireless LAN addresses reside on separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

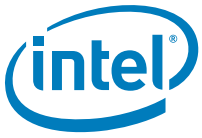
5.2 Tools for Testing

Intel® Flash Image Tool (fit.exe)

Intel® Flash Programming Tool (Intel® FPT) - is available in Windows* 32-bit (fptw.exe), Windows* 64-bit (fptw64.exe) operating systems, EFI 32-bit and EFI 64-bit.

Intel® Platform Enablement Test Suite (Intel® PETS) - Latest version of the tool is available in the Intel® CSE compliancy kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.

WinPE Tools: When using Windows* FW tools in WinPE, remember to load the MEI driver at every boot. This can be done by: *X:\Windows\System32>drvload.exe <path>\MEI.inf.* MEI.inf can be found in every FW kit release.



5.3 SPI Compliancy Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

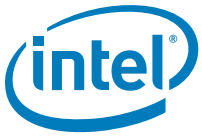
Test ID	Test Case Title	PETS Package Name	OS Supported	How?
SPI_01	Descriptor Mode Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_02	Serial Flash Discoverable Parameter Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_03	4 Kbytes Erasable Blocks Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_04	SFDP version 1.0 test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_05	SPI Flash Size Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_06	SPI Flash Vendor Specific Capabilities (VSCC) Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_07	Flash Descriptor Security Override Test	Compliance_SpiFlashConfiguration.xml	W	I

5.4 Descriptor Mode Test

Test ID:	**SPI_01**
Test Case Title:	Descriptor Mode Test
Platform:	LKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify the SPI flash controller in the SoC is operating in Descriptor Mode.
Test Pass Criteria:	Test passes if FDV bit is set to 1b.
Description:	Descriptor Mode is required for all SKUs of the SoC to ensure proper operation of features such as the Intel® CSE and SoC softstraps.
Procedure:	1. Boot to the target OS. 2. Verify the Flash Descriptor Valid bit is 14 in the Hardware Sequencing Flash Status (HSFS) register (SPIBAR + 04h) is set to 1b.

5.5 Serial Flash Discoverable Parameter Test

Test ID:	**SPI_02**
Test Case Title:	Serial Flash Discoverable Parameter (SFDP) Test
Platform:	LKFLKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify that the SPI flash controller in the SoC is able to detect a valid SFDP table in the SPI flash device.
Test Pass Criteria:	Test passes if all steps return expected values.
Description:	Proper SFDP support in the SPI flash device may be used to enable advanced SPI features like the Quad IO Fast Read.
Procedure:	1. Boot to target OS. 2. Does flash device 0 in the SUT supports SFDP? • If Yes, — Verify that the Component Property Parameter Table Valid (CPPTV) bit 31 of the Vendor Specific Component Capabilities 0 register (VSCC0[4]) is set to 1b. • If No, — Inform the test operator that SFDP support in the SPI flash device may be used to enable advanced SPI features like the Quad I/O Fast Read[3]. 3. Read the number of SPI parts by means of the Number of Components (NC) bits [9:8] in the Flash Map 0 (FLMAP0) register at (FDBAR + 14h). • If the number of components is 01b (2 Components) continue to next step else end test. 4. Does flash device 1 in the SUT supports SFDP? • If Yes, — Verify that the Component Property Parameter Table Valid (CPPTV) bit 31 of the Vendor Specific Component Capabilities 1 register (VSCC1[4]) is set to 1b. • If No, — Inform the test operator that SFDP support in the SPI flash device may be used to enable advanced SPI features like that Quad I/O Fast Read[3]. ***Notes:*** 1. VSCC0 register is located at (VTBA[4] + C4h). 2. VSCC1 register is located at (VTBA[4] + C4h + (n*8)h), where n=1. 3. Test considered pass, this is just additional information to user. 4. Refer to SPI Programming Guide for details of these registers.



5.6 4 Kbytes Erasable Blocks Test

Test ID:	SPI_03
Test Case Title:	4 Kbytes Erasable blocks Test
Platform:	LKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify the SPI flash device supports uniform 4 Kbytes erasable blocks.
Test Pass Criteria:	Test passes if all steps return expected values.
Description:	The SPI Flash device must provide uniform 4 Kbytes erasable blocks/sectors throughout the entire part. This is required by Intel® CSE firmware.
Procedure:	Part 1: Verify registers. 1. Boot to the target OS. 2. Verify the SUT is operating in Descriptor Mode by confirming that the Flash Descriptor Valid (FDV) bit 14 in the Hardware Sequencing Flash Status (HSFS) register (SPIBAR + 04h) has been set to '1'. 3. Verify all flash components support 4 Kbytes erasable blocks by confirming that the Block/Sector Erase Size (BERASE) bits [4:3] in the Hardware Sequencing Flash Status (HSFS) register (SPIBAR + 04) are set to 01b. Part 2: Check against SPI flash device datasheet. 1. Using the "MEInfo"[1] tool, read the SPI flash device ID from the SUT. 2. Verify the SPI flash device ID(s) read from the SUT are found in the vsccommn.bin[2] SPI part registry cached in Intel® PETS. ***Notes:*** 1. The "MEInfo" tool is part of the Intel® Converged Security Engine Firmware release package, under System Tools folder. 2. The vsccommn.bin file will be updated relative to the latest official version for each Intel® PETS release.

5.7 SFDP Version 1.0 Test

Test ID:	SPI_04
Test Case Title:	SFDP version 1.0 and above test
Platform:	LKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify SPI part is LakeField SFDP requirement compliant. This is SPI requirement to have minimum SFDP of v1.0. LKF SoC requires SPI flash devices support JEDEC standard JESD216 SDFDP v1.0 (Serial Flash Discoverable Parameters). Revision A (JESD216A - v1.5) or later is strongly recommended but not mandatory.
Test Pass Criteria:	SFDP version is 1.0 or above.
Description:	Intel SoC SKUs each have different requirements for SPI flash. This test verifies that the SPI flash device used meets the minimum SFDP version (1.5) required to use for LakeField platform.
Procedure:	Refer to the diagram and the table below for more details. 1. Locate the SFDP table in the SPI part 2. Read hex byte address offset 0x04, SFDP Minor Revision [7:0] and SFDP Major Revision [15:8] 3. Ensure Major.Minor revision is 1.0 or above

Figure 5-1. SFDP Mapping Diagram 1

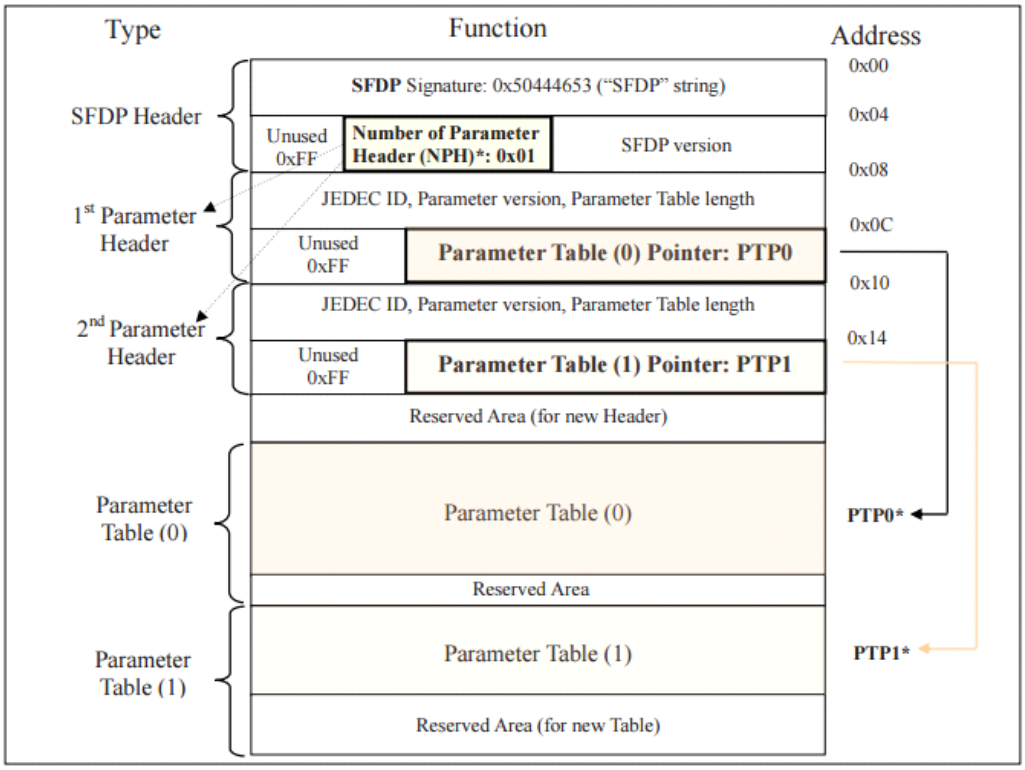


Figure 5-2. SFDP Mapping Diagram 2

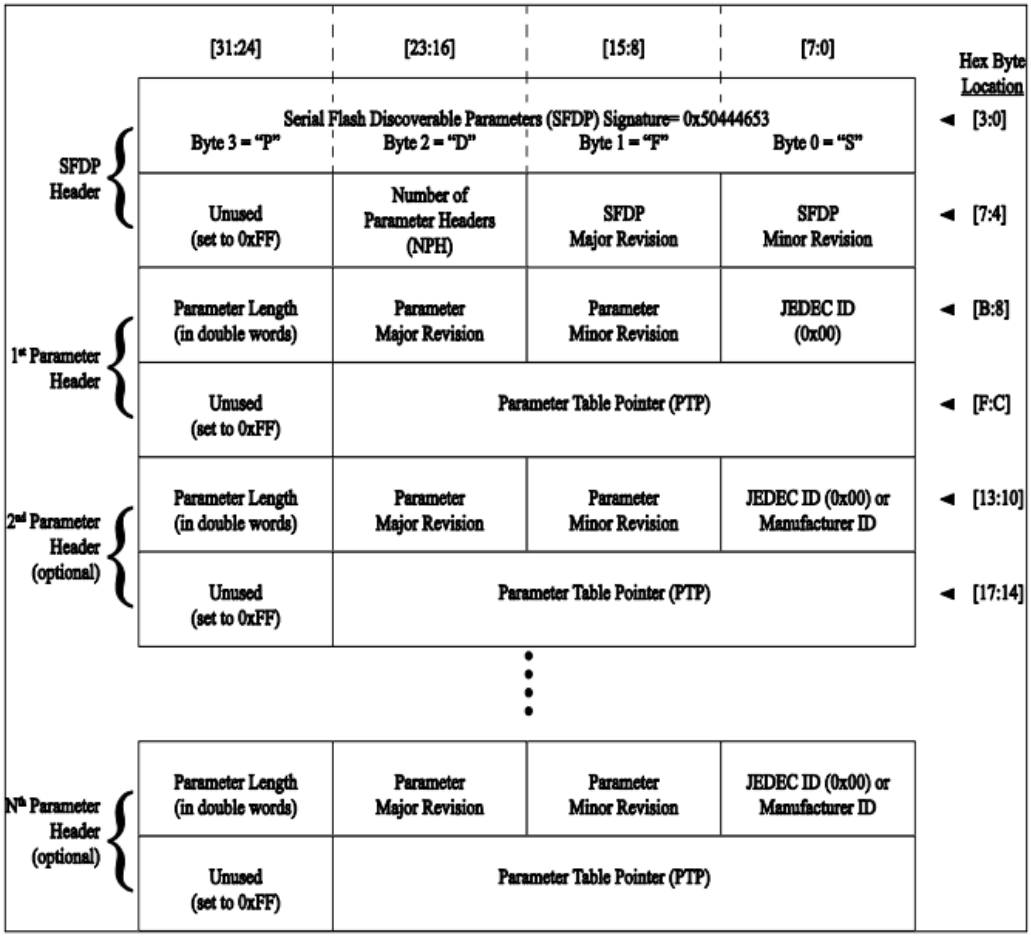


Table 5-1. SFDP Parameter Table Definition and Content

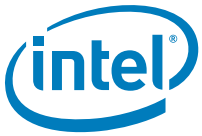
Table	Length	Maker	Definition Function
Parameter Table(0)	9 DWords	JEDEC STD	Sector Size, BP bits type
			4K Erase opcode
			Read mode, Address mode, DTR mode
			Flash density
			Read mode interface, Mode bits, dummy cycle
			Sector Size, Sector erase opcode
Parameter Table (1)	4 DWords	[SPI Vendor defines]	Voltage range
			Reset#, Hold# pin, Deep Power Down, SW Reset function, SuspendResume, Wrap-Around read
			Security function

5.8 SPI Flash Size Test

Test ID:	**SPI_05**
Test Case Title:	SPI Flash Size Test
Platform:	LKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify the correct SPI flash size is used for a given SoC SKU contained in the SUT.
Test Pass Criteria	The test passes if the following conditions is true: 1. The flash components' sizes in the SUT are equal to the size stated in the SPI device manufacturer datasheet.
Description:	Intel SoC SKUs each have different requirements for SPI flash sizes. This test verifies that the SPI flash device has enough space to store the whole SPI image created by Intel® FIT tool.
Procedure:	1. Boot to target OS. 2. Read following information from SPI Flash Descriptor in the SUT: a. The number of SPI parts by means of the Number of Components (NC) bits [9:8] in the Flash Map 0 (FLMAP0) register at (FDBAR + 14h). b. The size of the first flash component by means of the "Flash Density" in SFDP table above in Parameter Table (0) c. If the number of components is 01b (2 Components), read the size of the second flash component by means of the "Flash Density" in SFDP table above in Parameter Table (0) 3. Compare the SUT flash size against the: a. SPI flash device manufacturer datasheet[1]. ***Note:*** 1. Intel® PETS will maintain a list of SPI flash device sizes.

5.9 SPI Flash Vendor Specific Capabilities (VSCC) Test

Test ID:	**SPI_06**
Test Case Title:	SPI Flash Vendor Specific Component Capabilities (VSCC) Test.
Platform:	LKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	To verify VSCCn registers in memory mapped space and VSCC table in SPI Flash Descriptor is configured correctly.

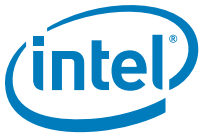


Test ID:	SPI_06
Test Pass Criteria:	Test results pass if VSCC0 or VSCC0 and VSCC1, and the VCSS table in SPI Flash Descriptor align with the Intel® CSE VSCC and SPI flash device manufacturer datasheet settings.
Description:	The VSCC registers are defined in two places. Host-based VSCCn registers (for example, VSCC0 and VSCC1) in memory mapped space and the Intel® CSE VSCC Table in the SPI Flash Descriptor. Intel® CSE only uses the VSCC table in the SPI Flash Descriptor, while the memory map VSCCn registers are used by BIOS. The Intel® CSE VSCC table is created using the FIT tool by ODM/OEM, while the memory mapped VSCCn registers are programed by BIOS. Incorrect VSCCn registers configuration may affect SPI flash functionality and also may lead to premature flash device wear out.
Procedure:	1. Boot to the target OS. 2. Read the Vendor Specific Component Capabilities Registers (VSCCn), in the memory mapped space, where these register are located at (SPIBAR + C4h) and (SPIDBAR + C4h + (1 * 8)h) respectively. 3. Verify the VSCCn values with the SPI Flash device manufacturer datasheet. 4. Read the VSCC table from the SPI flash device on the target system. The base address of the table is located at offset (FDBAR1 + EFCh). The Intel® CSE VSCC Table Base Address (VTBA) and the Intel® CSE VSCC Table Length (VTL) are located at (FDBAR + EFCh). 5. Every record in the table is 2 DWORDs long, the first 32 bits contain the SPI flash device's JEDEC ID, and the following 32 bits represent its VSCC value. 6. Iterate through the VSCC table searching for the matching JEDEC ID of the SPI devices in use on the SUT and verify the associated VSCC values matches both the SPI flash device manufacturer datasheet and the Intel® CSE VSCC value. ***Note:*** FDBAR is located at address 0 of the SPI flash device chip select 0.

5.10 Flash Descriptor Security Override Test

Test ID:	**SPI_07**
Test Case Title:	Flash Descriptor Security Override Test
Platform:	LKF (SPI based boot Only)
Mandatory/Optional:	Mandatory
Objective:	This test is to verify the platform has the ability to enable and disable Intel® CSE manufacturing mode, and to be able to reprogram the entire SPI flash.
Test Pass Criteria:	Test passes if FDOPSS bit is set to '1' by default and set to '0' when intending to enter Intel® CSE Test Mode.
Description:	This boots the platform in Intel® CSE Test Mode. This gives the ability to override Flash descriptor permissions debug/repair depot environments. This must NOT be default behavior. Flash Descriptor Override (FDP) is GPIO_42. Check LakeField Platform Design Guide (PDG) document for more details (# 567247).
Procedure:	LKF RVP for FDO is mapped to jumper J7E2. 1. Boot platform with **NOT** having FDO asserted high. Verify that FDOPSS is set to '1'. FDOPSS is in MMIO space (SPIBAR + 0x4) bit 13 2. Boot platform with having FDO asserted high. Verify that FDOPSS is set to '0'. FDOPSS is in MMIO space (SPIBAR + 0x4) bit 13. This assertion of FDO can be with a jumper or through another external mechanism. Care should be taken to ensure that assertion of this mechanism to assert FDO **cannot** be done remotely. PETS helps automate testing of this capability. Perform the test by enabling "State after G3 to S5" at BIOS setting. **Alternate Procedure** 1. Configure the platform with Intel® CSE Firmware. 2. Use FPT -f to flash new image. This test should fail. 3. Use the physical jumper to override the protection (asserts FDO GPIO_42 high). Boot system from G3 state 4. Use FPT -f to flash new image. This test should now pass

§ §



(This page is intentionally left blank)

6 Universal Flash Storage (UFS)

The purpose of this chapter is to describe the tests required in order to verify the functionality and system compliance for Universal Flash Storage (UFS) platforms.

The chapter provides a high level overview of UFS, its use cases, and the tests required to pass this section.

Note: This chapter is relevant only for systems with UFS flash device (as boot device).

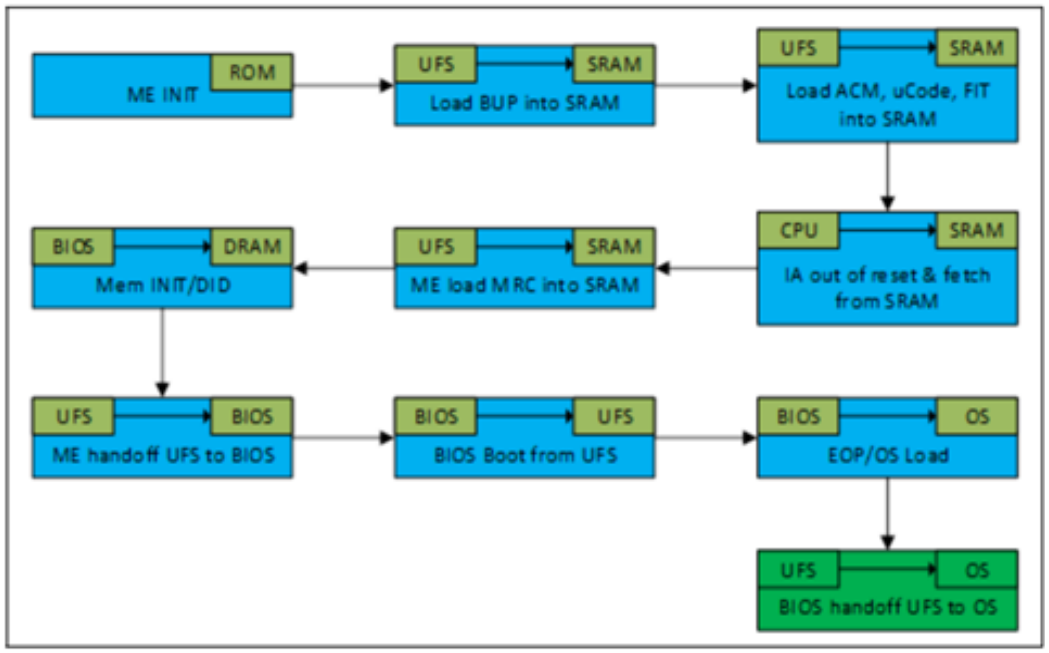
6.1 What is UFS?

UFS is a high-performance interface designed for use in applications where power consumption needs to be minimized, including mobile systems such as smart phones and tablets as well as automotive applications. Its high-speed serial interface and optimized protocol enable significant improvements in throughput and system performance.

Unlike SPI Flash which is used only to store boot FW (IFWI), UFS NVM is the main

storage on the platform. It stores OS, user files and Boot FW (IFWI). Hence UFS NVM is setup differently from SPI.

Figure 6-1. High level flow of booting from UFS



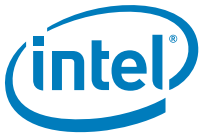


Figure 6-2. UFS Partitions

Partition	Size	Use	Comments
LUN0	-	OS user space	Size depends upon the size of the UFS Device
LUN1	32MB	Boot0 (IFWI)	Raw partition - Vendor created
LUN2	32MB	Boot1 (Not used)	Raw partition - Vendor created
LUN3	8MB	Platform factory data partition	Vendor created; Post factory - Read Only partition
LUN4	-	Not Used	OEM Choice
LUN5	-	Not Used	OEM Choice
LUN6	4MB	Temporary CSME data store to delay the RPMB key enrollment to end of manufacturing"	Raw partition - Vendor created
RPMB	4MB	Replay protected partition: Provisioned by Intel® ME and used for ME file system, PTT storage and UEFI variable storage	Vendor created - Provisioned with platform specific key by ME at EOM

Size recommendation for UFS partitions is based on Intel RVP system. Customer should evaluate their own requirements and allocate their partitions accordingly

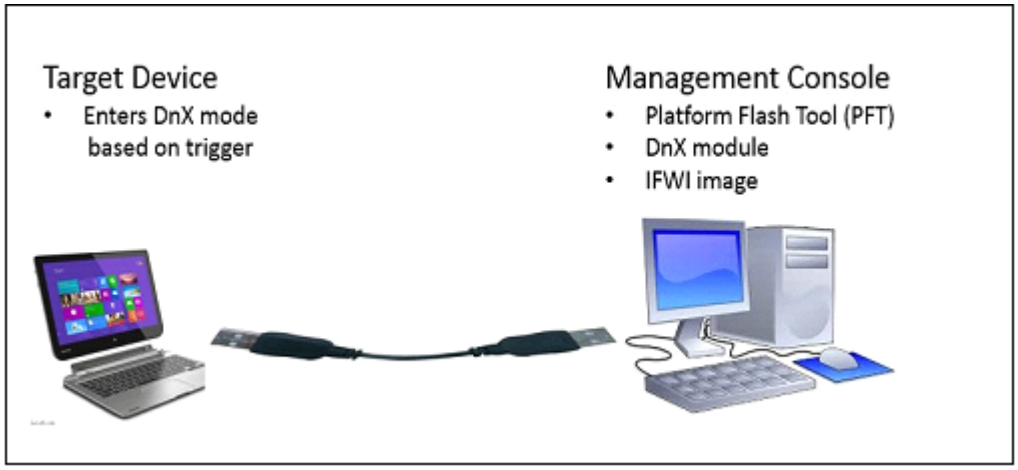
For the purpose of compliance, LUN0, LUN1, LUN6 and the RPMB partitions are the primary targets for testing. Other partitions are solely used for Vendor or OEM usage.

6.2 UFS Purpose and Detection

Primary use cases for UFS:

- Blank UFS Check
  - — Baseline: System has not yet been programmed with IFWI content.
  - — Diagnosis: Intel® CSE automatically identifies the condition and enters DnX state.
  - — Using DnX, user can flash a new image to the UFS and allow platform to boot up.
- Partitioning Capabilities
  - — Checking for creation and size limitations.
- Repartitioning Capabilities
  - — Ability to repartition over active partitions.
- Write new IFWI into UFS Boot partition
- LUN6 to RPMB after EOM
  - — Verifying UFS data is moved from Temporary Data Partition (LUN6) to RPMB after EOM has been sent.

6.3 Test Environment Setup and Tools



In order to complete this section, the following setup is required:

1. Target system with UFS programmable device.
2. DnX Image (IFWI image with DnX signed manifest) to be flashed on the device under test (can be created by Intel®FIT tool). Note: after EOM, flashing of unsigned IFWI will not be accepted by DnX Module.
3. Recovery/Management system – A host that can be used to execute the Intel® Platform Flash Tool (Intel® PFT).
4. USB cable connection between the recovery host to the System Under Test. USB cable should be connected to USB port0 of the SUT.
5. Intel® Platform Flash Tool (PFT) software - available in the Intel® CSE FW kit.
6. Configuration File (CFGFILE.XML) - Required for partitioning UFS device. This file is provided as input to the Intel® PFT tool and is included in the Intel® CSE Kit for LKF platform. OEMs are expected to update this file according to LUN partition numbers and size they are going to use for UFS device.
7. DnX_Module.bin – see details on usage of this file in UG. This binary runs on Management Console/Host System as an input to Intel® PFT Tool and available as a part of Intel® CSE FW Kit\Image Component\DnX*.bin.

6.4 UFS Compliance Test Coverage Summary

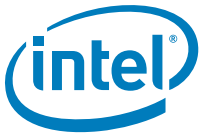
Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How is the test performed?

A = Fully Automated using Intel® PETS
I = Interactive using Intel® PETS
M = Manual

Test ID	Test Case Title	PETS Package Name	OS Supported	How?



UFS_01	Blank UFS Check	Compliance_UFS_Blank.xml	W	I
UFS_02	Partition Size allocation/verification	Compliance_UFS_Blank.xml	W	I
UFS_03	Re-Partition Check for UFS	Compliance_UFS_Blank.xml	W	I
UFS_04	Write new IFWI to UFS Boot partition	Compliance_UFS_Blank.xml	W	I
UFS_05	Data migration from LUN6 to RPMB at EOM	Compliance_UFS_OSProvisioned.xm	W	I

- All the Test cases below use Intel® PFT Tool. To keep unified format, all the test cases use command line interface of Intel® PFT.
- On the platforms that went through EOM: Get UFS Info (getcardinfo), Partitioning (configpart), Recovery (downloadfwos), Data Clear (clearrpmb) and Reading of the boot media (readbootmedia), DnX operations must be authorized by OEM. Authorization is done through OEM signed token.

  Flow to open DnX capabilities closed at EOM:

1. Use PFT to generate new OEM unlock token with active "DnX Capabilities" knob. Set value of this knob according to the desired DnX capability. For more details on how to generate a token refer to "LKF Secure Tokens Guide".
2. Sign OEM unlock token using PFT. To see more details on how to sign a token refer to "LKF Signing and Manifesting Guide". Place signed token file inside the PFT folder.
3. Prepare signed OEM KM and locate it inside the PFT folder. OEM KM should contain:
   a. Public key hash of private key used for signing OEM unlock token.
   b. Public key hash of private key used for signing DnX Image.
4. Using PFT run DnX command to download OEM KM to the SRAM
   .\dnxFwDownloader.exe --command **downloadoemkeymanifest** --key .\OEM_KM.bin --fw_dnx .\DNXP_0x1.bin
5. Using PFT run DnX command to set capabilities in OEM unlock token
   .\dnxFwDownloader.exe --command **setcapabilities** --capabilities .\OEM_Unlock-Token.tok --fw_dnx .\DNXP_0x1.bin
6. Now can run DnX command that was previously closed with EOM. Token is valid until end of the DnX session during which it was set (using set capabilities command).

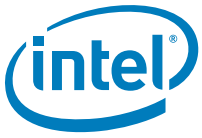
6.5 Blank UFS Check

Test ID:	**UFS_01**
Test Case Title:	Intel® CSE ROM to detect blank UFS and puts system into DnX mode.
Platform:	LKF (UFS based boot Only)
Mandatory/Optional:	Mandatory. ***Note:*** Test procedure is identical to test case DnX_01 and can be skipped if DnX_01 was already executed.
Objective:	Intel® CSE ROM detects blank UFS and triggers DnX mode and puts system into DnX mode.

Test ID:	UFS_01
Test Pass Criteria:	System Under Test goes to DnX mode; DnX device detected under Device Manger->USB stack.
Description:	This test will enumerate USB connection between SUT and Host system; Intel® CSE ROM will auto detect blank UFS and put system into DnX mode.
Procedure:	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Connect the SUT to a management console using a USB cable 3. Make sure that the Intel® Platform Flash Tool (Intel®PFT) is available on the management console. **Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode ***Note:*** In order to identify platform has entered DnX mode, check Device Manager ->USB should list DnX device. 3. On the management console, go into Intel®Platform Flash Tool (Intel® PFT) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and run DnX ID command: .\dnxFwDownloader.exe --command **iddevice**
Expected Response:	1. Check "ID DEVICE procedure" is success 2. Check "Response flags". Refer to DnX trigger table: • 0000b: HW Strap • 0010b: Bad NV content or Virgin Part • 0100b: Triggered by BIOS (BIOS config or error in BIOS loading) • 0101b: Unknown trigger (when running iddevice in Module context, since only ROM knows the trigger) • Others: Reserved **Example of expected response:** 08/20/18 10:17:02.151 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 10:17:02.156 DEBUG : Running command 'iddevice' on 08/20/18 10:17:02.157 INFO : Flags: 0 08/20/18 10:17:02.157 INFO : Starting ID DEVICE procedure 08/20/18 10:17:03.497 INFO : ID DEVICE procedure success 08/20/18 10:17:03.499 INFO : Response flags: 2 08/20/18 10:17:03.501 INFO : OEM platform ID: 0 08/20/18 10:17:03.508 INFO : Unique platform ID: 6268d2ce7f7b463d9d3243aba52e5d2a 08/20/18 10:17:03.512 INFO : Image errors: 00 00 00 00 00 00 00 00 00 00 00 00 ***Note:*** Boot will be on halt and system will be in DnX mode as there is no IFWI or OS image on the UFS device.

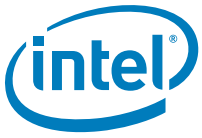
6.6 Partition Size Allocation/Verification

Test ID:	UFS_02
Test Case Title:	UFS partition creation and size allocation on blank UFS.
Platform:	LKF (UFS based boot Only)
Mandatory/Optional:	Mandatory ***Note:*** Test procedure is identical to test case DnX_02 and can be skipped if DnX_02 was already executed.
Objective:	Partitions of variable sizes can be created on blank UFS., including zero size partition.



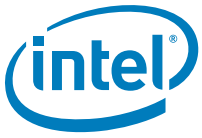
Test ID:	UFS_02
Test Pass Criteria:	LUN partitions with correct size are created on blank UFS successfully.
Description:	Test will partition blank UFS and verify correct configuration (number of LUNs, their size and attributes) using DnX
Procedure (1/2):	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Connect the SUT to a management console using a USB cable 3. Make sure that the PFT (Platform Flash Tool) is available on the management console. 4. Make sure DnX module available on the management console and located inside PFT folder 5. Make sure configuration file (cfgpart.xml indicating number of LUNs and their size) is available on the management console and located inside PFT folder **Cfgpart.xml creation:** The DNX tool can set the UFS's LUNs descriptor. Each LUN descriptor appears in the XML with the following lines: <lun idx="0"> <enable>true</enable> <boot-lun-id>0x1</boot-lun-id> <write-protect>0x0</write-protect> <mem-type>0x0</mem-type> <alloc-units>0x1000000</alloc-units> < data-reliability>0x0</data-reliability> <logical-block-size>0xc</logical-block-size> <provisioning-type>0x0</provisioning-type> <ctx-caps>0x0</ctx-caps> </lun> For basic usage, it is recommended to only change the enable, boot-lun-id and alloc-units lines. If an LUN isn't defined in the XML, it will be set as disabled, so need to define all the LUNs that need to be enabled. <alloc-units> is in endian swapped format. Some examples: 0x1000000 = 4MB 0x2000000 = 8MB 0x8000000 = 32MB 0x10000 = 1GB 0x100 = 256GB ***Note:*** Sample xml file, with size based on RVP recommendation, is located inside the kit. User/OEM is expected to update size per their requirement before creating partitions

Test ID:	UFS_02
Procedure (2/2):	**Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode ***Note:*** In order to identify platform has entered DnX mode, check Device Manager ->USB should list DnX device. 3. On the management console, go into PFT (Platform Flash Tool) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool". 4. Open command prompt and run DnX commands as following: a. Configuration command: .\dnxFwDownloader.exe --command **configpart** --fw_dnx .\DNXP_0x1.bin --path cfgpart.xml --device ufs --idx 0 b. Get Card Info command: .\dnxFwDownloader.exe --command **getcardinfo** --fw_dnx .\DNXP_0x1.bin --idx 0 --device ufs ***Note:*** Those commands require OEM authorization (using DnX Token) post EOM. See instructions above in this document. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.



Test ID:	UFS_02
Expected Response(1/2):	1. Check Response flags is 0 (0=success) 2. Check LUNs size Check "GET CARD INFO procedure success" ***Note:*** LUN1 must be enabled (bBootLunID is 0x1) with minimum size of 32MB and bLogicalBlockSize set to 0xc which is 4K blocks. This is the only configuration supported by Intel® CSE ***Note:*** bRefClkFreq must be 0, platform would not boot otherwise **Example of expected response:** 08/20/18 10:34:45.186 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 10:34:45.189 DEBUG : Running command 'getcardinfo' on 08/20/18 10:34:45.190 INFO : DnX module version: 1245 08/20/18 10:34:45.192 INFO : Device type: ufs 08/20/18 10:34:45.193 INFO : Device index: 0 08/20/18 10:34:45.194 INFO : Starting GET CARD INFO procedure 08/20/18 10:34:46.265 INFO : === UFS device info === --- Descriptor --- bBootEnable: [0x1] bDescAccessEn: [0] bInitPowerMode: [0x1] bHighPriorityLUN: [0x7f] bSecureRemovalType: [0] bInitActiveICCLevel: [0] wPeriodicRTCUpdate: [0] --- LUN [0] --- bLUEnable: [0x1] bBootLunID: [0] bLUWriteProtect: [0] bMemoryType: [0] dNumAllocUnits: [0x1b0000] bDataReliability: [0] bLogicalBlockSize: [0xc] bProvisioningType: [0x3] wContextCapabilities: [0] --- LUN [1] --- bLUEnable: [0x1] bBootLunID: [0x1] bLUWriteProtect: [0] bMemoryType: [0] dNumAllocUnits: [0x8000000] bDataReliability: [0] bLogicalBlockSize: [0xc] bProvisioningType: [0] wContextCapabilities: [0] ... --- LUN [7] --- bLUEnable: [0] bBootLunID: [0] bLUWriteProtect: [0] bMemoryType: [0] dNumAllocUnits: [0] bDataReliability: [0] bLogicalBlockSize: [0] bProvisioningType: [0] wContextCapabilities: [0]

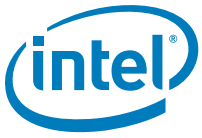
Test ID:	UFS_02
Expected Response(2/2):	--- Attributes --- AttributesEnables: [0] bRefClkFreq: [0] bBootLunEn: [0x1] bCurrentPowerMode: [0x11] bActiveICCLevel: [0] bOutOfOrderDataEn: [0] bMaxDataInSize: [0x40] bMaxDataOutSize: [0x40] bConfigDescrLock: [0] bMaxNumOfRTT: [0x2] wExceptionEventControl: [0] dSecondsPassed: [0] --- Geometry Descriptor --- bLength: [0x48] bLength: [0x48] bDescriptorType: [0x7] bMediaTechnology: [0] qTotalRawDeviceCapacity: [124993536] dSegmentSize: [2097152] bAllocationUnitSize: [1] bMinAddrBlockSize: [8] bOptimalReadBlockSize: [8] bOptimalWriteBlockSize: [8] bMaxInBufferSize: [64] bMaxOutBufferSize: [64] bRPMBReadWriteSize: [64] bDataOrdering: [0] bMaxContextIDNumber: [0x5] bSysDataTagUnitSize: [0] bSysDataTagResSize: [0] bSupportedSecRTypes: [0x9] wSupportedMemoryTypes: [0xf80] dSystemCodeMaxNAllocU: [0x9a3b0000] wSystemCodeCapAdjFac: [0x1] dNonPersistMaxNAllocU: [0x9a3b0000] wNonPersistCapAdjFac: [0x1] dEnhanced1MaxNAllocU: [0x9a3b0000] wEnhanced1CapAdjFac: [0x2] dEnhanced2MaxNAllocU: [0] wEnhanced2CapAdjFac: [0] dEnhanced3MaxNAllocU: [0] wEnhanced3CapAdjFac: [0] dEnhanced4MaxNAllocU: [0] wEnhanced4CapAdjFac: [0] GET CARD INFO procedure success



6.7 Re-Partition Check for UFS

Test ID:	UFS_03
Test Case Title:	Re-partition blank data over existing UFS Flash Partition Region LUNx.
Platform:	LKF (UFS based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify that UFS partition region is re-programmable.
Test Pass Criteria:	Re-partitioning command on already configured UFS succeeds and new configuration takes place

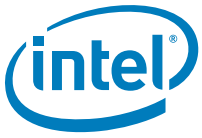
Test ID:	UFS_03
Description:	This test will write over existing UFS partition and verify no failure.
Procedure (1/2):	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Make sure UFS was already partitioned, refer to UFS_02 / DnX_02 for details. 3. Connect the SUT to a management console using a USB cable 4. Make sure that the PFT (Platform Flash Tool) is available on the management console. 5. Make sure DnX module available on the management console and located inside PFT folder 6. Make sure configuration file (cfgpart.xml indicating number of LUNs and their size) is available on the management console and located inside PFT folder **Cfgpart.xml creation:** The DNX tool can set the UFS's LUNs descriptor. Each LUN descriptor appears in the XML with the following lines: <lun idx="0"> <enable>true</enable> <boot-lun-id>0x1</boot-lun-id> <write-protect>0x0</write-protect> <mem-type>0x0</mem-type> <alloc-units>0x1000000</alloc-units> < data-reliability>0x0</data-reliability> <logical-block-size>0xc</logical-block-size> <provisioning-type>0x0</provisioning-type> <ctx-caps>0x0</ctx-caps> </lun> For basic usage, it is recommended to only change the enable, boot-lun-id and alloc-units lines. If an LUN isn't defined in the XML, it will be set as disabled, so need to define all the LUNs that need to be enabled. <alloc-units> is in endian swapped format. Some examples: 0x1000000 = 4MB 0x2000000 = 8MB 0x8000000 = 32MB 0x10000 = 1GB 0x100 = 256GB ***Note:*** Sample xml file, with size based on RVP recommendation, is located inside the kit. User/OEM is expected to update size per their requirement before creating partitions



Test ID:	UFS_03
Procedure (2/2):	**Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode 3. On the management console, go into PFT (Platform Flash Tool) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool". 4. Open command prompt and run DnX commands as following: a. Get Card Info command: .\dnxFwDownloader.exe --command **getcardinfo** --fw_dnx .\DNXP_0x1.bin --idx 0 --device ufs Save results (check expected results in test case UFS_02) b. Open config file from test case UFS_02 and change one or more LUNs <alloc-units> to different size (use examples above). Better to decrease the size not to overstep UFS total size c. Run Configuration command: .\dnxFwDownloader.exe --command **configpart** --fw_dnx .\DNXP_0x1.bin --path cfgpart.xml --device ufs --idx 0 d. Get Card Info command: .\dnxFwDownloader.exe --command **getcardinfo** --fw_dnx .\DNXP_0x1.bin --idx 0 --device ufs Compare output to the output saved in step 'a' and check that differences are as expected e. Return config file as in test case UFS_02 f. Run Configuration command: .\dnxFwDownloader.exe --command **configpart** --fw_dnx .\DNXP_0x1.bin --path cfgpart.xml --device ufs --idx 0 g. Get Card Info command: .\dnxFwDownloader.exe --command **getcardinfo** --fw_dnx .\DNXP_0x1.bin --idx 0 --device ufs Compare output to the output saved in a) and make sure they are the same ***Note:*** Those commands require OEM authorization (using DnX Token) post EOM. See instructions above in this document. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.

6.8 Write new IFWI to UFS Boot partition

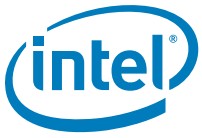
Test ID:	**UFS_04**
Test Case Title:	Write new image to UFS Flash Boot Partition
Platform:	LKF (UFS based boot Only)
Mandatory/Optional:	Mandatory ***Note:*** Test procedure is identical to test case DnX_03 and can be skipped if DnX_03 was already executed.
Objective:	Verify that UFS partition region LUN1 is programmable.
Test Pass Criteria:	Image was successfully flashed



Test ID:	UFS_04
Description:	SUT will enter DnX mode and use its capabilities to flash provided Image into UFS
Procedure:	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Make sure that UFS in the SUT was partitioned and Boot partition size > IFWI size, refer to DnX_02 or UFS_02 for details. 3. Connect the SUT to a management console using a USB cable 4. Make sure that the Platform Flash Tool (PFT) is available on the management console. 5. Make sure DnX module available on the management console and located inside PFT folder. 6. Make sure that the image to be flashed to the SUT, stored in the PFT directory. Refer to Bring Up Guide for instructions on how to create DnX Image. **Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode 3. On the management console, go into PFT (Platform Flash Tool) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and run following commands: a. Download Image command: .\dnxFwDownloader.exe --command **downloadfwos** --fw_dnx .\DNXP_0x1.bin --fw_image .\Image_DNX.bin --flags 0 ***Note:*** This command requires OEM authorization (using DnX Token) post EOM. See instructions above in this document. b. Device Reset command: .\dnxFwDownloader.exe --command **startover** --flags 9 ***Note:*** This is optional, issuing this command will provide remote reset followed by full boot. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.
Expected Response:	1. Check "DOWNLOADFWOS procedure" is success **Example of expected response:** 08/20/18 14:42:04.067 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 14:42:04.078 DEBUG : Running command 'downloadfwos' on 08/20/18 14:42:04.082 INFO : DnX module version: 1245 08/20/18 14:42:04.084 INFO : Starting DOWNLOADFWOS procedure 08/20/18 14:42:06.506 INFO : DOWNLOADFWOS procedure success 2. Check "STARTOVER procedure" is success **Example of expected response:** 08/20/18 14:44:46.935 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 14:44:46.939 DEBUG : Running command 'startover' on 08/20/18 14:44:46.940 INFO : Starting STARTOVER procedure 08/20/18 14:44:46.941 INFO : Flags: 10 08/20/18 14:44:48.001 INFO : STARTOVER procedure success 08/20/18 14:44:48.005 INFO : Current operation: 4 08/20/18 14:44:48.014 INFO : Current context: 1 3. Check that SUT boots normally w/o going into DnX mode

6.9 Data Migration from LUN6 to RPMB at EOM

Test ID:	**UFS_05**
Test Case Title:	Data migration from LUN6 to RPMB at EOM
Platform:	LKF (UFS based boot Only)
Mandatory/ Optional:	Mandatory
Objective:	Verify that data has transferred to RPMB partition and temporary data partition (LUN6) was cleared.
Test Pass Criteria	After EOM: LUN6 is cleared and RPMB contains the data



Test ID:	UFS_05
Description:	Test will check that after EOM, data migrates from LUN6 to RPMB while LUN6 is getting cleared by Intel® CSE
Procedure:	**Test preparation:** 1. Connect the SUT to a management console using a USB cable 2. Make sure that the PFT (Platform Flash Tool) is available on the management console. 3. Make sure DnX module available on the management console and located inside PFT folder 4. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Complete DnX_03 (or UFS_04) test to ensure IFWI was programmed on the NVM 2. Open command prompt and run Read Boot Media command (this operation allows the tool to read contents of the FW from NVM): a. To read Temp Data Partition (LUN6): .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpLUN6_BeforeEOM.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **22** 3. Perform EOM. Refer to MFG_01 for more detailed instructions. Windows* example: FPT.exe –closemnf -y 4. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 5. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 6. Open command prompt and run Read Boot Media command (this operation allows the tool to read contents of the FW from NVM): a. To read Temp Data Partition (LUN6): .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpLUN6_AfterEOM.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **22** b. To read RPMB partition: .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpRPMB.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **48** ***Note:*** Make sure that output file location can be accessed for write, otherwise operation will fail. ***Note:*** Make sure to set correct number of blocks to read, based on the partition size. **1 block = 1kByte.** E.g. how to read 4MB LUN: 4MB = 4096kB (in binary) --> need to read 4096 blocks ***Note:*** This command requires OEM authorization (using DnX Token) post EOM. See instructions above in this document. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.

Test ID:	UFS_05
Expected Response:	1. Check "READ BOOT MEDIA procedure" is success 2. Check Partition index is correct (e.g. 48 - for RPMB) 3. Compare "dumpLUN6_BeforeEOM.bin" vs "dumpRPMB.bin" 4. Check that "dumpLUN6_AfterEOM.bin" is empty **Example of expected response:** 08/20/18 13:25:44.542 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 13:25:44.547 DEBUG : Running command 'readbootmedia' on 08/20/18 13:25:44.549 INFO : DnX module version: 1245 08/20/18 13:25:44.550 INFO : Device type: ufs 08/20/18 13:25:44.566 INFO : Device index: 0 08/20/18 13:25:44.581 INFO : Start offset: 0 08/20/18 13:25:44.584 INFO : Blocks to read: 4096 08/20/18 13:25:44.591 INFO : Partition index: 0 08/20/18 13:25:44.603 INFO : Starting READ BOOT MEDIA procedure 08/20/18 13:25:48.057 INFO : READ BOOT MEDIA procedure success ***Note:*** Returned data is in the 'raw' as read from the media and is not processed at all by DnX module (i.e. no decryption etc. is performed, rather all data is returned as stored on the media)

7 ISH FW and Platform Sensors Compliancy

This section provides the ISH FW testing from the image creating stage to OS level, in each stage checking the ISH FW and sensors status. It also provides the system sensors test cases about sensor noise, performance and accuracy. It provides ISH FW testing (performed using PETS) from the image creating stage to OS level, in each stage checking the ISH FW and sensors status.
Platform Enablement Test Suite (PETS) is a test design application that enables users to design and run work flows on various devices. It is used for sensor compliancy testing.

Prerequisites:

- Intel® PETS (Platform Enablement Test Suite)
- The PDT Editor tool can be found in the CSE/ISH FW Kit
- The Sensor Viewer Tool can be found in the ISH FW Kit
- The Sensor Diagnostic Tool is a part of the Windows* Driver Kit (WDK)

7.1 Intel® ISH FW and Platform Sensors Compliancy Test Coverage Summary

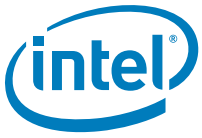
Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
ISS_TST_01	Sensor communication test	N/A	W	A
ISS_TST_02	Sensor data check	N/A	W	M
ISS_TST_03	ISH FW loading and execution	N/A	W	M
ISS_TST_04	Intel® SensorViewer test	N/A	W	M
ISS_TST_05	Test system sensor noise and effects on sensor algorithms	N/A	W	A
ISS_TST_06	Test worst case system interference and effect on sensor algorithms	N/A	W	A
ISS_TST_07	Test System Performance and Effective Calibration under a Specific Range of Movements	N/A	W	A
ISS_TST_09	Light Sensor (ALS) Accuracy Test	N/A	W	A
ISS_TST_10	Light Sensor (ALS) Angular Response Test	N/A	W	A
ISS_TST_11	360 Hinge and Swivel Accuracy Test with 2nd Accelerometer	N/A	W	A

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
ISS_TST_13	Heading Sensor Accuracy and Drift Test	N/A	W	A
ISS_TST_14	Intel® Integrated Sensor Solution Power States	N/A	W	A
ISS_TST_15	Sensor Activity Contexts	N/A	W	A
ISS_TST_16	Sensor Terminal Contexts	N/A	W	A
ISS_TST_17	Sensor Gesture Contexts	N/A	W	A
ISS_TST_18	Wake on Shake Test	N/A	W	M
ISS_TST_19	Step Counting Test	N/A	W	M



7.2 Sensor Communication Test

Test ID:	**ISS_TST_01**
Test Case Title:	Sensor communication test
Objective:	Verify communication with the ISH sensors
Test Pass Criteria:	Test passes if the ISSUtil test returns test completed successfully without any errors.
Description:	This test is checking basic communication with the ISH and ISH FW can be read.
Windows* Procedure:	• Boot the platform to WOS/EFI shell. • From elevated command line run the ISSUtil Tool: "ISSUtil.exe –BIST –test 0 –verbose" "ISSUtil.exe –BIST –test 1 –verbose" "ISSUtil.exe –BIST –test 2 –verbose" "ISSUtil.exe –BIST –test 3 –verbose" • In the tool output, the test results for each of the tests should be completed successfully without any errors.

7.3 Sensor Data Check

Test ID:	**ISS_TST_02**
Test Case Title:	Sensor data check
Objective:	Check the sensor data in the PDT editor to make sure it is compliant with the board.
Test Pass Criteria:	Test passes if the sensors information was configured correctly in the PDT Editor.
Description:	In the PDT Editor, we are configuring the sensors drivers, I2C data and calibration data. This test checks that the sensors information was configured correctly in PDT table.
Windows* Procedure:	Verify the sensors information in the PDT Editor: • Open the full SPI image in the FIT tool. (Decompose it) • In the FIT tool folder, a folder will be created with the name of the image that was decomposed using FIT. • Using the PDT Editor open the PDT table from that image, it is located under: FIT\"image_name"\Decomp\PdtBinary.bin. • In the PDT Editor, verify that each of the sensors configured with the right settings.

7.4 ISH FW Loading and Execution

Test ID:	**ISS_TST_03**
Test Case Title:	ISH FW Loading and Execution
Objective:	Verify that ISH is responsive and that ISH FW can be read

Test ID:	ISS_TST_03
Test Pass Criteria:	Test will pass if ISH status is "responding" and ISH FW can be read.
Description:	This test is checking basic communication with the ISH and the ISH FW can be read.
Windows* Procedure:	• Boot the platform to WOS shell. • From elevated command line run the ISSUtil Tool: ISSUtil.exe -INFO • In the tool output check that: a. ISH Status is "responding" b. ISH FW Version can be read and is as follow: "5.x.x.XXXX" (X-Stand for do not care)

7.5 Intel® Sensor Viewer Test

Test ID:	ISS_TST_04
Test Case Title:	Intel® SensorViewer test
Objective:	Verify that the ISH sensors are ready for use and that data is received from the sensor
Test Pass Criteria:	Test will pass if in the Sensor Diagnostic Tool, all of the sensors state is "Ready" and the data is received for each of the sensors
Description:	This test is checking that the ISH sensors are ready for use
Windows* Procedure:	• Boot the platform to Windows* • Open the Sensor Diagnostic Tool. • For each sensor on the platform check that the state is "Ready" and that Data is received, this may require a trigger of the sensor event, for example for the Orientation sensor the platform need to be moved in order to receive data in the sensor Diagnostic Tool.

7.6 Sensor Noise and Error Levels

Included below is a table of sensor noise and error levels that will be monitored by some tests within the compliance guide. These numbers should be measured after calibration has been applied.

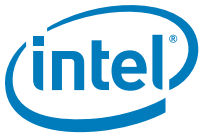


Table 7-1. Values Measured from the Physical Sensor

	Maximum Offset per Axis Compared to Average	Noise per Axis
Accelerometer	30 mg	10 mg
Magnetometer	50 mGauss	10 mGauss
Gyroscope	15 dps	0.2 dps

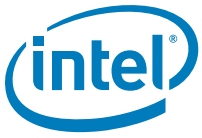
Table 7-2. Values Measured from the IISS Algorithms (Static - No Movement)

	Maximum Error	Average Error	STD
Inclinometer	2 degrees	2 degrees	0.75 deg
3D Compass	2 degrees	2 degrees	0.75 deg
3D Gyro	1.0 dps	1.0 dps	0.2 dps
3D Accelerometer	40 mg	40 mg	

3D Gyroscope and 3D Accelerometer values are “per axis

7.7 Test System Sensor Noise and Effects on Sensor Algorithms

Test ID:	**ISS_TST_05**
Test Case Title:	Test system sensor noise and effects on sensor algorithms
Mandatory/Optional:	Optional
Description:	The performance of the ISS sensor algorithms may degrade if the noise levels are too high. This test measures the noise levels on each sensor at when the system is at rest to indicate the likelihood of an impact to overall system sensor performance. The causes for higher noise levels can include selecting a poor quality sensor or could be related to system interference from other components (i.e. CPU) or due to PCB design issues. The test also measures any variance seen at the output of the sensor algorithms to also indicate unexpected variance (i.e. e-compass moving or drifting) that would also indicate a performance issue with the system.
Objective:	Gather statistical data on both sensor data input (RAW sensor data) and data output of sensor algorithms.
Procedure:	Automated (PETS) Initial state of the SUT should be S0. If the system is a 2-in-1 device, the test should start with the system in the "PC" context (screen facing user with keyboard facing-up on the table). **Intel® Platform Enablement Test Suite (Intel® PETS) will perform the following steps:** 1. Gather RAW and virtual sensor data over a designated period (i.e. 10s). Data will be gathered from all present physical sensors on platform and all available sensor SW drivers. 2. If the System is a 2-in-1 device, convert it into a tablet form-factor (screen on top of keyboard or detached from it_ and repeat step #1
Test Pass/Fail Criteria:	Test will pass if all sequences show: 1. RAW sensor statistical data shows noise levels within acceptable ranges. 2. Data output from sensor algorithms do not show movement or other performance issues when the system is at rest. ***Note:*** For #1 and #2 - the tool will refer to the pass/fail levels placed in the section "Sensor Noise and Error Levels". ***Note:*** In the case that the test results are above the pass/fail limits - the tests will raise a "warning" to the user.

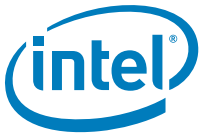


7.8 Test Worst Case System Interference and Effect on Sensor Algorithms

Test ID:	**ISS_TST_06**
Test Case Title:	Test worst case system interference and effect on sensor algorithms
Mandatory/Optional:	Optional
Description:	The system may contain noise sources that cause the worst system sensor performance issues when enabled. This can include the speakers, CPU, GPU, and others. The goal of this test is to measure both physical RAW sensor data and the outputs seen at the output of the sensor algorithms to understand if increased noise levels (or movement) is seen when typical noise sources are operated at their worst condition.
Objective:	Determine the worst-case system interference that can be seen on the sensors. Measures both interference seen on RAW sensor data and effect to virtual sensors.
Procedure:	Semi-Automated (PETS) Initial state of the SUT should be S0. The audio sub-system should be fully functional. If the system is a 2-in-1 device, the test should start with the system in the "PC" context (screen facing user with keyboard facing-up on the table). **Intel® Platform Enablement Test Suite (Intel® PETS) will perform the following steps:** 1. The system will exercise known interference sources to refer if they will have influences on the system. Data should be gathered at each step for at least 10 seconds. The interference sources include: • Outputting speaker data at maximum frequency with a tonal frequency of 100 Hz to 2000 Hz (100 Hz/step). This should be operated at maximum volume. • CPU operated at minimum and maximum load. • GPU operated at minimum and maximum load. • Turn the computer screen on/off. For each sample data sample - the system will gather RAW and virtual sensor data. The noise levels and any movement should be recorded and compared to pass/fail levels. 2. The system will exercise known interference sources to refer if they will have influences on the system. Data Should be gathered at each step. 3. If the system is a 2-in-1 device, convert it into a tablet form-factor (detached/screen on to of keyboard) and repeat steps #1 and #2
Test Pass/Fail Criteria:	Test will pass if all sequences show: 1. RAW sensor statistical data shows noise levels within acceptable ranges. 2. Data output from sensor algorithms do not show movement or other performance issues when the system is at rest. ***Note:*** For #1 and #2 - the tool will refer to the pass/fail levels placed in the section "Sensor Noise and Error Levels". ***Note:*** In the case that the test results are above the pass/fail limits - the tests will raise a "warning" to the user**.**

7.9 Test System Performance and Effective Calibration under a Specific Range of Movements

Test ID:	**ISS_TST_07**
Test Case Title:	Test system performance and effective calibration under a specific range of movements
Mandatory/Optional:	Optional. Mandatory if motion sensors are present
Description:	The data quality of the sensor algorithms can be impacted by a number of factors (e.g. inaccurate sensor calibration). This test moves the sensor across a number of positions and tests that all pass-through sensors and virtual algorithms respond as expected.
Objective:	Tests sensor configuration for correct orientation and data during both rest and movement.
Procedure:	Semi-Automated (PETS) Initial state of the SUT should be S0. The system should have run through the ISS sensor calibration procedure with the calibration data stored and used on the system. The system should be configured in a tablet context. If the device is a 2- in-1, suggest repeating in the PC form-factor with the system placed in a box that can be moved in the pattern shown below. The user will be asked to run through the following movements to test the gyroscope: **Test Sub-Section A: Gyroscope Z-Axis:** 1. Place the system flat on the table with the screen facing upwards. 2. Rotate the system clockwise - the gyroscope should identify a negative angular velocity on the Z-axis. 3. Rotate the system counter-clockwise - the gyroscope should identify a positive angular velocity on the Z-axis. **Test Sub-Section B: Gyroscope X-Axis:** 1. Place the system face-up on the table with the screen facing towards in the "portrait" position. 2. Rotate the system clockwise - the gyroscope should identify a positive angular velocity on the Y-axis. 3. Rotate the system counter-clockwise - the gyroscope should identify a negative angular velocity on the Y-axis. **Test Sub-Section C: Gyroscope Y-Axis:** 1. Place the system face-up on the table with the screen facing towards in the "landscape" position. The right-hand side of the screen should be pointing upwards. 2. Rotate the system clockwise - the gyroscope should identify a negative angular velocity on the X-axis. 3. Rotate the system counter-clockwise - the gyroscope should identify a positive angular velocity on the X-axis. **Test Sub-Section D: Accelerometer:** Place the system in the following positions: 1. Flat on the table facing up. (Z-UP) The accelerometer should read (0,0,-g0). 2. Flat on the table facing down. (Z-down) The accelerometer should read (0,0,g0). 3. Facing the user on the table in landscape mode. (X-DOWN) The accelerometer should read (g0,0,0). 4. The same position as the previous step but now placed up-side-down. The accelerometer should read (-g0,0,0). 5. Facing the user on the table in portrait mode. (Y-DOWN) The accelerometer should read (0,-g0,0). 6. The same position as the previous step but now placed up-side-down. The accelerometer should read (0,g0,0).
Test Pass/Fail Criteria:	Test will pass if all sequences show: For the gyroscope: The correct direction was recorded from the gyroscope when moving the system. For the accelerometer: The accelerometer reading was correct within a 5 degree error.



7.10 Light Sensor (ALS) Accuracy Test

Test ID:	**ISS_TST_09**
Test Case Title:	Light sensor (ALS) accuracy test
Mandatory/Optional:	Mandatory
Description:	This test will review the accuracy of the ambient light sensor after it has been characterized.
Objective:	The Ambient Light Sensor accuracy may be affected by a number of factors including the mechanical design of the housing, cover glass, and the calibration applied within the ISS system. The test is meant to test the accuracy of the ALS after it has been calibrated.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. System is in a dark room or placed within a lighting tent (made out of diffused lighting material) and covered with a black cloth. The following equipment should be used: 1. Tunable light source that can emit halogen light. 2. Light meter to measure the lighting level incident on the SUT. The light meter is placed next to the system ALS sensor. The system should be orientated orthogonal to the light source. 3. Light source is tuned to maximum amplitude. ALS reading should be displayed on the screen. Check that the received ALS value is within +/- 10% of the recorded light meter value. The screen brightness should appear not too bright or too dark. 4. Lower the light source to mid-way. Compare again the difference between the ALS and light meter value. The screen brightness should adjust such that it is not too bright or too dark relative to the ambient light level. 5. Tune light source to the lowest level. Compare again the difference between the ALS and light meter value. The screen brightness should adjust such that it is not too bright or too dark relative to the ambient light level. 6. (optional) If a fluorescent light source is available, expose the system to the same "low" light level seen in the previous step. Check that the ALS light levels are correct relative to the light meter. And that the screen brightness is not too bright or too dark.
Test Pass/Fail Criteria:	Test will pass if all sequences show: For all light levels tested - the ALS is correct within +/- 10%.

7.11 Light Sensor (ALS) Angular Response Test

Test ID:	**ISS_TST_10**
Test Case Title:	Light sensor (ALS) angular response test
Mandatory/Optional:	Mandatory
Description:	This test will test the angular response of the ALS sensor to determine if it will fall within the requirements of the MSFT HW certification guidelines. MSFT asks that the light response does not fall by more than 50% when changing the angle of incident light from 0 to 35 degrees. Issues can occur with the sensor angular response due to the light sensor cavity/hole design or other materials covering the light sensor.

Test ID:	ISS_TST_10
Objective:	Confirm that the ambient light sensor angular response is greater than 50% at a 35 degree angle of incidence.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW.System is in a dark room or placed within a lighting tent (made out of diffused lighting material) and covered with a black cloth. The following equipment should be used: 1. Tunable light source that can emit halogen light. 2. Light meter to measure the lighting level incident on the SUT. The light meter is placed next to the system ALS sensor. The system should be orientated orthogonal to the light source. Before starting the test: 1. The system should be directly facing the light source. 2. The ALS reading should be within +/- 10% of the value read by the light meter. Recommended target lighting is 100lux with the ALS reading 90-110 lux. When running the test: 1. Rotate the system so that the ALS is at a 35 degree angle to the incident light without changing the distance.
Test Pass/Fail Criteria:	Test will pass if all sequences show: The recorded light level of the ALS does not fall more than 50%.

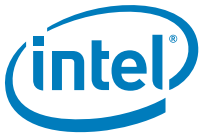
7.12 360 Hinge and Swivel Accuracy Test with 2nd Accelerometer

Test ID:	ISS_TST_11
Test Case Title:	360 hinge and swivel accuracy test with 2nd accelerometer
Mandatory/Optional:	Required only if the 2nd accelerometer is present on the design.
Description:	Placing an accelerometer both in the base and lid of the system design will enable the system to determine the angle between the lid and base. This algorithm (also called a virtual protractor) will tell the system how to operate if the system is closed, in a PC use case, or if the lid is flipped such that the system is in a tablet mode. The goal of this test is to confirm that the lid angles are reported correctly.
Objective:	Confirm that the angle between the base and lid is accurately reported.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. Place the system on a flat table. Record the reported angle over a 5 second period. 1. 0 degrees. Lid closed (screen facing keyboard). 2. 90 degrees. Screen open and facing the user. Screen and keyboard are orthogonal with user seeing screen and keyboard at the same time. 3. 180 degrees. Screen and keyboard both facing up. 4. 270 degrees. Screen and keyboard are orthogonal. The user cannot refer the screen and keyboard at the same time. 5. 360 degrees. System flat on table. The screen is facing up and the keyboard is facing down.
Test Pass/Fail Criteria:	Test will pass if all sequences show: The detected angle should be within a ± 10 degrees of accuracy. Over the 5 seconds, the variance of the angle should have been less than +/5 degrees.

7.13 Heading Sensor Accuracy and Drift Test

Test ID:	ISS_TST_13
Test Case Title:	Heading sensor accuracy and drift test
Mandatory/Optional:	Mandatory. Required if the system supports a magnetometer.
Description:	The e-compass using the system accelerometer and magnetometer can experience errors for multiple reasons including incorrect sensor calibration. This test is designed to show that the heading accuracy is correct in a number of angles/directions.

Test ID:	ISS_TST_13
Objective:	Confirm that the system reports the correct heading accuracy.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. If the system is a 2-in-1 device, the test should start with the system in the "PC" context (screen facing user with keyboard facing-up on the table). To test that the system is free of external magnetic influence: 1. Gather data from the magnetometer (@ rest) - confirm that the magnetometer is not moving more than 1-2 degrees while the system remains still. 2. Move the system 0.5 meters in each direction. Confirm that the compass reading does not change more than 1-2 degrees. **Intel® Platform Enablement Test Suite (Intel® PETS) will perform the following steps:** Test System Flat on Table (Z-UP) With a compass, place the system facing north on a flat table: 1. Start with the system placed facing north and flat on the table 2. Rotate the system to 90 degrees from North 3. Rotate the system to 180 degrees from North 4. Rotate the system to 270 degrees from North 5. Rotate the system to face the North ***Note:*** If system is a 2-in-1 device, convert it into a tablet form-factor (detached / screen on top of keyboard) and repeat this test sub-section.
Test Pass/Fail Criteria:	Test will pass if all sequences show: System heading error should not exceed 10 degrees at any rest position.



7.14 Intel® Integrated Sensor Solution Power States

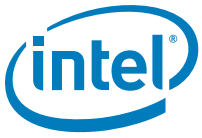
Test ID:	**ISS_TST_14**
Test Case Title:	Intel® Integrated Sensor Solution power states
Mandatory/Optional:	Mandatory
Description:	The purpose of this test is validate that the IISS is alive after system power transitions.
Objective:	IISS is alive without errors after power transitions.
Procedure:	**Automated (PETS)** Initial state of the SUT should be S0 (OS up) with the IISS configured in the system FW. Before running this test record the output of each IISS algorithm seen at the OS level. And confirm that the full sensor functional test has passed. Run the following power transitions from S0: 1. Resume from S4 on AC + DC 2. Resume from S4 on DC 3. Resume from S5 on AC + DC 4. Resume from S5 on DC 5. Resume after system reset (cold reset, HW RST button) 6. Resume after system reboot (warm reset, host based) After each system resume - check the output of each IISS algorithm seen at the OS level. And confirm that the full sensor functional test has passed. For manual testing - the sensor diagnostic tool can be used to read the output of the sensors. The sensor functional test can be run with the ISSUtil tool ("ISSUtil.exe -BIST -test 3").
Test Pass/Fail Criteria:	Test will pass if all sequences show: 1. System functional test records a "pass" after the system resumes to S0. 2. The algorithm outputs are within a +/-10% range of their previous values prior to the system power transition. ***Note:*** If the sensor or sensor micro-driver does not support the "built in functional test" (test level 3) then the test will return a warning to the user.

7.15 Sensor Activity Contexts

Test ID:	ISS_TST_15
Test Case Title:	Sensor Activity Contexts
Mandatory/Optional:	Optional. Perform the test if the system holds motion sensors.
Description:	The IISS contains activity context algorithms that can determine the user activities. This includes determining if the user is (1) sitting, (2) walking, or (3) running [at a safe speed]. These tests will confirm if the sensor activity contexts algorithms within the IISS are working properly.
Objective:	Confirm that the system will detect the system user activity contexts.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. Place the system on a flat table. If the system is a 2-in-1 system start in the tablet form factor. 1. Sit on a chair while looking at the system. The system should detect that the system is sedentary. 2. Pick up the system and begin walking with it. The system should detect that are walking with the system. 3. Start lightly running with the system. The system should detect that are running with the system.
Test Pass/Fail Criteria:	Test will pass if all sequences show: The system accurately detected the user contexts.

7.16 Sensor Terminal Contexts

Test ID:	ISS_TST_16
Test Case Title:	Sensor Terminal Contexts
Mandatory/Optional:	Optional. Perform the test if the system holds motion sensors.
Description:	The IISS contains terminal context algorithms that can determine how the user is holding the system. This includes determining if the system is held (1) face up / down, (2) portrait up / down, or (3) landscape left / right. These tests will confirm if the sensor terminal contexts algorithms within the IISS are working properly.
Objective:	Confirm that the system will detect the system user terminal contexts.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. Place the system on a flat table. If the system is a 2-in-1 system start in the tablet form factor. 1. Place the system face up and face down. 2. Place the system portrait up and portrait down. 3. Place the system landscape left and landscape right.
Test Pass/Fail Criteria:	Test will pass if all sequences show: The system accurately detected the terminal contexts.



7.17 Sensor Gesture Contexts

Test ID:	ISS_TST_17
Test Case Title:	Sensor Gesture Contexts
Mandatory/Optional:	Optional. Perform the test if the system holds motion sensors.
Description:	The IISS contains gesture context algorithms that can determine how the user is holding the system. This tests will confirm if the sensor gesture contexts algorithm within the IISS are working properly.
Objective:	Confirm that the system will detect the system user gesture contexts.
Procedure:	**Semi-Automated (PETS)** Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. Place the system on a flat table. If the system is a 2-in-1 system start in the tablet form factor. 1. Lift the system from the table and look at the system.
Test Pass/Fail Criteria:	Test will pass if all sequences show: The system accurately detected the terminal contexts.

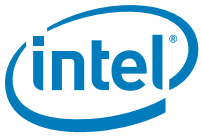
7.18 Wake on Shake Test

Test ID:	ISS_TST_18
Test Case Title:	Wake on shake test
Mandatory/Optional:	Mandatory
Description:	Wake on different events is a mandatory feature in Win10. As such a test that will focus on the ability to wake the system from S0i3 (CS) is a must.
Objective:	Test that ISH can send a wake event to Win OS and the OS waken from S0i3 to S0
Procedure:	1. Make sure that system is set in CS state (S0ix). 2. Make sure that shake event is defined in PDT and in Windows* (use SDT to check it). 3. Shake the system. 4. Windows* should wake and log on screen should appear. 5. Repeat the test 3 times. 6. There is a timeout (usually 2 minutes) until Win will go to SC again, unless the configuration of the specific copy of Windows* on the device set the timer to a different value.
Test Pass/Fail Criteria:	Test will pass if Windows* awakes all 3 times

7.19 Step Counting Test

Test ID:	**ISS_TST_19 (Manual)**
Test Case Title:	Step counting test
Mandatory/Optional:	Optional, Mandatory if the step counting is operational
Description:	Step counting is a standard virtual sensor that is being exposed in Windows* 10. The goal is to test that step counting sensor is working correctly
Objective:	Test that step counting sensor is working correctly and measure user steps
Procedure:	Initial state of the SUT should be S0 (OS up). The ISS should be configured in the system FW. 1. User should hold the tablet/notebook while he/she stands. 2. User should check SDT or any other sensor data report SW on the OS for the current number of step counter 3. User should start walking while counting his/her steps in a straight line. 4. After counting 50 steps user should stop. 5. User should compare the 50 steps he/she made to the number of steps shown on the software (after doing the needed math of subtracting the initial number of steps…) Remark: the step counter will start acting of 10 sec of stepping, so tests that will take 10 sec or will not be able to check the counter
Test Pass/Fail Criteria:	Amount of steps made by the user should be identical to step counter number on the SDT or any other sensor data SW.

§ §



(This page is intentionally left blank)

8 Manufacturing Flow Simulation

Intel provide Manufacturing Tool kit in every Intel® CSE firmware kit release which can be used for creating, modifying, and writing firmware binary image files, manufacturing testing, Intel® CSE setting information gathering. Those tools are located in Kit root directory and categorize in different folder for multi-OS support.

Manufacturing Flow Simulation section serves as a checklist to allow OEM to mature and validate their manufacturing process starting from earlier engineering validation stage in order to avoid any potential costly line-down issues and customer impact.

8.1 Test Environment Step

- LakeField Platform with Intel® CSE enabled firmware preload in SPI or UFS flash.
- Windows* Desktop or OS-A installed on OS storage device.
- Intel® CSE driver installed in Windows* Desktop OS or OS-A.

8.2 Tools for Testing

- FIT, FPT, MEManuf, Dnx module, and MEInfo – those tools can be found from Intel® CSE firmware kit.
- Recovery host – A host that can be used to execute the DnX tool
- USB cable connection between the recovery host to the system under test
- Intel® Platform Flash Tool (PFT)

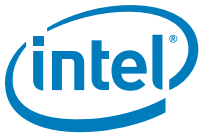
8.3 Manufacturing Flow Simulation Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows* 10 desktop, A = Microsoft* Windows* OS-A, U = UEFI shell.

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

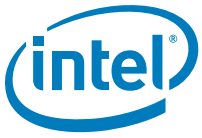
Test ID	Test Case Title	PETS Package Name	OS Supported	How?
MFG_01	Windows* Manufacturing Flow Simulation Test	N/A	W, A, U	M
MFG_02	Windows* Repair and Refurbish with UFS	N/A	W, A, U	M
MFG_03	Windows* Repair and Refurbish with SPI	N/A	W, A, U	M



8.4 Windows* Manufacturing Flow Test

Test ID:	**MFG_01**
Test Case Title:	Windows* Manufacturing Flow Test
Platform	LKF
Mandatory/Optional	Mandatory
Objective:	This test is to run Intel manufacturing tools in manufacturing simulation during the development phase to capture configuration, settings, and other potential issues that customers might encounter later in manufacturing, which will be costly.
Test Pass Criteria:	Pass only when all the tools run above return pass result. ***Note:*** When encounter failure, check: • CRB test result in Compliance kit. • Intel® CSE Firmware release note for known issues.
Description	For platforms with Intel® CSE, it is necessary to perform steps in the manufacturing line to ensure the Intel® CSE is functional and the system is secure, and ready for shipment. The minimum requirements can be met by following the Intel Manufacturing Reference Flow.
Windows* Procedure:	**Test Environment:** • System configuration should be as close as possible to what it will be like during production/manufacturing phase. For example, NFC module installed, and so forth. • Use the same OS environment as planning to use in the manufacturing line (with all the necessary driver/software installed, for example, Intel® MEI driver for Windows* OS, and so forth) **Test Preparation:** • Configure the desired CSE related configuration setting (DnX Ping/USB enumeration Timeout, VGA Display port/eDP port configuration, ODM ID and OEM SkuRule etc) in MEManuf.xml under variable check • If this test is conducted on platform with PV or later Production Firmware with production SoC, also configure the desired secure boot setting (OEM public key hash FPF, Key Manifest ID FPF and PTT enable FPF etc.) for Boot Guard and PTT in MEManuf.xml under configuration check • IFWI image created by FIT tool (i.e. IFWI.bin). • FPRR and BIOS lock should be disabled before perform this test case. • Customized MEManuf.xml file generated by MEManuf –cfggen MEManuf.xml and modified with properly changed per customer need. **Test Procedure:** 1. Use the version of FPT, DnX and MEManuf executable suitable for the chosen OS environment (located in the latest Intel® CSE kit) to simulate at least the Intel® CSE Manufacturing reference flow (Below steps). 2. If using SPI flash as boot source and pre-lock (the descriptor Master Access permission set to Intel recommended production value during image preparation by FIT tool), skip steps 4 and 6. 3. Reprogram the image currently on board (IFWI.bin). • If UFS as the boot media — Refer to Test Case DnX_03 to flash the FW • If SPI as the boot media — Windows* example: FPT.exe –f IFWI.bin — Reset Intel® CSE and Host after IFWI programming successfully (Windows* example: FPT.exe –greset) 4. Check Boot Guard and PTT setting etc NVAR matches with setting configured in FIT — Windows* example: MEManuf.exe -EOL var –f MEManuf.xml (use the MEManuf.xml configured during preparation) 5. Verify Intel® CSE — Windows* example: MEManuf.exe

Test ID:	MFG_01
	6. Set Intel® CSE EOM bit and descriptor Master Access permission to Intel recommended production value, then perform global reset to make sure Intel® CSE manufacturing mode is disabled and RPMB provisioning (only need if UFS is used for boot media) and FPF commit will be done after next boot. (Only perform this steps when ready for fusing and EOM flow, otherwise skip this step). — Windows* example: FPT.exe –closemnf -y 7. Perform end of line check on Intel recommended default check and also Boot Guard and PTT setting check — Windows* example: MEManuf.exe -EOL or — Windows* example: MEManuf.exe -EOL config –f MEManuf.xml — (Use the MEManuf.xml configured during preparation with any sub-test OEM would like to enable for final check). ***Note:*** It is highly recommended to create own script file to automatically run the above steps in order to better simulate the manufacturing flow.



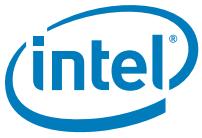
8.5 Windows* Repair Flow with UFS

Test ID:	MFG_02
Test Case Title:	Windows* Repair Flow with UFS
Platform	LKF
Mandatory/Optional	Mandatory
Objective:	This test is to go through repair and republish flow during the development phase to capture any potential manufacturing issues on data clear flow that customers might encounter later in manufacturing, which will be costly.
Test Pass Criteria:	Pass only when all the tools run above return pass result. ***Note:*** When encounter failure, check: • CRB test result in Compliance kit. • Intel® CSE Firmware release note for known issues.
Description	For platforms with Intel® CSE, it is necessary to perform steps in the manufacturing line to ensure Intel® CSE firmware can be updated or recovered after disable CSE manufacturing mode by FPT -closemnf command.
Windows* Procedure:	**Test Environment:** • System configuration should be as close as possible to what it will be like during production/manufacturing phase. For example, NFC module installed, and so forth. • Use the same OS environment as planning to use in the manufacturing line (with all the necessary driver/software installed, for example, Intel® MEI driver for Windows* OS, and so forth) **Test Preparation:** • Complete MFG_01 test case to ensure CSE MFG mode is disabled and EOM is set correctly. • IFWI image created by FIT tool (IFWI.bin) **Test Procedure:** 1. Trigger the FW DnX mode 2. Reflash Boot partition with IFWI image — Windows* example for UFS design: dnxFwDownloader.exe --command downloadfwos -- fw_dnx DNXP_0x1.bin --fw_image IFWI.bin --flags 0 3. Waiting for DnX command to complete flash process 4. Reset Intel® CSE and Host after IFWI programming successfully — Windows* example for UFS design: dnxFwDownloader.exe --command startover --flags 6 5. Verify Intel® CSE: — Windows* example: MEManuf.exe 6. Set Intel® CSE manufacturing done bit, then perform global reset to make sure Intel® CSE manufacturing mode is disabled: — Windows* example: FPT.exe –closemnf –y 7. Perform end of line check on Intel recommended default check and also ME/FPF setting check. (use the same MEManuf.xml configured during preparation which is under the same folder with MEManuf tool): — Windows* example: MEManuf.exe -EOL or MEManuf.exe -EOL config –f MEManuf.xml — (use the MEManuf.xml configured during preparation with any sub-test OEM would like to enable for final check)

8.6 Windows* Repair Flow with SPI

Test ID:	**MFG_03**
Test Case Title:	Windows* Repair Flow with SPI
Platform	LKF
Mandatory/Optional	Mandatory
Objective:	This test is to go through repair and republish flow during the development phase to capture any potential manufacturing issues on data clear flow that customers might encounter later in manufacturing, which will be costly.
Test Pass Criteria:	Pass only when all the tools run above return pass result. ***Note:*** When encounter failure, check: • CRB test result in Compliance kit. • Intel® CSE Firmware release note for known issues.
Description	For platforms with Intel® CSE, it is necessary to perform steps in the manufacturing line to ensure Intel® CSE firmware can be updated or recovered after disable CSE manufacturing mode by FPT -closemnf command.
Windows* Procedure:	**Test Environment:** • System configuration should be as close as possible to what it will be like during production/manufacturing phase. For example, NFC module installed, and so forth. • Use the same OS environment as planning to use in the manufacturing line (with all the necessary driver/software installed, for example, Intel® MEI driver for Windows* OS, and so forth) **Test Preparation:** • Complete MFG_01 test case to ensure CSE MFG mode is disabled and EOM is set correctly. • Flash Descriptor Override Strap implemented on the board and can be issued by EC or OEM tool. • IFWI image created by FIT tool (IFWI.bin). **Test Procedure:** 1. Power off system. 2. Assert Flash Descriptor Override Strap pin by any of physical mechanism available on design, for example jumper, push button or EC firmware. 3. Power on system and enter OS environment. 4. Update entire SPI flash by FPT. — Windows*: FPT.exe –f IFWI.bin 5. If jumper is used, power off system and remove flash descriptor jumper from the board, otherwise make sure Flash Descriptor Override Strap pin has been set to normal mode before next step. 6. Restart the system 7. Verify Intel® CSE: — Windows*: MEManuf.exe 8. Set Intel® CSE manufacturing done bit and Flash Descriptor Master Access permission to Intel recommended production value, then perform global reset to make sure Intel® CSE manufacturing mode is disabled: — Windows*: FPT.exe –closemnf 9. Perform end of line check on Intel recommended default check and also ME/FPF setting check. (use the same MEManuf.xml configured during preparation which under same folder with MEManuf tool): — Windows* example: MEManuf.exe -EOL or MEManuf.exe -EOL config –f MEManuf.xml — (use the MEManuf.xml configured during preparation with any sub-test OEM would like to enable for final check)

§ §



(This page is intentionally left blank)

9 Intel® Platform Trust Technology (Intel® PTT) Compliancy

Intel® Platform Trust Technology (Intel® PTT) is the Intel implementation of TCG TPM 2.0 standard in firmware. For more information about Intel® PTT integration with BIOS, refer BIOS Writers Guide and Intel® PTT Overview documentation.

The purpose of this section is to describe the tests required to verify PTT is functional, main PTT end to end use cases are working and platform meets Windows* 10 requirements for TPM 2.0 support.

The scope of this section is end to end testing and is not intended to provide TPM command level testing.

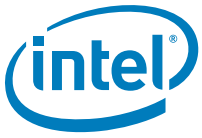
Note: Intel® Boot Guard testing with Intel® PTT is out of scope of this chapter and should be done as part of Intel® Boot Guard testing.

9.1 Test Environment Setup

- LakeField Platform with Intel® PTT enabled
- Windows* 10 Professional or Enterprise installed in UEFI mode
- Intel® CSE firmware and Intel® PTT enabled

9.2 Tools for Testing

- **Intel® Platform Enablement Test Suite** - Latest version of the tool is available in the Intel® CSE compliancy kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.
- Windows* 10 HLK Testing Environment
- manage-bde.exe (Windows* command line tool for BitLocker Driver Configuration)
- bdehdcfg.exe (Windows* command line tool for BitLocker Drive Encryption)
- makecert.exe (command line tool, part of Windows* 8, Windows* 10 SDK)
- pvk2pfx.exe (command line tool, part of Windows* 8, Windows* 10 SDK)
- CertUtil.exe (Windows* 8, Windows* 10 Command line tool)



9.3 Intel® Platform Trust Technology (Intel® PTT) Compliancy Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

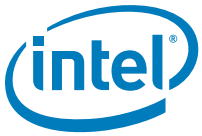
How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Physical eSPI configuration: S= SPI, U= UFS

Test ID	Test Case Title	PETS Package Name	OS Supported	How?	eSPI Config
PTT_001	CRB Interface Communication Test	Compliance_PTT.xml	W	A	S U
PTT_002	Intel® PTT Windows* 10 Basic Functionality	Compliance_PTT.xml	W	A	S U
PTT_003	TPM Clear and Physical Presence	Compliance_PTT.xml	W	A	S U
PTT_004	Windows* 10 BitLocker Integration	Compliance_PTT.xml	W	A	S U
PTT_005	Windows* 10 BitLocker TPM Protection	Compliance_PTT.xml	W	A	S U
PTT_006	Windows* 10 Virtual Smart Card (VSC) Tests	Compliance_PTT.xml	W	A	S U
PTT_008	Intel® PTT Enable/Disable from BIOS	Compliance_PTT.xml	W	M	S U
PTT_009	Power Transition Testing with Intel® PTT Enabled	Compliance_PTT.xml	W	A	S U
PTT_010	Dictionary Attack Lockout After Coin Battery Removal with EOM Commit	Compliance_PTT.xml	W	M	S U

9.4 CRB Interface Communication Test

Test ID:	**PTT_001**
Test Case Title:	CRB Interface Communication Test
Platform	LKF
Mandatory/Optional:	Mandatory ***Note:*** This test uses CRB access and therefore needs to run with disabled driver to ensure elimination of false failures.
Objective:	Verify BIOS is able to successfully send commands to Intel® PTT
Test Pass Criteria:	If TPM_CRB_CTRL_START register returns 0x00 after the duration listed in Table 15 of the TCG specification for the test command sent and before the listed timeout, the TPM command is received by PTT through HCI, the test passes, else fails. Test fails also if a timeout occurs at any other stage. ***Note:*** HCI reference code provides serial output status of whether or not TPM command is received by PTT. Check PttHciReceive function for more details.
Description:	The test confirms that BIOS correctly implements the CRB protocol for communication with Intel® PTT
Procedure:	1. Confirm Intel® PTT is enabled in the image. 2. Disable the Microsoft* TPM driver: From an Elevated Command Window issue the following command: reg add HKLM\SYSTEM\CurrentControlSet\Services\TPM /f /v ImagePath /t REG_EXPAND_SZ /d \SystemRoot\system32\drivers\tpm.sy_ Reboot the system 3. Relinquish locality 0: Write 1 to TPM_LOC_CTRL_0.Relinquish (0xfed40008, bit 1). 4. Request locality 0: Write 1 to TPM_LOC_CTRL_0.RequestAccess (0xfed40008, bit 0). 5. Verify TPM_LOC_STATE_x.locAssigned field (0xfed40000, bit 1) is set to 1 and that TPM_LOC_STATE_x.activeLocality field (0xfed40000, bits 2-4) is set to 000. 6. Write 1 to TPM_CRB_CTRL_REQ_0.cmdReady (0xfed40040, bit 0) 7. Poll TPM_CRB_CTRL_REQ_0.cmdReady every 5 ms for 500 ms until it is 0 8. Verify TPM_CRB_CTRL_STS_0.tpmIdle (0xfed40044, bit 1) is 0 9. Write a TPM command such as TPM2_SelfTest to TPM_CRB_DATA_BUFFER register (0xfed4_0080) 10. Write "1" to the TPM_CRB_CTRL_START register (0xFED4_004C). 11. Poll the TPM_CRB_CTRL_START register (0Xfed4_004C) until its value becomes "0". 12. Write 1 to TPM_CRB_CTRL_REQ_0.goIdle (0xfed40040, bit 1). 13. Poll TPM_CRB_CTRL_REQ_0.goIdle for 500ms until it is 0. 14. Relinquish locality 0: Write 1 to TPM_LOC_CTRL_0.Relinquish (0xfed40008, bit 1). 15. Verify TPM_LOC_STATE_x.locAssigned field (0xfed40000, bit 1) is set to 0 and TPM_LOC_STATE_x.activeLocality field (0xfed40000, bits 2-4) is set to 000. 16. Request locality 0: Write 1 to TPM_LOC_CTRL_0.RequestAccess (0xfed40008, bit 0). 17. Re-enable the Microsoft* TPM driver: From an Elevated Command Window issue the following command: reg add HKLM\SYSTEM\CurrentControlSet\Services\TPM /f /v ImagePath /t REG_EXPAND_SZ /d \SystemRoot\system32\drivers\tpm.sys Reboot the system ***Note:*** For detailed information on how to send a TPM command, refer to the PC client specific platform TPM profile for TPM 2.0

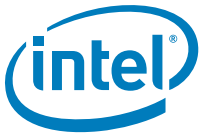


9.5 Intel® Platform Trust Technology (Intel® PTT) Basic Functionality under Windows* 10

Test ID:	**PTT_002**
Test Case Title:	Intel® PTT Basic Functionality Under Windows* 10
Platform	LKF
Mandatory/Optional:	Mandatory
Objective:	Windows* 10 can successfully communicate with Intel® PTT
Test Pass Criteria:	No "yellow bang" in device manager, Intel® PTT is the TPM device and all TPM queries return "true"
Description:	Verify Intel® PTT has been enabled on the platform and Intel® PTT is functional on Windows* 10
Procedure:	1. Boot to Windows* 10 UEFI installation 2. Open Device Manager (devmgmt.msc) and verify a "Trusted Platform Module 2.0" device exists in "Security Devices" 3. Open Trusted Platform Module (TPM) Management Page (tpm.msc) 4. Verify Manufacturer Name = **INTC**, TPM Specification Version = **2.0** 5. Verify Status is "The TPM is ready for use." 6. Open an elevated command prompt with admin privileges and enter powershell (type powershell at prompt) 7. Prepare the WMI object for querying Intel® PTT information by typing: *$ptt = get-wmiobject -namespace "root/cimv2/security/microsofttpm" win32_tpm* 8. Check different Intel® PTT parameters by typing the following at the PS prompt: a. $ptt.IsEnabled() b. $ptt.IsActivated() c. $ptt.IsAutoProvisioningEnabled() d. $ptt.IsOwned() e. $ptt.IsReadyInformation()

9.6 Trusted Platform Module (TPM) Clear and Physical Presence

Test ID:	**PTT_003**
Test Case Title:	TPM Clear and Physical Presence
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Verify TPM clear and take ownership flows work correctly under Windows* 10 OS and physical presence asserted
Test Pass Criteria:	OS takes ownership of TPM, new/old keys differ
Description:	TPM Clear command erases user data on the TPM. TPM Clear requires BIOS to check for physical presence to authorize the TPM Clear operation. We will save the SrkPublicKey and verify that new/old SRK keys differ after TPM Clear.
Procedure:	1. Save the current SrkPublicKey by performing the following actions: a. Open elevated command prompt and enter PowerShell by typing "powershell" at the prompt and type: b. $ptt = get-wmiobject -namespace "root/cimv2/security/microsofttpm" win32_tpm c. $ret = $ptt.GetSrkPublicKeyModulus() d. $ret.SrkPublicKeyModulus > SrkPubModOld.txt 2. Run "tpm.msc" to open TPM Management Console 3. Click 'Clear TPM…' in the Actions pane on right. 4. In the pop-up window click 'Restart' to invoke TPM Clear flow. 5. Upon reboot, a physical presence authorization message may be displayed (BIOS setting dependent) requiring the user to press a key to authorize the TPM clear or abort. In CRB, F12 will authorize, ESC rejects the operation. 6. Upon booting to Windows*, pop-up window will show up indicating OS is taking ownership of the TPM 7. After ownership operation completes, press OK. 8. Save the new SrkPublicKey by performing the following actions: a. Open elevated command prompt and enter PowerShell by typing "powershell" at the prompt and type: b. $ptt = get-wmiobject -namespace "root/cimv2/security/microsofttpm" win32_tpm c. $ret = $ptt.GetSrkPublicKeyModulus() d. $ret.SrkPublicKeyModulus > SrkPubModNew.txt 9. Compare the old and new keys



9.7 Windows* 10 BitLocker Integration

Test ID:	**PTT_004**
Test Case Title:	Windows* 10 BitLocker Integration
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Test BitLocker integration with Intel® PTT
Test Pass Criteria:	All system boots complete successfully and OS loads
Description:	BitLocker uses Intel® PTT to store and retrieve keys securely, in addition Windows* BitLocker confirms system components did not change by checking system load measurements saved to TPM. The test will verify BitLocker can be activated, BitLocker can encrypt, decrypt, and restart encryption after reboot.
Procedure:	1. In elevated permissions command line run: "bdehdcfg.exe –driveinfo" and check system drive is configured to support BitLocker 2. Set BitLocker to use TPM for measuring boot devices in Windows* Group Policy by: a. Run "gpedit.msc" to open Group Policy Editor b. Open "Local Computer Policy" > "Computer Configuration" > "Administrative Templates" > "Windows Components" > BitLocker Drive Encryption" > "Operating System Drives" c. On the right pane double click "Configure TPM platform profile for native UEFI firmware configuration" d. Check the enabled radio button. Verify PCR 0, PCR2, PCR4 and PCR11 are checked in the "Options" pane. e. Click apply and OK. f. Commit the group policy change by typing "gpupdate /force" in an elevated command prompt ***Note:*** This action is required once per OS installation 3. Set up tpm as a bitlocker protector with recovery password and turn-on BitLocker by typing the following at the command prompt a. manage-bde -protectors -add c: -tpm b. manage-bde -protectors -add c: -rp 000000-000000-000000-000000-000000-000000-000000-000000 c. manage-bde -on c: d. shutdown –r –t 0 4. After OS completes reboot, verify no error messages displayed. Wait for "Encryption in Progress" notification or type "manage-bde -status" to check on encryption status 5. After encryption reaches 10%, restart system, and verify encryption continues without error message after reboot completes. 6. Turn off BitLocker by typing "manage-bde –off c:" at the command line, decryption process should start 7. After decryption process ends, reboot and verify system boots into OS without error message. BitLocker should be off

9.8 Windows* 10 BitLocker TPM Protection

Test ID:	**PTT_005**
Test Case Title:	Windows* 10 BitLocker TPM Protection
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Verify BitLocker is using Intel® PTT for TPM protection
Test Pass Criteria:	BitLocker completes drive encryption successfully and reboots. System displays BitLocker recovery screen after choosing Disable Intel® PTT or Clear TPM in BIOS setup.
Description:	When BitLocker is set to use TPM protection, BitLocker will enter recovery mode if any protected component changed during boot. By disabling Intel® PTT, we will check BitLocker is indeed using TPM protection.
Procedure:	1. Encrypt the OS drive using BitLocker with TPM protection (follow instructions in PTT_004 steps 1 through 5, and wait till drive encryption reaches 10%) 2. Run manage-bde –status and verify drive is "protected" 3. Create a measured boot failure in order to trigger Bitlocker Recovery a. In BIOS, choose disable Intel® PTT or send a TPM_Clear command. ***Note:*** Clearing TPM by means of the OS will disable Bitlocker and will not prompt the user for his recovery password. The TPM must be cleared by the BIOS. b. System should boot into BitLocker recovery screen. Provide the recovery password to continue boot. c. Verify boot completes successfully 4. Disable BitLocker by typing "manage-bde –off c:" at the command line, decryption process should start 5. After decryption process ends, reboot and verify system boots into OS without error message. BitLocker should be off

9.9 Windows* 10 Virtual Smart Card Tests

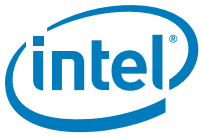
Test ID:	**PTT_006**
Test Case Title:	Windows* 10 Virtual Smart Card (VSC) Tests
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Intel® PTT can be used to support VSC use case
Test Pass Criteria:	VSC created successfully, certificate can be loaded and is persistent across reboot. VSC can be removed after key is deleted
Description:	Virtual Smart Card is a new Microsoft* use case for TPMs. More information on VSC can be found on Microsoft* web site. This test verifies a VSC can be created and certificate installed so VSC is accessible
Procedure:	1. Create a VSC running the following command on an elevated command line: tpmvscmgr.exe create /name TPM2VSC /adminkey random /PUK default /pin default /generate 2. Verify that TPM2VSC smart card reader was created in "Smart card readers" in device manager 3. Restart Windows*, and check the device is not yellow banged in device manager 4. Create and import a self-signed certificate into the VSC a. Ensure the following registry keys exist under [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Base Smart Card Crypto Provider]: • "AllowPrivateSignatureKeyImport"=DWord:00000001 • "AllowPrivateExchangeKeyImport"=DWord:00000001 b. Open an elevated command prompt c. Type: MakeCert.exe -sky exchange -r -n "CN=TPM2VSCCert" -pe -a sha1 -len 2048 -ss My -m 36 -sv "TPM2VSCCert.pvk" "TPM2VSCCert.cer" d. When requested, create a password. When asked for the password, provide the password created (for this example, using "123" as the password) e. Convert certificate to PFX format using the following command: pvk2pfx.exe -pvk "TPM2VSCCert.pvk" -pi 123 -spc "TPM2VSCCert.cer" -pfx "TPM2VSCCert.pfx" –f f. Import the certificate into the smart card using the following command: CertUtil.exe -p 123 -csp "Microsoft Base Smart Card Crypto Provider" -pin 12345678 -importpfx TPM2VSCCert.pfx AT_KEYEXCHANGE 5. Verify import was successful by examining the certificate in the VSC using the following command: CertUtil.exe -scinfo –pin "12345678". Window allowing to view the certificate will pop up, click OK to close 6. Restart the platform, and run step 5 again, to verify certificate persists after reboot 7. Remove the key from the VSC using the following commands a. Retrieve the name of the container to use by typing: CertUtil.exe -key -csp "Microsoft Base Smart Card Crypto Provider" -pin "12345678" -v -privatekey –user b. Use the container name returned in the previous command prefixed to the "[Default Container]" and replace the text in bold: CertUtil.exe -delkey -csp "Microsoft Base Smart Card Crypto Provider" -pin 12345678" -v -privatekey **"TPM2VSCCert-0d6e6c94-9bd6-4640-aa-63900"** 8. Destroy the VSC by running: `TpmVscMgr.exe destroy /instance ROOT\SMARTCARDREADER\0000`, making sure to use the correct index of the smartcard created

9.10 Intel® Platform Trust Technology (Intel® PTT) Disable/Enable from BIOS

Test ID:	**PTT_008**
Test Case Title:	Intel® PTT Disable/Enable from BIOS
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Ensure BIOS can enable and disable Intel® PTT successfully and that BIOS clears the TPM during disable
Test Pass Criteria:	When Intel® PTT is disabled; Intel® PTT does not show up in TPM management console. (It's possible for dTPM to show up pending on platform design).
Description:	BIOS may implement option to disable/enable Intel® PTT, or switch between Intel® PTT and a discrete TPM 1.2
Procedure:	1. Boot to OS, verify PTT_002 passing. 2. Reboot, enter BIOS and disable Intel® PTT through BIOS 3. Boot to Windows*, enter TPM Management Console (tpm.msc) and verify that either TPM is not available, or if TPM is available it is not Intel® PTT 4. Reboot, enter BIOS and enable Intel® PTT through BIOS 5. Boot to OS, verify PTT_002 passing ***Note:*** Intel® PTT enable/disable interface in BIOS is dependent on implementation and therefore not described

9.11 Intel® Platform Trust Technology (Intel® PTT) and Power Flows

Test ID:	**PTT_009**
Test Case Title:	Power Flow Testing
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Verify Intel® PTT does not interfere with system power operations
Test Pass Criteria:	All power flow tests pass, BitLocker does not enter into recovery mode
Description:	System with Intel® PTT enabled must pass all platform power flow testing. Intel® PTT must also be able to support all power flows when BitLocker is enabled and using Intel® PTT as a protector
Procedure:	1. Perform all platform power flow tests with Intel® PTT enabled 2. Encrypt the OS drive using BitLocker with TPM protection (follow instructions in PTT_004 steps 1 through 5, and wait till drive encryption reaches 3%) 3. Perform the following power transitions during encryption phase and after encryption has reached 10%: a. OS Restart b. OS ShutdownPower up c. Cold Reset (boot to internal EDK shell and type mm cf9 e -io) d. G3 (complete power off) e. Connected Standby (Windows* 8.1 CS) 4. After each Flow – test/verify PTT functionality (in S0) – run PPT_002



9.12 Dictionary Attack Lockout after Coin Battery Removal with EOM Commit

Test ID:	**PTT_010**
Test Case Title:	Dictionary Attack Lockout Mechanism with coin battery removal
Platform:	LKF
Mandatory/Optional:	Optional
Objective:	Allows OEM to validate the dictionary attack scenario after first coin battery removal, causing the counters to be reset. ***Note:*** This test is only for designs with RTC powered by coin cell battery.
Test Pass Criteria:	Intel® PTT will not allow access to user data (VSC) during lockout period post coin battery removal ***Note:*** In this test Field Programmable Fuses (FPF) will be blown on every battery removal and there is no recovery for it. Only select few processors to be used for this test and track them.
Description:	Intel® PTT keeps monotonic counters for Dictionary Attack (DA) under RTC power well. When RTC power is lost, Intel® PTT will enter lockout period to avoid Dictionary Attack for 2 hours. This is only after the coin battery has been removed 10 times and after EOM. Before that, Intel® PTT will not enter the lockout period of 2 hours. ***Note:*** During the 2 hour lockout period, no other Intel® PTT tests can be executed; even if correct credentials are provided. Execution of this test does not impact other non-Intel® PTT related testing. ***Note:*** This test can be run only once on a specific part. After this test is run, all FPF bits related to the feature will be blown. With such parts, test will consistently enter dictionary attack scenario after every RTC clear operation. **WARNING:** This flow is irreversible, the part will be permanently fused causing every RTC clear to cause a 2 hour TPM lockout.
Procedure:	1. System must post EOM procedure, as DA lockout will not occur during manufacturing mode 2. Set up a VSC with certificate (Instructions can be found in test PTT_006 steps 1 through 6) 3. Shutdown system, and perform RTC clear operation by removing all power and RTC battery from the board and close the RTC jumper. Repeat this procedure 11 times. 4. Return RTC battery and power, boot system to Windows* 5. Try to view the certificate in VSC by running: CertUtil.exe -scinfo -pin "12345678". 6. The command should fail due to Dictionary Attack lockout 7. Check the configured lockout configuration [3 min/10 tries OR 2 hours/ 32 tries] 8. According to the configuration (3 min/2 hours), wait for lockout to pass, and try again, it should be possible to access the certificate 9. Remove the certificate and VSC (Instruction can be found in test PTT_006 steps 8 and 9) ***Note:*** At step#3, the Intel® PTT is expected to enter a lockout period to avoid Dictionary Attack for 2 hours. This period cannot be adjusted.

§ §

(This page is intentionally left blank)

10 Download and Execute (DnX)

The purpose of this chapter is to describe the tests required in order to verify the functionality and system compliance to Download and Execute (DnX) solution for UFS platforms. While DnX can also be used to flash SPI parts, the focus of these tests are UFS centric.

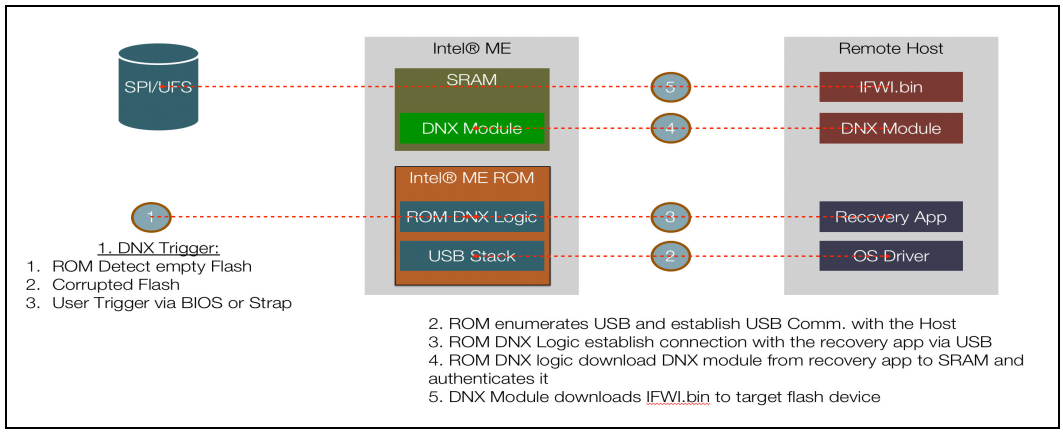
The chapter provides a high level overview of DnX, its use cases and the tests required to pass this section.

Note: This chapter is relevant only for systems with UFS flash device (as boot device).

10.1 What is DnX?

DnX is a Intel® CSE capability to use a USB port on the device and download content from another machine. Intel® CSE can use the downloaded content as an execution unit (i.e. it verifies the content and executes it) or as a data unit (it writes the content in UFS).

Figure 10-1. The DnX Flow



10.1.1 DnX Purpose and Detection

Primary use cases for DnX:

- Empty UFS
  - — Baseline: System has not yet been programmed with UFS content.
  - — Diagnosis: Intel®CSE automatically identifies the condition and enters DnX state.
  - — Using DnX, user can flash a new image to the UFS content and allow platform to boot up.
- Intel® CSE Detectable Corruption
  - — Baseline: System FW in UFS has been corrupted by an attacker and bad content has been placed in UFS such that it prevents platform boot (EC boot or Intel® CSE boot or BIOS boot)

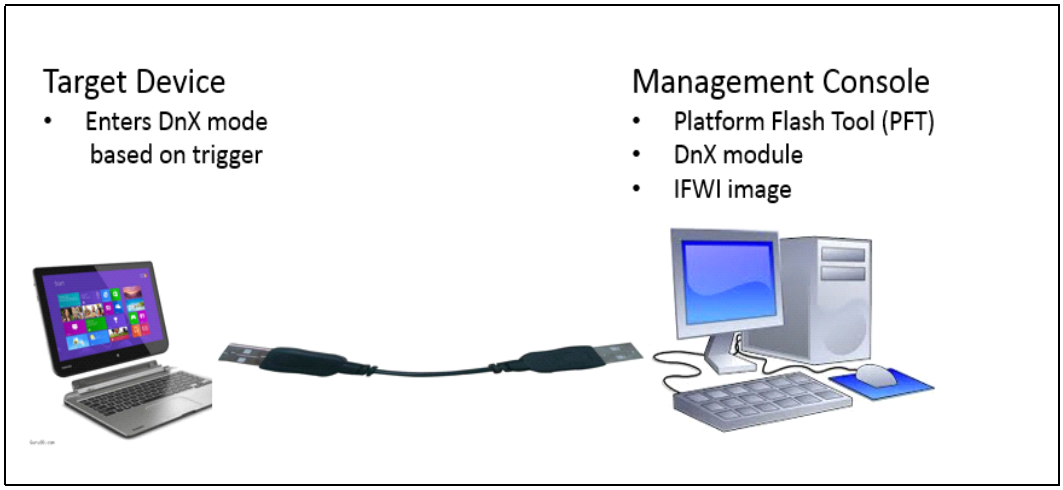
— Diagnosis: Intel® CSE automatically identifies the condition and enters DnX state.

— Using DnX, user can recover the UFS content and allow platform to boot up again.

- User Initiated DnX
  - **Baseline:**
    1. System FW in UFS has been compromised by attacker such that platform boots to a brick condition, e.g. BIOS executes a JMP $ instead of launching the OS boot loader or (PCH only) EC FW does not assert PCH PWROK even when PMC requests it via SLP_SX# signal de-assertion. In such conditions, CSE is unable to determine that the IA FW is corrupted
    2. System FW and OS are fully operational but user wants to use DnX to perform FW upgrade/downgrade
  - **Diagnosis:** User must provide Intel® CSE with DnX entry hint and CSE must be able to detect such hint and enter DnX (e.g. by pressing power button or combination of buttons). OEM needs to define a HW strap in order to set the DnX trigger. After the user uses the trigger, the system will reset and Intel® CSE will check for the HW strap during startup. If exists, Intel® CSE will enter DnX mode.

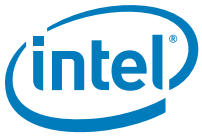
10.2 Test Environment Setup and Tools

Figure 10-2. DnX Test Setup



In order to complete this section, the following setup is required:

1. Recovery host – A host that can be used to execute the DnX tool
2. USB cable connection between the recovery host to the system under test. USB cable should be connected to USB port0 of the SUT.
3. Intel® Platform Flash Tool (PFT) - available in the Intel® CSE FW kit
4. DnX module (Provided in the Firmware kit)
5. IFWI image with DnX signed manifest to be flashed on the device under test (Can be created by Intel® FIT tool). This will not work with normal IFWI.



6. Configuration File (CFGFILE.XML)
   a. Required, if UFS device must be partitioned
   b. Can be used to create partition (LUNs) on blank UFS device. This can be used with Intel® PFT tool and will be included in the Intel® CSE Kit for LakeField platform. OEMs are expected to update this file according to LUN partition numbers and size they are going to use for UFS device.
7. OpenSSL: Free-ware, can be found in Open source community.
8. Flash Image Tool (FIT): Tool used to stitch FW image, can be found in Intel® CSE FW kit.
9. Flash Programming Tool (FPT): Tool used to burn images on SPI platforms, and set EOM state.

10.3 DnX Compliance Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support:

W = Microsoft* Windows*

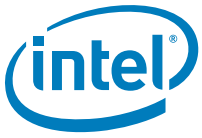
How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS, and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
DnX_01	DnX triggered on blank UFS	Compliance_Dnx_Blank.xml	W	I
DnX_02	Create Partitions on blank UFS device	Compliance_Dnx_Blank.xml	W	I
DnX_03	Flash IFWI to Blank UFS	Compliance_Dnx_Blank.xml	W	I
DnX_04	DnX triggered by User	Compliance_Dnx_OSProvisioned.xml	W	I
DnX_05	Read LUNs content	Compliance_Dnx_OSProvisioned.xml	W	I
DnX_06	Clear RPMB	Compliance_Dnx_OSProvisioned.xml	W	I
DnX_07	Write_OEMUnlockToken	Compliance_Dnx_OSProvisioned.xml	W	I
DnX_08	Read_OEMUnlockToken	Compliance_Dnx_OSProvisioned.xml	W	I
DnX_09	Erase_OEMUnlockToken	Compliance_Dnx_OSProvisioned.xml	W	I

- All the Test cases below uses Intel® PFT Tool. To keep unified format, all the test cases use command line interface of Intel® PFT.
- On the platforms that went thru EOM: Get UFS Info (getcardinfo), Partitioning (configpart), Recovery (downloadfwos), Data Clear (clearrpmb) and Reading of the boot media (readbootmedia), DnX operations must be authorized by OEM. Authorization is done thru OEM signed token.

Flow to open DnX capabilities closed at EOM:

1. Use PFT to generate new OEM unlock token with active "DnX Capabilities" knob. Set value of this knob according to the desired DnX capability. For more details on how to generate a token refer to "LKF Secure Tokens Guide"
2. Sign OEM unlock token using PFT. To see more details on how to sign a token refer to "LKF Signing and Manifesting Guide". Place signed token file inside the PFT folder.
3. Prepare signed OEM KM and locate it inside the PFT folder. OEM KM should contain:
   a. Public key hash of private key used for signing OEM unlock token
   b. Public key hash of private key used for signing DnX Image
4. Using PFT run DnX command to download OEM KM to the SRAM
   .\dnxFwDownloader.exe --command **downloadoemkeymanifest** --key .\OEM_KM.bin --fw_dnx .\DNXP_0x1.bin
5. Using PFT run DnX command to set capabilities in OEM unlock token
   .\dnxFwDownloader.exe --command **setcapabilities** --capabilities .\OEM_Unlock-Token.tok --fw_dnx .\DNXP_0x1.bin
6. Now can run DnX command that was previously closed with EOM. Token is valid until end of the DnX session during which it was set (using set capabilities command).

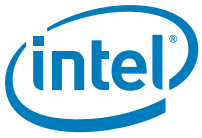


10.4 DnX Triggered on Blank UFS

Test ID:	**DnX_01**
Test Case Title:	DnX triggered on blank UFS
Platform:	LKF - Only UFS based boot platforms
Mandatory/Optional:	Mandatory ***Note:*** Test procedure is identical to test case UFS_01 and can be skipped if UFS_01 was already executed.
Objective:	Verify that CSE is able to detect an empty UFS and enter DnX mode
Test Pass Criteria:	Able to execute DnX commands and get valid response
Description:	Test will verify that CSE ROM recognizes empty UFS as one of its triggers and enters DnX mode
Procedure:	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Connect the SUT to a management console using a USB cable 3. Make sure that the Intel® Platform Flash Tool (Intel®PFT) is available on the management console. **Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode 3. On the management console, go into Intel®Platform Flash Tool (Intel® PFT) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and run DnX ID command: .\dnxFwDownloader.exe --command **iddevice**
Expected Response:	1. Check "ID DEVICE procedure" is success 2. Check "Response flags". Refer to DnX trigger table: • 0000b: HW Strap • 0010b: Bad NV content or Virgin Part • 0100b: Triggered by BIOS (BIOS config or error in BIOS loading) • 0101b: Unknown trigger (when running iddevice in Module context, since only ROM knows the trigger) • Others: Reserved **Example of expected response:** 08/20/18 10:17:02.151 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 10:17:02.156 DEBUG : Running command 'iddevice' on 08/20/18 10:17:02.157 INFO : Flags: 0 08/20/18 10:17:02.157 INFO : Starting ID DEVICE procedure 08/20/18 10:17:03.497 INFO : ID DEVICE procedure success 08/20/18 10:17:03.499 INFO : Response flags: 2 08/20/18 10:17:03.501 INFO : OEM platform ID: 0 08/20/18 10:17:03.508 INFO : Unique platform ID: 6268d2ce7f7b463d9d3243aba52e5d2a 08/20/18 10:17:03.512 INFO : Image errors: 00 00 00 00 00 00 00 00 00 00 00 00

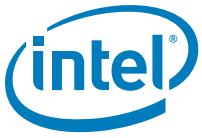
10.5 Create Partitions on Blank UFS Device

Test ID:	**DnX_02**
Test Case Title:	Create Partitions on blank UFS device
Platform:	LKF - Only UFS based boot platforms
Mandatory/Optional:	Mandatory ***Note:*** Test procedure is identical to test case UFS_02 and can be skipped if UFS_02 was already executed.
Objective:	Verify that CSE is able to configure UFS partitions
Test Pass Criteria:	UFS partitions are configured to the correct size
Description:	Test will partition blank UFS and verify correct configuration (number of LUNs, their size and attributes) using DnX
Procedure (1/2):	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Connect the SUT to a management console using a USB cable 3. Make sure that the PFT (Platform Flash Tool) is available on the management console. 4. Make sure DnX module available on the management console and located inside PFT folder 5. Make sure configuration file (cfgpart.xml indicating number of LUNs and their size) is available on the management console and located inside PFT folder **Cfgpart.xml creation:** The DNX tool can set the UFS's LUNs descriptor. Each LUN descriptor appears in the XML with the following lines: <lun idx="0"> <enable>true</enable> <boot-lun-id>0x1</boot-lun-id> <write-protect>0x0</write-protect> <mem-type>0x0</mem-type> <alloc-units>0x1000000</alloc-units> < data-reliability>0x0</data-reliability> <logical-block-size>0xc</logical-block-size> <provisioning-type>0x0</provisioning-type> <ctx-caps>0x0</ctx-caps> </lun> For basic usage, it is recommended to only change the enable, boot-lun-id and alloc-units lines. If an LUN isn't defined in the XML, it will be set as disabled, so need to define all the LUNs that need to be enabled. <alloc-units> is in endian swapped format. Some examples: 0x1000000 = 4MB 0x2000000 = 8MB 0x8000000 = 32MB 0x10000 = 1GB 0x100 = 256GB ***Note:*** Sample xml file, with size based on RVP recommendation, is located inside the kit. User/OEM is expected to update size per their requirement before creating partitions. Refer to DnX User Guide for more details on the parameters in xml file.



Test ID:	DnX_02
Procedure (2/2):	**Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode 3. On the management console, go into PFT (Platform Flash Tool) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool". 4. Open command prompt and run DnX commands as following: a. Configuration command: .\dnxFwDownloader.exe --command **configpart** --fw_dnx .\DNXP_0x1.bin --path cfgpart.xml --device ufs --idx 0 b. Get Card Info command: .\dnxFwDownloader.exe --command **getcardinfo** --fw_dnx .\DNXP_0x1.bin --idx 0 --device ufs ***Note:*** Those commands require OEM authorization (using DnX Token) post EOM. See instructions above in this document. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.

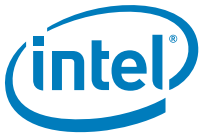
Test ID:	**DnX_02**
Expected Response(1/2):	1. Check Response flags is 0 (0=success) 2. Check that LUNs size as set in configuration file Check "GET CARD INFO procedure success" **Example of expected response:** 08/20/18 10:34:45.186 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 10:34:45.189 DEBUG : Running command 'getcardinfo' on 08/20/18 10:34:45.190 INFO : DnX module version: 1245 08/20/18 10:34:45.192 INFO : Device type: ufs 08/20/18 10:34:45.193 INFO : Device index: 0 08/20/18 10:34:45.194 INFO : Starting GET CARD INFO procedure 08/20/18 10:34:46.265 INFO : === UFS device info === --- Descriptor --- bBootEnable: [0x1] bDescAccessEn: [0] bInitPowerMode: [0x1] bHighPriorityLUN: [0x7f] bSecureRemovalType: [0] bInitActiveICCLevel: [0] wPeriodicRTCUpdate: [0] --- LUN [0] --- bLUEnable: [0x1] bBootLunID: [0] bLUWriteProtect: [0] bMemoryType: [0] dNumAllocUnits: [0x1b0000] bDataReliability: [0] bLogicalBlockSize: [0xc] bProvisioningType: [0x3] wContextCapabilities: [0] --- LUN [1] --- bLUEnable: [0x1] bBootLunID: [0x1] bLUWriteProtect: [0] bMemoryType: [0] dNumAllocUnits: [0x8000000] bDataReliability: [0] bLogicalBlockSize: [0xc] bProvisioningType: [0] wContextCapabilities: [0] ... --- LUN [7] --- bLUEnable: [0] bBootLunID: [0] bLUWriteProtect: [0] bMemoryType: [0] dNumAllocUnits: [0] bDataReliability: [0] bLogicalBlockSize: [0] bProvisioningType: [0] wContextCapabilities: [0]



Test ID:	DnX_02
Expected Response(2/2):	--- Attributes --- AttributesEnables: [0] bRefClkFreq: [0] bBootLunEn: [0x1] bCurrentPowerMode: [0x11] bActiveICCLevel: [0] bOutOfOrderDataEn: [0] bMaxDataInSize: [0x40] bMaxDataOutSize: [0x40] bConfigDescrLock: [0] bMaxNumOfRTT: [0x2] wExceptionEventControl: [0] dSecondsPassed: [0] --- Geometry Descriptor --- bLength: [0x48] bLength: [0x48] bDescriptorType: [0x7] bMediaTechnology: [0] qTotalRawDeviceCapacity: [124993536] dSegmentSize: [2097152] bAllocationUnitSize: [1] bMinAddrBlockSize: [8] bOptimalReadBlockSize: [8] bOptimalWriteBlockSize: [8] bMaxInBufferSize: [64] bMaxOutBufferSize: [64] bRPMBReadWriteSize: [64] bDataOrdering: [0] bMaxContextIDNumber: [0x5] bSysDataTagUnitSize: [0] bSysDataTagResSize: [0] bSupportedSecRTypes: [0x9] wSupportedMemoryTypes: [0xf80] dSystemCodeMaxNAllocU: [0x9a3b0000] wSystemCodeCapAdjFac: [0x1] dNonPersistMaxNAllocU: [0x9a3b0000] wNonPersistCapAdjFac: [0x1] dEnhanced1MaxNAllocU: [0x9a3b0000] wEnhanced1CapAdjFac: [0x2] dEnhanced2MaxNAllocU: [0] wEnhanced2CapAdjFac: [0] dEnhanced3MaxNAllocU: [0] wEnhanced3CapAdjFac: [0] dEnhanced4MaxNAllocU: [0] wEnhanced4CapAdjFac: [0] GET CARD INFO procedure success

10.6 Flash IFWI to Blank UFS

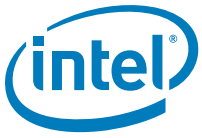
Test ID:	**DnX_03**
Test Case Title:	Flash IFWI to Blank UFS
Platform:	LKF - Only UFS based boot platforms
Mandatory/Optional:	Mandatory ***Note:*** Test procedure is identical to test case UFS_04 and can be skipped if UFS_04 was already executed.
Objective:	Make sure IFWI can be flashed into UFS
Test Pass Criteria:	Image was successfully flashed
Description:	SUT will go into DnX mode and use its capabilities to flash provided Image into UFS
Procedure:	**Test preparation:** 1. Make sure that UFS in the SUT is blank (has no image flashed) 2. Make sure that UFS in the SUT was partitioned and Boot partition size > IFWI size, refer to DnX_02 for details. 3. Connect the SUT to a management console using a USB cable 4. Make sure that the PFT (Platform Flash Tool) is available on the management console. 5. Make sure DnX module available on the management console and located inside PFT folder 6. Make sure that the image to be flashed to the SUT, stored in the PFT directory. Refer to Bring Up Guide for instructions on how to create DnX Image. **Test procedure:** 1. Power on the SUT 2. SUT will start boot and then enter DnX mode 3. On the management console, go into PFT (Platform Flash Tool) folder e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and run following commands: a. Download Image command: .\dnxFwDownloader.exe --command **downloadfwos** --fw_dnx .\DNXP_0x1.bin --fw_image .\Image_DNX.bin --flags 0 ***Note:*** This command requires OEM authorization (using DnX Token) post EOM. See instructions above in this document. b. Device Reset command: .\dnxFwDownloader.exe --command **startover** --flags 9 ***Note:*** this is optional, issuing this command will provide remote reset followed by full boot. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.



Test ID:	DnX_03
Expected Response:	1. Check "DOWNLOADFWOS procedure" is success **Example of expected response:** 08/20/18 14:42:04.067 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 14:42:04.078 DEBUG : Running command 'downloadfwos' on 08/20/18 14:42:04.082 INFO : DnX module version: 1245 08/20/18 14:42:04.084 INFO : Starting DOWNLOADFWOS procedure 08/20/18 14:42:06.506 INFO : DOWNLOADFWOS procedure success 2. Check "STARTOVER procedure" is success **Example of expected response:** 08/20/18 14:44:46.935 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 14:44:46.939 DEBUG : Running command 'startover' on 08/20/18 14:44:46.940 INFO : Starting STARTOVER procedure 08/20/18 14:44:46.941 INFO : Flags: 10 08/20/18 14:44:48.001 INFO : STARTOVER procedure success 08/20/18 14:44:48.005 INFO : Current operation: 4 08/20/18 14:44:48.014 INFO : Current context: 1 3. Check that SUT boots normally w/o going into DnX mode

10.7 DnX Triggered by User

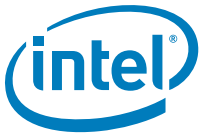
Test ID:	**DnX_04**
Test Case Title:	DnX triggered by User
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Verify that CSE is able to detect a user initiated DnX trigger and go into DnX mode
Test Pass Criteria:	Able to execute DnX commands and get valid response
Description:	After the system is booted to OS, user presses the hotkey defined by OEM to enter DnX mode
Procedure:	**Test preparation:** 1. Connect the SUT to a management console using a USB cable 2. Make sure IFWI + OS are provisioned. Refer to DnX_02 and DnX_03. 3. Make sure that the PFT (Platform Flash Tool) is available on the management console. 4. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Power on the SUT 2. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 3. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and run DnX ID command: .\dnxFwDownloader.exe --command **iddevice**
Expected Response:	1. Check "ID DEVICE procedure" is success 2. Check "Response flags". Refer to DnX trigger table: • 0000b: HW Strap • 0010b: Bad NV content or Virgin Part • 0100b: Triggered by BIOS (BIOS config or error in BIOS loading) • 0101b: Unknown trigger (when running iddevice in Module context, since only ROM knows the trigger) • Others: Reserved **Example of expected response:** 08/20/18 10:17:02.151 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 10:17:02.156 DEBUG : Running command 'iddevice' on 08/20/18 10:17:02.157 INFO : Flags: 0 08/20/18 10:17:02.157 INFO : Starting ID DEVICE procedure 08/20/18 10:17:03.497 INFO : ID DEVICE procedure success 08/20/18 10:17:03.499 INFO : Response flags: 0 08/20/18 10:17:03.501 INFO : OEM platform ID: 0 08/20/18 10:17:03.508 INFO : Unique platform ID: 6268d2ce7f7b463d9d3243aba52e5d2a 08/20/18 10:17:03.512 INFO : Image errors: 00 00 00 00 00 00 00 00 00 00 00 00



10.8 Read LUN's Content

Test ID:	**DnX_05**
Test Case Title:	Read LUN's content
Platform:	LKF - Only UFS based boot platforms
Mandatory/Optional:	Mandatory
Objective:	Verify that LUN can be read when in DnX mode
Test Pass Criteria:	LUN was successfully read and its content dump to output file
Description:	Test will put SUT in DnX mode (trigger by User), then issue the DnX command to read desired LUN
Procedure:	**Test preparation:** 1. Make sure UFS was configured and IFWI + OS are provisioned. Refer to DnX_02 and DnX_03. 2. Connect the SUT to a management console using a USB cable 3. Make sure that the PFT (Platform Flash Tool) is available on the management console. 4. Make sure DnX module available on the management console and located inside PFT folder 5. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Power on the SUT 2. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 3. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and run Read Boot Media command (this operation allows the tool to read contents of the FW from NVM): a. To read OS user space partition: .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpLUN0.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **16** b. To read Boot partition: .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpLUN1.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **17** c. To read Temp Data Partition (LUN6) partition: .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpLUN6.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **22** d. To read RPMB partition: .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpRPMB.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **48** ***Note:*** Make sure that output file location can be accessed for write, otherwise operation will fail. ***Note:*** Make sure to set correct number of blocks to read, based on the partition size. **1 block = 1kByte.** E.g. how to read 4MB LUN: 4MB = 4096kB (in binary) --> need to read 4096 blocks ***Note:*** This command requires OEM authorization (using DnX Token) post EOM. See instructions above in this document. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.

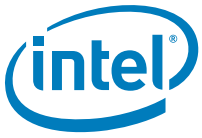
Test ID:	**DnX_05**
Expected Response:	1. Check "READ BOOT MEDIA procedure" is success 2. Check Partition index is correct (e.g. 48 – for RPMB) **Example of expected response:** 08/20/18 13:25:44.542 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 13:25:44.547 DEBUG : Running command 'readbootmedia' on 08/20/18 13:25:44.549 INFO : DnX module version: 1245 08/20/18 13:25:44.550 INFO : Device type: ufs 08/20/18 13:25:44.566 INFO : Device index: 0 08/20/18 13:25:44.581 INFO : Start offset: 0 08/20/18 13:25:44.584 INFO : Blocks to read: 4096 08/20/18 13:25:44.591 INFO : Partition index: 0 08/20/18 13:25:44.603 INFO : Starting READ BOOT MEDIA procedure 08/20/18 13:25:48.057 INFO : READ BOOT MEDIA procedure success ***Note:*** when platform is pre EOM and no RPMB provisioning was done, reading RPMB partition will fail with following error: 08/20/18 14:32:53.760 INFO : Starting READ BOOT MEDIA procedure 08/20/18 14:32:54.814 ERROR : RPMB not provisioned ***Note:*** Returned data is in the 'raw' as read from the media and is not processed at all by DnX module (i.e. no decryption etc. is performed, rather all data is returned as stored on the media)



10.9 Clear RPMB

Test ID:	**DnX_06**
Test Case Title:	Clear RPMB
Platform:	LKF - Only UFS based boot platforms
Mandatory/Optional:	Optional
Objective:	Verify RPMB data can be cleared
Test Pass Criteria:	RPMB data was successfully cleared
Description:	Test will put SUT in DnX mode (trigger by User), then issue the DnX command to clear RPMB partition
Procedure:	**Test preparation:** 1. Connect the SUT to a management console using a USB cable 2. Make sure that the PFT (Platform Flash Tool) is available on the management console. 3. Make sure DnX module available on the management console and located inside PFT folder 4. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Complete DnX_03 (or UFS_04) test to ensure IFWI was programmed on the NVM 2. Perform EOM. Refer to MFG_01 for more detailed instructions. Windows* example: FPT.exe –closemnf -y 3. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 4. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 5. Open command prompt and run Clear RPMB commands: a. Read RPMB partition (before erasing it): .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpRPMB_before.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **48** b. Clear RPMB partition: .\dnxFwDownloader.exe --command **clearrpmb** --fw_dnx .\DNXP_0x1.bin --idx 0 --device ufs c. Read RPMB partition (after erasing it): .\dnxFwDownloader.exe --command **readbootmedia** --fw_dnx .\DNXP_0x1.bin --path C:\temp\dumpRPMB_after.bin --device ufs --idx 0 --start 0 --blocks 4096 --part **48** ***Note:*** Make sure that output file location can be accessed for write, otherwise operation will fail. ***Note:*** Make sure to set correct number of blocks to read, based on the partition size. **1 block = 1kByte.** E.g. how to read 4MB LUN: 4MB = 4096kB (in binary) --> need to read 4096 blocks ***Note:*** This command requires OEM authorization (using DnX Token) post EOM. See instructions above in this document. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide.

Test ID:	**DnX_06**
Expected Response:	Check "CLEAR RPMB procedure" is success Check output files before and after clearing PRMB: Inside the "After clear RPMB" file should see 32 bits in addresses 0x0000 and 0x2000 cleared to '0'. (32 bits in addresses 0x10000 and 0x30000 will also be cleared, but they have no impact, so this test doesn't need to check them). **Example of expected response:** 08/20/18 14:17:48.067 INFO : dnxFwDownloader version 1.0.0.0 (API: 13.30.0.7063(DBG)) build time: Monday July 09th 2018, 10:44:27 UTC 08/20/18 14:17:48.089 DEBUG : Running command 'clearrpmb' on 08/20/18 14:17:48.100 INFO : DnX module version: 1245 08/20/18 14:17:48.101 INFO : Device type: ufs 08/20/18 14:17:48.108 INFO : Device index: 0 08/20/18 14:17:48.115 INFO : Starting CLEAR RPMB procedure 08/20/18 14:17:49.300 INFO : CLEAR RPMB procedure success



10.10 Write_OEMUnlockToken

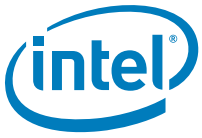
Test ID:	**DnX_07**
Test Case Title:	Write_OEMUnlockToken
Platform:	LKF
Mandatory/Optional:	Mandatory ***Note:*** Test procedure can be skipped if **SIGN_SECTOK_03** was already executed and DnX was used to inject the token.
Objective:	Verify that OEM Unlock Token can be successfully injected into the platform thru DnX
Test Pass Criteria:	Getting "success" response from PFT when sending inject token command
Description:	Test will put SUT in DnX mode (trigger by User), then issue the DnX command to Write OEM Unlock Token, then check the response from PFT
Procedure:	**Test preparation:** 1. Create a pair of keys for the Debug Token, for example, using OpenSSL. In order to use the Intel® PFT to create tokens, go to Security tab (on the top) --> General Setting --> inside the Signing tab choose "Local keys" signing method and enter key information --> press OK. In the main PFT page under "Security" tab (on the left), create a new OEM unlock token (make sure to set "OEM Unlock" and the "ISH GDB Debug" knobs to "Activated"), fill in desired details and press "Generate & Sign token" button. For more details see LKF Secure Tokens guide inside the CSE FW kit. 2. Enter the public key hash into the OEM Key Manifest's field for OEMUnlockTokens. Details of procedure for creating the hash and for entering the hash into the OEM Key Manifest are in the LKF Signing and Manifesting Guide. 3. Use MEU to manifest and sign the OEM Key Manifest. Details of procedure are in the LKF Signing and Manifesting Guide. 4. Use FIT to build an IFWI image including the OEM Key Manifest. Details of procedure are in the LKF Signing and Manifesting Guide. 5. Burn the IFWI image to the platform, and use FPT –EOM to close manufacturing state. ***Note:*** Do **Not** stitch token within the IFWI image. 6. Connect the SUT to a management console using a USB cable 7. Make sure that the PFT (Platform Flash Tool) is available on the management console. 8. Make sure DnX module available on the management console and located inside PFT folder 9. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Power on the SUT 2. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 3. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and inject token via Intel® Platform Flash Tool command .\dnxFwDownloader.exe --command writetoken --fw_dnx .\DNXP_0x1.bin --token C:\temp\token_to_write.bin --slot 0 ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide. ***Note:*** Also PFT GUI can be used instead of command line, Refer to LKF Secure Token Guide for instructions. 5. Reboot the platform. The token will be consumed and validated by the firmware on the next platform reset.
Expected Response:	Check PFT response that token has been written successfully

10.11 Read_OEMUnlockToken

Test ID:	**DnX_08**
Test Case Title:	Read_OEMUnlockToken
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Verify that OEM Unlock Token can be successfully read from the platform thru DnX
Test Pass Criteria:	Getting "success" response from PFT when sending read token command
Description:	Test will put SUT in DnX mode (trigger by User), then issue the DnX command to Read OEM Unlock Token, then check the response from PFT
Procedure:	**Test preparation:** 1. Complete DnX_07 test to ensure the SUT has token 2. Connect the SUT to a management console using a USB cable 3. Make sure that the PFT (Platform Flash Tool) is available on the management console. 4. Make sure DnX module available on the management console and located inside PFT folder 5. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Power on the SUT 2. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 3. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and read token via Intel® Platform Flash Tool command .\dnxFwDownloader.exe --command **readtoken** --fw_dnx .\DNXP_0x1.bin --path C:\temp\read_token.bin --slot 0 ***Note:*** Make sure that output file location can be accessed for write, otherwise operation will fail. ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide. ***Note:*** Also PFT GUI can be used instead of command line, Refer to LKF Secure Token Guide for instructions.
Expected Response:	Check PFT response that token has been read successfully from the device into file

10.12 Erase_OEMUnlockToken

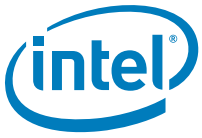
Test ID:	**DnX_09**
Test Case Title:	Erase_OEMUnlockToken
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	Verify that OEM Unlock Token can be successfully erased from the platform thru DnX
Test Pass Criteria:	Getting "success" response from PFT when sending erase token command



Test ID:	DnX_09
Description:	Test will put SUT in DnX mode (trigger by User), then issue the DnX command to Erase OEM Unlock Token, then check the response from PFT
Procedure:	**Test preparation:** 1. Complete DnX_07 test to ensure the SUT has token 2. Connect the SUT to a management console using a USB cable 3. Make sure that the PFT (Platform Flash Tool) is available on the management console. 4. Make sure DnX module available on the management console and located inside PFT folder 5. DnX hotkey sequence it determined by the OEM (and is usually unique for each OEM). **Test procedure:** 1. Power on the SUT 2. After the system is booted to OS, press the hotkey to enter DnX mode. The platform will reset and should enter DnX mode when exiting reset. 3. On the management console, go into PFT (Platform Flash Tool) folder. e.g. "C:\Program Files (x86)\Intel\Platform Flash Tool" 4. Open command prompt and erase token via Intel® Platform Flash Tool command .\dnxFwDownloader.exe --command erasetoken --fw_dnx .\DNXP_0x1.bin --slot 0 ***Note:*** For explanations about the parameters in command line, refer to DnX User Guide. ***Note:*** Also PFT GUI can be used instead of command line, Refer to LKF Secure Token Guide for instructions.
Expected Response:	Check PFT response that token has been erased successfully from the device

§ §

(This page is intentionally left blank)



11 Intel® Boot Guard 2.1

Boot Guard is an Intel platform boot integrity protection technology. Boot Guard can help protect the platform boot integrity by preventing execution of unauthorized boot block. With Boot Guard, the OEM can create a platform boot policies such that invocation of an unauthorized (or compromised) boot block will trigger the platform protection per the OEM policies. Based in the hardware, Boot Guard will also extend the trusted boundary of the platform boot process down to the hardware. A benefit of this protection is that Boot Guard can help OEM maintains platform integrity by preventing reuse of the OEM hardware to run unauthorized software stack.

11.1 Scope

This section describes a validation strategy for Boot Guard 2.1. This chapter is intended for validation purposes. The objective is to provide validation professionals with additional insight into Boot Guard by highlighting validation considerations. This chapter is not a technology overview and does not replace the existing Boot Guard collateral. The reader is expected to be familiar with Boot Guard and to use this document as a validation supplement to develop his own validation plan.

11.2 Pre-requisite

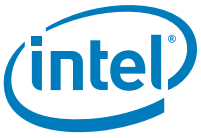
11.2.1 Tools Supported

This Boot Guard evaluation plan documented in this chapter requires the following components and tools for execution.

Tool/Component	Revision	Comments
BPMgen2	2.0 or higher	Tool designed to generate the Boot Policy Manifest (BPM) and can be used in a batch file to automatically update the BIOS image with the new BPM.
FIT	CSE Firmware Kit with Boot Guard Support	FIT is required to define the Boot Guard Boot Policies (persistent policies). Available on VIP
MEInfo	CSE Firmware Kit with Boot Guard Support	MEInfo is required to confirm Boot Guard Policies.
Flash Programming Tool (FPT)	CSE Firmware Kit with Boot Guard Support	FPT is used to commit the boot guard related values to IFP, read and display the values from IFP
MEManuf	CSE Firmware Kit with Boot Guard Support	To compare the flash/IFP Boot Guard values and display the result
TxtBtgInfo	0.7.10 or higher	The tool will verify the integrity of the OEM Key manifest and the Boot Policy Meta data File Extension

11.2.2 Boot Media Support

The below tests are applicable for all boot Media Devices (SPI and UFS):



11.3 Boot Guard Test Coverage Summary

Note: Profile 1 and profile 2 support has been deprecated. Only Profile 0: NO_FVME, Profile 3: VM, Profile 4: FVE and Profile 5: FVME is supported.

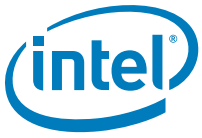
Boot Guard disabled is referred to as Profile 0:NO_FVME.

(Profile 3 is only supported for pre-production/debug configuration and its not supported after end of manufacturing has been committed.)

(Verified only refers to profile 4 as unlike profile 5 - measured and verified it only has verified enabled)

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M= Manual.

Network Factor: LAN = systems with LAN interface and test is performed using LAN interface, WLAN = systems with WLAN interface and test is performed using the WLAN interface.



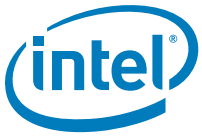
11.4 ME Boot Guard 001

Test ID	Test Case Title	Target OS	How?
ME_BtG_001	Platform Boot with Boot Guard Disabled (using FPF Emulation)	W	M
ME_BtG_002	Successful verified only Boot to OS (using FPF Emulation)	W	M
ME_BtG_003	Unsuccessful verified only boot (using FPF Emulation)	W	M
ME_BtG_004	Successful VM (Verified Measured) Boot to OS (using FPF Emulation)	W	M
ME_BtG_005	Unsuccessful VM (Verified Measured) Boot to OS (using FPF Emulation)	W	M
ME_BtG_006	Successful VM (Verified Measured) Boot to OS using HW FPFs	W	M

Test ID:	**ME_BtG_001**
Test Case Title:	Platform Boot with Boot Guard Disabled (using FPF Emulation)
Objective:	Objective of the test is to verify that the platform boots with Boot Guard Disabled.
Test Pass Criteria:	Platform successfully boots to OS and verified or measured boot should not have executed. MEinfo output for Boot guard profiles under the "ME" column should show Verified Boot and Measured Boot Disabled. All other settings should be as per the configuration done via FIT tool.
Description:	In this test case, boot guard flow for verification and measurement of IBB will not be executed during the platform boot process.
Procedure:	**Prerequisite** • Use FIT tool to create the full image with Boot Guard Profile 0 in the Platform Protection tab as per the details in the Firmware Bring up guide • Run the TxtBtgInfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -f <IFWI Image> — The test results should show pass for KM and BPM (would not be present in case of a No_FVME profile) **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to No_FVME profile. Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Boot the platform to OS or EFI shell 3. Run the MEinfo tool to check the profile settings.

11.5 ME Boot Guard 002

Test ID:	**ME_BtG_002**
Test Case Title:	Successful verified only Boot to OS (using FPF Emulation)
Objective:	This test verifies that the SUT has all the required components: hardware, firmware and BIOS. Additionally, the SUT has been correctly provisioned in manufacturing for the platform to boot with the Boot Guard for IBB verification
Test Pass Criteria:	1. Platform boot should be successful without any hangs or errors. 2. MEInfo output should be as per the boot guard profile configured. It should show Verified only enabled and check for the fields under "FPF" column for FPF contents and "ME" for NVAR contents. Ensure that these matches with what was provisioned during the image creation process.
Description:	In this Test case, Boot guard will authenticate and load the IBBL and perform a successful verification of the SUT (IBB) or Initial Boot Block, IBB would in turn authenticate the rest of the BIOS components.
Procedure:	**Prerequisite** • Use the BPMgen2 tool for signing the BIOS components and generating the manifest (BPM and KM) • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide. **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to Verified only profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Flash the image and boot the platform to OS 3. Run the MEinfo tool to check the profile settings 4. Check the platform behavior as per the details in the Test Pass Criteria.



11.6 ME Boot Guard 003

Test ID:	ME_BtG_003
Test Case Title:	Unsuccessful verified only Boot to OS (using FPF Emulation)
Objective:	To verify that the platform will fail to boot if verified boot is enabled if there is an IBB corruption or ACM
Test Pass Criteria:	Platform should shutdown or enter the DnX mode
Description:	In this Test case, Boot guard will fail the verification of the SUT (IBB) or Initial Boot Block and prevent the platform from booting
Procedure:	**Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to Verified only profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Provision an incorrect BootPolicyManifest Hash Public Key while creating the image or change signature of the manifests using the BPMgen2 tool. ***Note:*** To provision an incorrect "Boot Policy Manifest" key for this test: a. Change the build settings in the FIT tool for the field "Verify manifest signing keys against the OEM key Manifest" to "No" to avoid build failure with an incorrect value for the negative test. b. Using BPMgen2, create OEM Key Manifest binary with corrupted key hash binary for BootPolicyManifest (refer to Signing and Manifesting guide for how to create OEM KM binary). i. May corrupt the key hash binary using hex editor. 3. Install the targeted OS if not already installed on the SUT and try booting the platform.

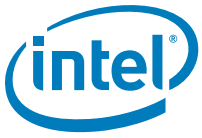
11.7 ME Boot Guard 004

Test ID:	ME_BtG_004
Test Case Title:	Successful VM(Verified/Measured) Boot to OS (using FPF Emulation)
Objective:	This test verifies that the SUT has all the required components: hardware, firmware and BIOS. Additionally, the SUT has been correctly provisioned in manufacturing for the platform to boot with the Boot Guard for IBB verification and Executes Measured only boot.
Test Pass Criteria:	1. Platform successfully boots to OS 2. Check for the fields under "ME" for NVAR contents, verified boot and Measured Boot should be enabled. Ensure that these matches with what was provisioned during the image creation process.
Description:	In this test Platform would boot with IBB successful IBB verification and measurement of the BIOS components

Test ID:	ME_BtG_004
Procedure:	**Prerequisite** • Use the BPMgen2 tool for signing the BIOS components and generating the manifest (BPM and KM) • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide. **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to VM profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Boot the system to OS 3. Run the MEinfo tool to check the profile settings.

11.8 ME Boot Guard 005

Test ID:	ME_BtG_005
Test Case Title:	Unsuccessful VM(Verified/Measured) Boot to OS (using FPF Emulation)
Objective:	To verify that the platform will fail to boot if verified boot and measured boot is enabled under the condition when there is an IBB corruption or ACM
Test Pass Criteria:	Platform should shutdown or enter the DnX mode
Description:	In this test case boot guard will perform an unsuccessful verification and measurement of the SUT (System Under test) Initial Boot Block (IBB). Upon verification failure platform boot will be prevented and System will enter the recovery mode.
Procedure:	**Prerequisite** • Use the BPMgen2 tool for signing the BIOS components and generating the manifest (BPM and KM) • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide. **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to VM profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Flash the Intel® CSE firmware and BIOS image that are Boot Guard enabled with VM profile 3. Corrupt the Manifests or Hash keys to create a fail scenario. (E.g. enter the wrong public hash key while image creation) 4. Try booting the platform to OS.



11.9 ME Boot Guard 006

Test ID:	**ME_BtG_006**
Test Case Title:	Successful VM(Verified/Measured) Boot using HW FPFs
Objective:	This test verifies that the SUT has all the required components: hardware, firmware and BIOS. Additionally, the SUT has been correctly provisioned in manufacturing for the platform to boot with the Boot Guard for IBB verification and executes Verified and Measured boot using the values from the FPFs instead of FPF mirror
Test Pass Criteria:	1. Platform successfully boots to OS 2. Run the MEinfo tool to check the profile settings. It should show Verified and Measured Boot enabled under "FPF" column. And all the other settings for Boot Guard should be reflected as per the configurations done using FIT tool.
Description:	In this test case Boot Guard will perform the feature testing using the values from the FPFs (i.e. accessing the profile values from the FPFs instead of the flash variables, NVARs)
Procedure:	**Prerequisite** • Use the BPMgen2 tool for signing the BIOS components and generating the manifest (BPM and KM) • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide • Perform this test ONLY when all the above tests (ME_BtG_001 to ME_BtG_005) has passed using the profile values from the FPF mirror or NVARs. ***Note:*** The profiles selected to be committed into FPFs will become the final profile which cannot be altered later. Take care of the prerequisites before proceeding further with the test **Prepare the SUT** 1. Install the targeted OS if not already installed on the SUT 2. Provision the SUT with a desired Boot Guard Profile (Say VM) per testing objective. Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings along with the Signing and Manifest Guide 3. Perform the step to commit the Boot guard profile value to FPFs. This will be done automatically after CSE manufacturing mode is disabled (during the global reset from FPT –closemnf or first boot for pre-lock image) if FW and Si are both production, or done by means of a specific FPF MEI command (If combination of FW and Si is Pre-production). Below commands can be used for FPF commit (Also refer to the CSE tools user guide for the commands usage) "FPT –FPFs" - To retrieve the FPF names "FPT –CLOSEMNF" – To commit all of FPF values FPF HW Boot the Platform to the desired OS.

§ §

(This page is intentionally left blank)

12 Signing, Manifesting, and Secure Tokens

This chapter includes tests to verify that OEMs are able to create signed IFWI images, create Secure Tokens for Debug, and successfully inject them into the platform. It also verifies that OEMs are able to enter the hash of the token public key into the OEM Key Manifest, and build and image with this manifest, such that the platform will recognize the injected token.

Secure Tokens are only supported on platforms with a signed IFWI image.

The tests in this chapter are only relevant for OEMs creating signed IFWI images. OEMs not enabling Secure Boot will not need to create signed IFWI images.

12.1 Test Environment Setup

Signing and manifesting documentation can be found in CSE FW kit that details usage of signing the tokens.

12.2 Tools for Testing

- Platform Flash Tool (PFT): Tool used for DnX mode, and token creation/injection. Tool can be found in latest CSE FW kit.
- OpenSSL: Freeware, can be found in Open source community.
- Flash Image Tool (FIT): Tool used to stitch FW image, can be found in CSE FW kit.
- Flash Programming Tool (FPT): Tool used to burn images on SPI platforms, and set EOM state.

12.3 Signing, Manifesting, and Secure Tokens Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

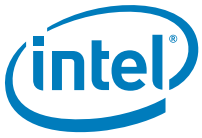
OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
SIGN_SECTOK_02	Image Creation with OEM Signed Components	N/A	W	M
SIGN_SECTOK_03	Debug Token	N/A	W	M

12.4 Image Creation with OEM Signed Components

Test ID:	**SIGN_SECTOK_02**
Test Case Title:	Image Creation with OEM signed components
Objective:	This test verifies that OEMs are able to create an IFWI image with signed OEM components, and successfully bring up the platform
Test Pass Criteria:	Platform boots to OS
Description:	OEM will sign OEM-provided binaries in the IFWI image. The public key hashes of all the OEM-provided binaries signatures will be entered into the OEM Key Manifest, which will be itself signed and included in the IFWI image.
Windows* Procedure:	**Manual procedure.** 1. Create pairs of keys for signing OEM-provided binaries, using OpenSSL. Details of procedure are in the LKF Signing and Manifesting Guide. A minimum of one pair of keys must be created that can be used for all signing, but a separate pair is ideally used for each OEM-provided binary and the OEM Key Manifest. The OEM-provided binaries include ISH, iUnit, aDSP, if the OEM plans to replace the Intel provided binaries with his own. 2. Enter the public key hashes of all the keys into the OEM Key Manifest's respective fields. If multiple key hashes are entered, separate nodes need to be created in the OEM Key Manifest xml, one for each different hash. Details of procedure for creating hashes are in the LKF Signing and Manifesting Guide. 3. Use MEU to manifest and sign the OEM Key Manifest. Details of procedure are in the LKF Signing and Manifesting Guide. 4. Use MEU to sign (or resign) each OEM-provided binary whose hash has been entered into the OEM Key Manifest. 5. Enter the hash of the OEM Key Manifest key and the OEM Key Manifest binary into FIT, and then use FIT to build an IFWI image including the OEM Key Manifest. Details of procedure are in the LKF Signing and Manifesting Guide. 6. Burn the IFWI image to the platform. 7. Verify that the platform boots to the OS.



12.5 Debug Token

Test ID:	**SIGN_SECTOK_03**
Test Case Title:	Debug Tokens
Objective:	This test verifies that OEMs are able to create Secure Tokens for Debug, and successfully inject them into the platform. It also verifies that OEMs are able to enter the hash of the token public key into the OEM Key Manifest, and build an image with this manifest, such that the platform will recognize the injected token.
Test Pass Criteria:	Platform is in OEM Unlock State
Description:	OEM will create a token. The public key hash will be entered into the OEM Key Manifest, which will be included in the IFWI image. The token will be injected into the platform using DnX. Platform moves to OEM Unlock Status
Windows* Procedure:	**Manual procedure.** 1. Create a pair of keys for the Debug Token, for example, using OpenSSL. Details of procedure are in the LKF Secure Tokens guide found in latest CSE FW kit. In order to use the Intel® Platform Flash tool to create tokens, the Private key and the password used to create this key should be entered in the Intel® Platform Flash tool under Security tab (on the top) -> General Settings as Certificate and password respectively. 2. Enter the public key hash into the OEM Key Manifest's field for OEMUnlockTokens. Details of procedure for creating the hash are in the LKF Signing and Manifesting Guide, chapter 3, and details for entering the hash into the OEM Key Manifest are in chapter 5. 3. Use MEU to manifest and sign the OEM Key Manifest. Details of procedure are in the LKF Signing and Manifesting Guide, chapter 5. 4. Use FIT to build an IFWI image including the OEM Key Manifest. Details of procedure are in the LKF Signing and Manifesting Guide, chapter 5. a. In order to create and sign an OEM Unlock token, use the Intel® Platform Flash Tool ensuring to set the OEMUnlockEnabled knob to OEMUnlockEnabled, and the ISH GDB Debug knob to "enabled" — follow instructions in the LKF Secure Token guide. b. To stitch token within the IFWI image, — Use FIT also to add token in image. — Burn the IFWI image to the platform, and use FPT –EOM to close manufacturing state. c. If the user chooses NOT to stitch within IFWI, then continue to inject token via Intel® Flash Programming tool OR Intel® Platform Flash Tool if the user is using DnX APIs (Refer to LKF Secure Token Guide for instructions) 5. Verify that platform functionality is (Orange) unlocked, and available for debugging. Ensure the following are done: a. After injecting token via DnX or stitched in image by FIT, boot platform with DCI enabled b. Connect Lauterbach to capture NPK messages (using NPK decoder released in compliance kit) c. Initiate warm reset d. Verify NPK log contains the following message: "Accept secure token".

§ §

(This page is intentionally left blank)

13 Intel® Trace Hub

NPK Set of silicon features + Software/API.

With NPK it's possible to:

- System level debug
- Power transitions which involves PMC, PUNIT, BIOS and CSE
- Debugging cross-component by issuing time-stamps for every SW/HW

13.1 Tools for Testing

- System Trace tool from Intel® System Debugger (part of Intel® System Studio NDA product) installed on the host computer, where the tests are run. The latest version of Intel® System Studio NDA can be downloaded from https://registrationcenter.intel.com/en/forms/?productid=2336&SupportCode=ENA&pass=yes. For setup and usage refer to the System Trace User Guide located at "C:\IntelSWTools\system_studio_2018_nda\documentation_2018\en\debugger\system_studio_2018_nda\system_debugger\system_trace".
- Enable DCI by setting Direct Connect Interface (DCI) Enabled under the debug tab of Intel® FIT to 'Yes'. Click Build Image and generate the full SPI image. Refer to the Bringup Guide for more details on image creation.
- Refer to DCI user guide (#597454) for more detail about how to configure the environment

13.2 Intel® Trace Hub Test Coverage Summary

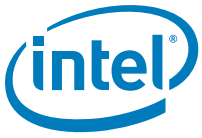
Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*
How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
ME_ITH_001	Intel® CSE FW - DCI enable Using MIPI-60 Connector	N/A	W	M
ME_ITH_002	Intel® CSE FW - DCI enable Using USB3 Connector	N/A	W	M
ME_ITH_003	Capture ITH BIOS and CSE tracing via LTB	N/A	W	M

13.3 Intel® CSE FW - DCI Enable Using MIPI-60 Connector

Test ID:	**ME_ITH_001**
Test Case Title:	Intel® CSE FW - DCI enable using MIPI60 connector
Platform	LKF
Mandatory/Optional:	Mandatory
Objective:	To enable DCI through MIPI-60 connector
Test Pass Criteria:	Test passes if we are able to connect to the target over DCI with MIPI-60 connector by using Lauterbach with LA-4512 probe
Description:	Intel® ITH is using DCI solution for locating logs from target platform. DCI can connect to target platform through the MIPI-60 connector. Refer to the LKF Platform Design Guide (PDG) to configure and enable DCI. DCI can enabled in differ ways as: • Enable DCI through BIOS • Connect the LA-4512 probe and coming out of G3 This test focuses on checking the connection using MIPI-60 connector
Procedure:	1. Flash the full image on to the platform. 2. Enable DCI with any of the above options. 3. Connect the target end of the LTB LA-4512 probe into MIPI-60 on the target. 4. Connect to the target by clicking the connect button on the Target Connection tab and **verify there are not any error message as**: [ERROR] [npk_config_api] Failed Intel® Trace Hub hardware detection check!



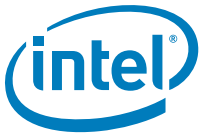
13.4 Intel® CSE FW - DCI Enable Using USB3 Connector

Test ID:	**ME_ITH_002**
Test Case Title:	Intel® CSE FW - DCI enable
Platform	LKF
Mandatory/Optional:	Mandatory
Objective:	To enable DCI through USB3 connector
Test Pass Criteria:	Test passes if we are able to connect to the target over DCI with USB3 connector
Description:	Intel® ITH is using DCI solution for locating logs from target platform. DCI can connect to target platform through the USB3 connector. Refer to the platform design guide to configuring and enabling DCI. DCI can enabled in differ ways as: • Enable DCI through BIOS • Connect the USB3 cable and come out of G3 This test focuses on checking the connection using USB3 connector
Procedure:	1. Flash the full image on to the platform. 2. Enable DCI with any of the above options. 3. Connect the target end of the USB3 cable into USB3 on the target. 4. Connect to the target by clicking the connect button on the Target Connection tab and verify there are not any error message as: [ERROR] [npk_config_api] Failed Intel® Trace Hub hardware detection check!

13.5 Capture ITH BIOS and CSE Tracing via LTB

Test ID:	**ME_ITH_003**
Test Case Title:	Capture ITH BIOS and CSE tracing via LTB
Platform	LKF
Mandatory/Optional:	Mandatory
Objective:	Collect Intel® Trace Hub logs using LTB device
Test Pass Criteria:	The test passes if we are able to collect BIOS and Intel® CSE logs in the STT and the messages are time correlated.
Description:	Collect Intel® CSE and BIOS logs using the STT tool
Windows* Procedure:	1. Install Intel® System Studio NDA SW on the host 2. Connect LTB between the target and host 3. Boot the target using a debug BIOS with DCI enabled 4. Open the STT tool in eclipse 5. Use a fresh workspace and use the setup project menu to configure the trace project 6. Connect to the target by clicking the connect button on the Target Connection tab 7. Check CSME and BIOS for the trace sources ***Note:*** Selecting BIOS is optional, if the BIOS does not support trace messages over Intel® Trace Hub 8. Start the trace by clicking the play button in the Trace Capture tab 9. Restart the target by selecting the restart option from windows* menu 10. Shut down target by selecting shutdown option from windows* menu 11. Check if we are able to collect BIOS and Intel® CSME logs in the STT and the messages are time correlated. 12. Power on the target to boot from S5 to S0 state 13. Check if we are able to collect BIOS and Intel® CSME logs in the STT and the messages are time correlated. 14. Execute a cold reset by writing to CF9 register (mm CF9 0xE –io) 15. Check if we are able to collect BIOS and Intel® CSME logs in the STT and the messages are time correlated. 16. Execute a warm reset by writing to CF9 register (mm CF9 0x6 –io) 17. Check if we are able to collect BIOS and Intel® CSME logs in the STT and the messages are time correlated.

§ §



(This page is intentionally left blank)

14 Intel® Dynamic Application Loader (Intel® DAL)

14.1 Introduction

Intel® Dynamic Application Loader (Intel® DAL) is an Intel® CSME infrastructure for applications such as Intel® Identity Protection Technology (Intel® IPT).

The following table documents compliancy tests to verify Intel® Dynamic Application Loader (Intel® DAL) is working on the platform.

This Test plan is targeted at all OEMs.

Note: Intel® IPT testing is out of this compliance guide scope. Intel® IPT has a dedicated kit which includes validation and collateral (available on VIP).

14.2 Test Environment for the Intel® Dynamic Application Loader (Intel® DAL)

Note: No OEM implementation is required on the board/BIOS or EC level. Intel® CSME should be set to Enabled in FIT when creating the firmware image.

The Management console could be a laptop or a desktop a version of Windows* supported by Intel® Platform Enablement Test Suite. The network to use is a hub/ switch and network cables.

The Intel® DAL tests should not be conducted in Windows* Server 2008 as Intel® DAL currently does not supports this OS.

Note: DAL Applet needs to be signed with RSA 3K due to CSE signing method upgrade.

14.2.1 Tools for Testing

Intel® Platform Enablement Test Suite—Latest version of the tool from the Intel® CSME Compliancy kit release. Refer to the Intel® Platform Enablement Test Suite (Intel® PETS) user guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.

Package DAL.xml should be loaded to Intel® PETS in order to compete the tests in this section

14.2.2 Verify Needed Software is Installed on Host

The following software components need to be available in the platform OS:

Intel® MEI Driver:
This is the interface used for communication between the host OS components and the Intel® CSME components (included in the general Intel® CSME installer kit).

Intel® Dynamic Application Loader (Intel® DAL) host software components:
Exposes an API that allows communication between the host client and the application (included in the general Intel® CSME installer kit)

14.3 Intel® Dynamic Application Loader (Intel® DAL) Test Coverage Summary and Details

Test ID	Test Case Title	PETS/Manual	Form Factor
DAL_001	Intel® DAL applications cleanup	PETS	DT/MB/AIO/WS-Server
DAL_002	Intel® DAL test application installation and load	PETS	DT/MB/AIO/WS-Server
DAL_003	Intel® DAL communication channel exercise	PETS	DT/MB/AIO/WS-Server

Note: DT = Desktop, MB = Mobile, AIO = All In One, WS-Server = WS-Server

Test ID:	**DAL_001**
Test Case Title:	Intel® DAL applications cleanup
Mandatory/Optional:	Mandatory
Firmware SKU:	All Consumer and Corporate SKUs (Desktop, Mobile, and Workstation)
Description:	Intel® DAL applications cleanup mechanism test
Objective:	To test that the Intel® Dynamic Application Loader (Intel® DAL) clean-up mechanism works properly, and no application is currently running in Intel® DAL
Procedure:	Start test DAL_001 in the Intel® Platform Enablement Test Suite from management console. **Intel® Platform Enablement Test Suite performs the following steps:** 1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. Perform cleanup of all Intel® DAL applications.
Test Pass/Fail Criteria:	All steps return the value "Passed"

Test ID:	**DAL_002**
Test Case Title:	Intel® DAL test application installation and load
Mandatory/Optional:	Mandatory
Firmware SKU:	All Consumer and Corporate SKUs (Desktop, Mobile, and Workstation)
Description:	Intel® DAL test application is installed and loaded, verifying basic functionality of Intel® DAL applications execution capability.

Test ID:	**DAL_002**
Objective:	To test that the Intel® Dynamic Application Loader (Intel® DAL) basic functionality works properly.
Procedure:	Start test DAL_002 in the Intel® Platform Enablement Test Suite from management console. **Intel® Platform Enablement Test Suite performs the following steps:** 1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm that the needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. Confirm test application can be installed and loaded to Intel® Dynamic Application Loader. 4. Unload the test application.
Test Pass/Fail Criteria:	All steps return the value "Passed"

Test ID:	**DAL_003**
Test Case Title:	Intel® DAL communication channel exercise
Mandatory/Optional:	Mandatory
Firmware SKU:	All Consumer and Corporate SKUs (Desktop, Mobile, and Workstation)
Description:	Intel® DAL test application is installed and loaded, followed by a communication channel exercise between application and host side application.
Objective:	To test that the Intel® Dynamic Application Loader (Intel® DAL) application can communicate successfully with a host application.
Procedure:	Start test DAL_003 in the Intel® Platform Enablement Test Suite from management console. **Intel® Platform Enablement Test Suite performs the following steps:** 1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. Exercise basic communication channel between test application and host to verify connectivity flow 4. Unload the test application.
Test Pass/Fail Criteria:	All steps return the value "Passed".

§ §

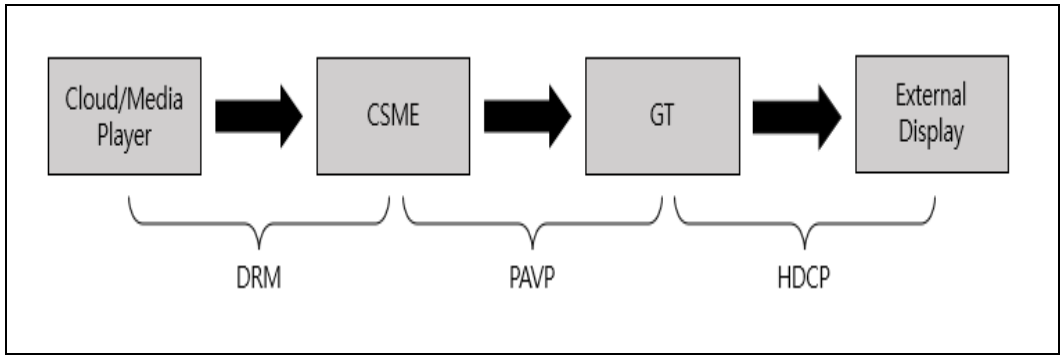
15 Protected Media Playback

15.1 Overview

Protected Media Playback is supported by Intel® CSE Firmware. Intel® CSE employs the following content protection mechanism to safe guard premium content form copy:

a. Intel® Protected Audio Video Path

b. Intel® High-bandwidth Digital Content Protection



The Protected Audio/Video Path (PAVP) is an Intel-specific collection of content protection features in the Intel® "Gen" graphics products. The purpose of PAVP is to support premium content video playback including Blu-ray discs and provide a protected path from the media player application to the GPU HW.

Protection of the data as it leaves the GPU and goes to an external display is typically done using industry standard HDCP.

15.2 Scope

This chapter describes a validation strategy for protected content Protected Media Playback. This chapter is intended for validation purposes. The objective is to provide validation professionals with additional insight into Media Playback protection offered by Intel® CSE by highlighting validation considerations. This chapter is not a technology overview. The reader is expected to be familiar with Protected Media Playback or Content Protection and to use this document as a validation supplement to develop his own validation plan.

15.3 Prerequisite

This Protected Media Playback evaluation plan documented in this chapter requires the following components and tools for execution.

Intel® Flash Image Tool (fit.exe)

Intel® Flash Programming Tool (Intel® FPT) - is available in Windows* 32-bit (fptw.exe), Windows* 64-bit (fptw64.exe) operating systems, EFI 32-bit and EFI 64-bit.

15.4 Test Environment Setup

The System under Test (SUT) is to be configured in manual configuration mode a with wired LAN or wireless LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured to ensure that the wired LAN and wireless LAN addresses reside on separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

15.5 Media Playback Test Coverage Summary

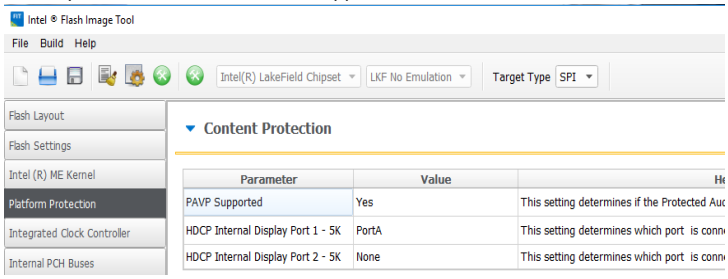
'OS Support', and 'How?' Columns describes the test methodology.

OS Support: W = Microsoft* Windows*

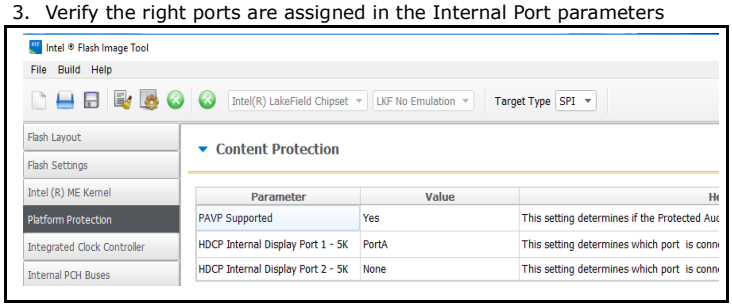
How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Network Factor: LAN = systems with LAN interface and test is performed using LAN interface, WLAN = systems with WLAN interface and test is performed using the WLAN interface.

Test ID	Test Case Title	How?	OS Support	Network Factor
Media_001	Verify default configuration settings for Protected Audio Video Path [PAVP] in Flash Image Tool [FIT]	M	W	All
Media_003	Verify Internal Port configuration in Flash Image Tool [FIT]	M	W	All
Media_004	Verify PAVP Enabled in BIOS (***Only if the SUT BIOS menu displays PAVP Mode***)	M	W	All

Test ID:	**Media_001**
Test Case Title:	Verify default configuration settings for Protected Audio Video Path [PAVP] in Flash Image Tool [FIT]
Platform	LKF
Mandatory/Optional:	Mandatory
Description:	Intel® CSE initiates PAVP secure session in firmware for key exchange and encryption for Content from Media player or cloud. PAVP can be enabled or disabled using FIT Tool. In this test we verify the PAVP is enabled in the SUT SPI image using FIT.
Objective:	Verify if the PAVP control in Intel® FIT are set correctly
Procedure:	1. Open customer image in FIT tool 2. Got to Platform Protection tab 3. Verify and ensure if the 'PAVP Supported Parameter' is set to 'Yes' 
Test Pass/Fail Criteria:	Test passes is FIT PAVP parameter is set to 'Yes' when we open SPI image in FIT.

Test ID:	**Media_003**
Test Case Title:	Verify Internal Port configuration in Flash Image Tool [FIT]
Platform	LKF
Mandatory/Optional:	Mandatory
Description:	For configuring ports that are connected to internal SUT panel or eDP panel Intel® CSE provides configuration parameter in FIT to assign port to internal. In this test, user verifies what are the Internal port assignment set in SUT SPI image and confirm if they are intended ports to configured as internal. ***Note:*** Only ports that are planed to be connected to internal panels /eDP should be assigned as internal Port A is set as the default Internal port by Intel® ME. If the FIT parameter for internal port is set to 'None' Intel® CSE will assign Port A to internal. When a port is set to internal HDCP encryption is by-passed by Intel® CSE even if the content license requires it. Do not assign ports that are planned to be connected to HDMI,DVI,DP to internal.

Test ID:	Media_003
Objective:	Verify if the internal port configuration parameter in FIT assigns the right port.
Procedure:	1. Open customer image in FIT tool 2. Got to Platform Protection tab 3. Verify the right ports are assigned in the Internal Port parameters 
Test Pass/Fail Criteria:	***Test pass criteria:*** The FIT Internal Port parameters have the right ports assigned intended to be connected to internal panel/eDP

Test ID:	Media_004
Test Case Title:	Verify PAVP Enabled in BIOS
Platform	LKF
Mandatory/Optional:	Mandatory (Only if the SUT BIOS menu displays PAVP Mode)
Description:	PAVP can be configured in the BIOS. In this test we will verify what the PAVP mode is enabled in SUT BIOS.
Objective:	Verify PAVP configuration in BIOS
Procedure:	1. Boot system to BIOS menu 2. Navigate in BIOS menu and PAVP Option [e.g. in Intel BIOS goto - Intel Advance Menu->System Agent (SA) Configuration->Graphics Configuration-> PAVP Enable 3. Verify the PAVP mode setup 
Test Pass/Fail Criteria:	Test passes if we PAVP is enabled in SUT BIOS.

§ §

16 Intel® Integrated Clock Control Compliancy

This chapter covers details of Intel® ICC test cases supported on Lake Field platforms.

Intel® ICC feature support:

Below table displays ICC Feature/Configuration supported on LKF platforms.

ICC Feature/Configuration Supported
• Standard • Adaptive

ICC Profile and parameters configuration recommendation

- Review Intel® Bringup Guide to get familiar with supported frequency and SSC configurations for above features.
- Review Intel ® ICC Tools user guide to get familiar with the ICC SDK
- OEMs are recommended to configure ICC Boot profile and parameters for the profile via Intel® FIT -> ICC tab. Make sure to choose appropriate profile and configure parameters that meet platform and HW requirements.

Intel® PETS test package detail for Intel® ICC

The test cases supported by platforms using Intel® Platform Enablement Test Suite (Intel® PETS) are defined as a part of Compliance_ICC_*.xml.

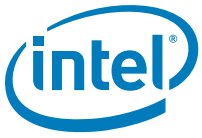
Note: For LKF, for each Intel® ICC test case, the ICC boot profile used by the SUT is checked and only test cases applicable to the currently used boot profile by the SUT are executed. Intel® PETS skips irrelevant tests and does not execute Non applicable test cases.

16.1 Intel® Integrated Clock Control Test Coverage Summary and Details

This section provides summary of ICC test cases applicable to Lake Field based platforms.

Test ID	Test Case Title	Mandatory	PETS/ Manual	Network Factor
ICC_TST_01	Test default settings for Standard configuration	Yes (Only mandatory when SUT's boot profile is selected based on standard profile under FIT or by means of BIOS)	PETS / Manual using ICC SDK embedded	N/A
ICC_TST_02	Test default settings for Adaptive configuration	Yes (Only mandatory when SUT's boot profile is selected based on adaptive profile under FIT or by means of BIOS)	PETS/Manual using ICC SDK embedded	N/A

Test ID	Test Case Title	Mandatory	PETS/ Manual	Network Factor
ICC_TST_04	Test Get and Set of MPHY setting	Yes	PETS/Manual using ICC SDK embedded	N/A



16.2 Intel® Integrated Clock Control Test Cases

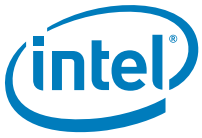
16.2.1 Test Default Settings for Standard Configuration

Test ID:	**ICC_TST_01**
Test Case Title:	Test default settings for Standard configuration
Mandatory/Optional:	Mandatory. ***Note:*** Only for SUTs with boot profile that to "standard" profile under FIT ->ICC -> Boot Profile or by means of BIOS ***Note:*** For FIT Tool, Check parameter under FIT| Integrated Clock Controller | Boot Profile selection. if Boot profile selection is based on Standard profile, then this test is mandatory otherwise it can be skipped. ***Note:*** For BIOS, Check parameter using the request to HECI: ICC_GET_PROFILE_REQ if Boot profile selection is based on Standard profile, then this test is mandatory otherwise it can be skipped.
Description:	Verify if the current ICC registers setting in the SUT are set correctly based on standard configuration
Objective:	Ensure that critical ICC register values are configured correctly for standard configuration.
Procedure:	Get BCLK PLL Settings: • API: ICC_GET_CLOCK_SETTINGSEx • library method: *EXTERNAL_API UINT32IccLibGetCurrentClockSettingsWrapper(const ICC_HECI_CLOCK_ID clockId, ICC_GET_CLOCK_SETTINGSEx * const clockSettings);* An error should be returned in case the test has failed
Test Pass/Fail Criteria:	Pass if the critical ICC registers values read are set correctly based on the standard configuration. Frequency= 400 MHZ SSC = 0.5 ***Note:*** For FIT, Check parameter under Flash Image Tool| Integrated Clock Controller | Boot profile selection. If Boot profile is not based on standard profile then this test is expected to fail. ***Note:*** For BIOS, check parameter using the request to HECI : ICC_GET_PROFILE_REQ if Boot profile is not based on standard profile then this test is expected to fail.

16.2.2 Test Default Settings for Adaptive Configuration

Test ID:	**ICC_TST_02**
Test Case Title:	Test default settings for Adaptive configuration
Mandatory/Optional:	Mandatory ***Note:*** Only for SUTs with boot profile set to "Adaptive" profile under FIT ->ICC -> Boot Profile or by means of BIOS. ***Note:*** For FIT Tool, Check parameter under FIT| Integrated Clock Controller | Boot profile selection. if boot profile selection is based on Adaptive profile, This test is mandatory else the user can skip to execute it. ***Note:*** For BIOS check parameter using the request to HECI : ICC_GET_PROFILE_REQ if Boot profile selection is based on Adaptive profile, then this test is mandatory otherwise it can be skipped.
Description:	Verify if the current ICC registers setting in the SUT are set correctly based on Adaptive configuration

Test ID:	**ICC_TST_02**
Objective:	Ensure that critical ICC register values match defaults for Adaptive configuration
Procedure:	Get BCLK PLL Settings: • API: _ICC_SET_CLOCK_SETTINGSEx • Library method: *EXTERNAL_API UINT32IccLibGetCurrentClockSettingsWrapper(const ICC_HECI_CLOCK_ID clockId, ICC_GET_CLOCK_SETTINGSEx * const clockSettings);* An error should be returned in case the test has failed Set the BCLK PLL settings: • API: _ICC_SET_CLOCK_SETTINGSEx • Library method: EXTERNAL_API UINT32 IccLibSetCurrentClockSettingsWrapper(const ICC_HECI_CLOCK_ID clockId, ICC_SET_CLOCK_SETTINGSEx * clockSettings); An error should be returned in case the test has failed
Test Pass/Fail Criteria:	Pass if the critical ICC registers values read are set correctly based on the Adaptive configuration. ***Note:*** For FIT, Check parameter under Flash Image Tool| Integrated Clock Controller | Boot profile selection. If Boot profile is not based on Adaptive profile then this test is expected to fail. ***Note:*** For BIOS check parameter using the request to HECI : ICC_GET_PROFILE_REQ if Boot profile selection is based on Adaptive profile, then this test is mandatory otherwise it can be skipped. ***Note:*** Default frequency and SSC supported for Adaptive is 400MHz with 0.50%. Supported Min.-Max. frequency range is [390.00- 400.00 MHz]. This test checks default configuration for Adaptive clocking. Test may fail if customer change SSC or frequency from default value; however make sure to check if settings are within the expected range supported for Adaptive clocking.



16.2.3 GET and SET MPHY settings

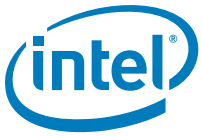
Test ID:	**ICC_TST_04**
Test Case Title:	Get and Set of MPHY setting
Mandatory/Optional:	Mandatory, This is informative test.
Description:	This test output high level detail like CRC count into a bin file , Version and product detail of chipset initialization settings. this test apply a new version of chipset User to manually verify data is correct or not.
Objective:	Verify if correct version of chipset initialization settings are applied or not. In case issue is seen, detail like CRC count, Version and product detail can be used for debug purpose. Apply a new version of chipset initialization settings

Procedure:	GET MPHY Version:
	API:_GET_MPHY_VERSION
	library method: EXTERNAL_API UINT32 IccLibGetMphyVersion(GET_MPHY_VERSION *survTable);
	
	GET MPHY table:
	library method: EXTERNAL_API UINT32 IccLibGetMphySettingsWrapper(UINT32 length, UINT32 offset, UINT8 *buffer,UINT32 *bytesRead);
	
	Set MPHY table:
	library method: EXTERNAL_API UINT32 IccLibSetMphySettingsWrapper(char *mphyFileName);
	
	Note: Retrieving Chipset Initialization file and information can be blocked by some restrictions enforced with End-of-Post being issued. Tester may require to disable End-of-Post message from BIOS menu for the test to successfully pass.
	Note: This test currently displays the command result only.

Test Pass/Fail Criteria:	This is informative test and displays details like CRC count, Version and product detail. User to manually confirm if data looks correct or not.

§ §

(This page is intentionally left blank)



17 Platform Controller Hub (PCH) SoftStrap Configuration

Overview:

The Intel® PCH SoftStraps are load into the appropriate strapping registers within the PCH at boot time from the SPI flash device's Flash Descriptor. Some of the features within the PCH are configurable through the PCH SoftStraps such as the Flexible I/O, SMLINK, GbE, and Intel® ME. The PCH SoftStraps are configure using the FIT tool. Refer to the SPI Programming Guide for the details description on all the available PCH SoftStraps.

All the test case in this chapter are currently cover automatically by PETS on the target system at runtime. Static checking on the image created by FIT is not supported.

Tools for Testing:

Intel® Platform Enablement Test Suite (PETS)—Latest version of tools from this kit. Refer to the Intel® PETS user guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® PETS software.

Intel® Flash Image Tool (FIT.exe)

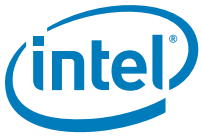
Intel® Flash Programming Tool—Available in DOS (fpt.exe), EFI (fpt.efi), Windows* 32-bit (ftpw.exe), and Windows* 64-bit operating systems.

Test Environment:

The System Under Test (SUT) is to be configured in manual configuration mode a with wired LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured to ensure that the wired LAN and wireless LAN addresses reside on separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

17.1 Test Coverage Summary

Test ID	Test Case Title	PETS/Manual	Network Factor
PSS_003	Flexible I/O Test	PETS	N/A
PSS_004	BIOS Boot-Block Size Test	PETS	N/A
PSS_008	TPM on SPI Test	PETS	N/A



17.2 Flexible I/O Test

<table>
<tr><td>Test ID:</td><td>PSS_003</td></tr>
<tr><td>Test Case Title:</td><td>Flexible I/O Test</td></tr>
<tr><td>Mandatory/Optional:</td><td>Mandatory</td></tr>
<tr><td>Description:</td><td>Flexible I/O is an architecture that allows some high speed signals to be configured as PCIe*, USB 3.x or SATA signals. Through SoftStraps, the functionality on these multiplexed signals are selected to meet I/O needs on the target platform.</td></tr>
<tr><td>Objective:</td><td>To verify correct configuration of Flexible I/O SoftStraps.</td></tr>
<tr><td>Procedure:</td><td>Boot to targeted OS. Verify correct configuration of the PCH SoftStraps below:

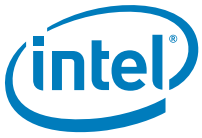
1. Is PCIe Controller 1 Lane Reversed?
<table><tr><th>Name</th><th>Location</th><th>Value</th></tr><tr><td>PCIe Controller 1 Lane Reversal</td><td>Offset 0x131 [2]</td><td>1h</td></tr></table>
— If NOT Reversed:
<table><tr><th>Name</th><th>Location</th><th>Value</th></tr><tr><td>PCIe Controller 1 Lane Reversal</td><td>Offset 0x131 [2]</td><td>0h</td></tr></table>
2. Is PCIe Controller 2 Lane Reversed?
<table><tr><th>Name</th><th>Location</th><th>Value</th></tr><tr><td>PCIe Controller 2 Lane Reversal</td><td>Offset 0x139 [2]</td><td>1h</td></tr></table>
— If NOT reversed:
<table><tr><th>Name</th><th>Location</th><th>Value</th></tr><tr><td>PCIe Controller 2 Lane Reversal</td><td>Offset 0x139 [2]</td><td>0h</td></tr></table></td></tr>
</table>

17.3 BIOS Boot-Block Size Test

<table>
<tr><td>Test ID:</td><td>PSS_004</td></tr>
<tr><td>Test Case Title:</td><td>BIOS Boot-Block size Test</td></tr>
<tr><td>Mandatory/Optional:</td><td>Mandatory (SPI Configurations Only)</td></tr>
<tr><td>Description:</td><td>BIOS Boot-Block size deals with a BIOS recovery mechanism. If this is not set correctly, then BIOS boot-block recovery mechanism will not work.</td></tr>
<tr><td>Objective:</td><td>To verify BIOS boot-block size of correctly setup.</td></tr>
<tr><td>Procedure:</td><td>
Boot to targeted OS. Verify correct configuration of the PCH SoftStraps below:

1. What size is SPI flash BIOS boot block?

a. If 64KB
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>0h</td></tr>
</table>
b. 128KB
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>1h</td></tr>
</table>
c. 256KB
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>2h</td></tr>
</table>
d. 512KB
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>3h</td></tr>
</table>
e. 1MB
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>4h</td></tr>
</table>
</td></tr>
<tr><td>Test Pass/Fail Criteria:</td><td>Test passes if SoftStraps/register setting in this step matches to the configuration in the target system.</td></tr>
</table>



17.4 Trusted Platform Module (TPM) on SPI Test

<table>
<tr><td>Test ID:</td><td>PSS_008</td></tr>
<tr><td>Test Case Title:</td><td>Trusted Platform Module on SPI Test</td></tr>
<tr><td>Mandatory/Optional:</td><td>Mandatory</td></tr>
<tr><td>Description:</td><td>TPM can be configured through PCH SoftStraps to operate over LPC or SPI, but no more than 1 TPM is allowed in the target system.</td></tr>
<tr><td>Objective:</td><td>To verify TPM on SPI is correctly configured.</td></tr>
<tr><td>Procedure:</td><td>
Boot to targeted OS. Verify correct configuration of the PCH SoftStraps below:

1. Does this platform have a TPM connected to SPI controller?
— If YES, Skip to Boot to targeted OS testing step.
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>TPM Over SPI Bus Enable</td><td>Offset 0x160 [0] A-Step
Offset 0x168 [0] B-Step</td><td>1h
1h</td></tr>
</table>
— If NO (default),
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>TPM Over SPI Bus Enable</td><td>Offset 0x160 [0] A-Step
Offset 0x168 [0] B-Step</td><td>0h
0h</td></tr>
</table>
Boot to targeted OS. Verify correct configuration of the PCH SoftStraps below:

1. What Clock Frequency is being used for TPM on SPI?
a. If 48MHz
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>SPI TPM Clock Frequency</td><td>Offset 0x119 [2:0]</td><td>2h</td></tr>
</table>
b. 30MHz
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>SPI TPM Clock Frequency</td><td>Offset 0x119 [2:0]</td><td>4h</td></tr>
</table>
c. 17MHz
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>SPI TPM Clock Frequency</td><td>Offset 0x119 [2:0]</td><td>6h</td></tr>
</table>
</td></tr>
<tr><td>Test Pass/Fail Criteria:</td><td>Test passes if SoftStraps/register setting in this step matches to the configuration in the target system.</td></tr>
</table>

§ §

(This page is intentionally left blank)

18 Dekel PHY FW Compliance

18.1 Background

Dekel PHY AKA SPHY is a PHY firmware located on the PCH for Type-C and Multi PHY usages.In LKF there are 4 Dekel PHY controllers and each of them needs firmware to operate, The 4 PHY controllers are UFS, PCIe*, and the 2 USB Type-C ports. These FW images reside on the system boot NVM (e.g. where BIOS and CSE FW are located), Intel® CSE Firmware loads and authenticates the images directly into the PHY IP from the system NVM and starts the FW executing on the PHY controller by performing a sideband WR to each respective PHY.

18.2 Scope

This chapter is intended to describe SPHY tests use cases for USB type C, PCIe* and UFS.

18.3 Tools for Testing

Intel® Automated Power Switch – The SUT should be connected to an Intel® APS 3 unit. In case Intel® APS 3 is not available, select the Manual configuration in the Intel® PETS SUT profile configuration menu.

Intel® Platform Enablement Test Suite (PETS)—Latest version of tools from this kit. Refer to the Intel® PETS user guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® PETS.

Intel® MEInfo - Latest version of tool from this KIT

Note: Intel® PETS package will be available on a later version of thetool.In the meantime, please perform these tests manually.

18.4 Dekel PHY FW Compliance Test Coverage Summary

Test ID	Test Case Title	PETS Package Name	OS Support	How?
SPHY_01	successful DKL loading post Warm Reset	compliance _ME_SPHY	W	I
SPHY_02	successful DKL loading post Cold Reset	compliance _ME_SPHY	W	I
SPHY_03	successful DKL loading post Global Reset	compliance _ME_SPHY	W	I
SPHY_04	successful DKL loading post S5 state	compliance _ME_SPHY	W	I
SPHY_05	successful DKL loading post G3 state	compliance _ME_SPHY	W	I
SPHY_06	Boot from UFS	compliance _ME_SPHY	W	I

OS Support: W = Microsoft Windows*

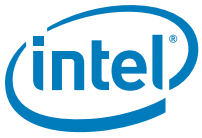
How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

18.5 Test SPHY_01

Test ID:	SPHY_01
Test Case Title:	successful DKL PHY FW loading post Warm Reset
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	send warm reset command and ensure type-C/PCIe functionality
Test Pass Criteria:	successful boot, connect/disconnect detection and file copy are working properly
Description:	the test confirm Type-C and PCIe functionality
Procedure:	1. perform Warm Reset 2. boot to OS 3. run *-MeInfo* command SPHY version should be in format MM.xxx.yyy.ZZZZ 4. connect type-C device 5. verify type-C device is connected 6. copy a file to the connected type-C device 7. Disconnect type-C device 8. verify type-C device is disconnected 9. connect PCIe device 10. verify PCIe device is connected 11. copy a file to the connected PCIe device 12. Disconnect PCIe device 13. verify PCIe device is disconnected

18.6 Test SPHY_02

Test ID:	SPHY_02
Test Case Title:	successful DKL PHY FW loading post Cold Reset
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	send cold reset command and ensure type-C/PCIe functionality
Test Pass Criteria:	successful boot, connect/disconnect detection and file copy are working properly
Description:	the test confirm Type-C and PCIe functionality
Procedure:	1. Perform cold Reset 2. Boot to OS 3. Run *-MeInfo* command SPHY version should be in format:MM.xxx.yyy.ZZZZ 4. Connect type-C device 5. Verify type-C device is connected 6. Copy a file to connected type-C device 7. Disconnect type-C device 8. Verify type-C device is disconnected 9. Connect PCIe device 10. Verify PCIe device is connected 11. Copy a file to connected PCIe* device 12. Disconnect PCIe device 13. Verify PCIe device is disconnected



18.7 Test SPHY_03

Test ID:	SPHY_03
Test Case Title:	Successful DKL PHY PW loading post Global Reset
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	send global reset command and ensure type-C/PCIe functionality
Test Pass Criteria:	successful boot, connect/disconnect detection and file copy are working properly
Description:	the test confirm Type-C and PCIe functionality
Procedure:	1. perform Global Reset 2. boot to OS 3. run -*MeInfo* command SPHY version should be in format:MM.xxx.yyy.ZZZZ 4. connect type-C device 5. verify type-C device is connected 6. copy a file to connected type-C device 7. Disconnect type-C device 8. verify type-C device is disconnected 9. connect PCIe device 10. verify PCIe device is connected 11. copy a file to connected PCIe device 12. Disconnect PCIe device 13. verify PCIe device is disconnected

18.8 Test SPHY_04

Test ID:	SPHY_04
Test Case Title:	successful DKL PHY FW loading post S5
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	send machine to S5 state, resume and ensure type-C/PCIe functionality
Test Pass Criteria:	successfully resume, connect/disconnect detection and file copy are working properly
Description:	the test confirm Type-C and PCIe functionality
Procedure:	1. send machine to S5 state 2. boot to OS 3. run -*MeInfo* command SPHY version should be in format:MM.xxx.yyy.ZZZZ 4. connect type-C device 5. verify type-C device is connected 6. copy a file to connected type-C device 7. Disconnect type-C device 8. verify type-C device is disconnected 9. connect PCIe device 10. verify PCIe device is connected 11. copy a file to connected PCIe device 12. Disconnect PCIe device 13. verify PCIe device is disconnected

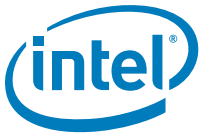
18.9 Test SPHY_05

Test ID:	SPHY_05
Test Case Title:	successful DKL PHY FW loading post G3
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	send machine to G3 state, resume and ensure type-C/PCIe functionality
Test Pass Criteria:	successfully resume, connect/disconnect detection and file copy are working properly
Description:	the test confirm Type-C and PCIe functionality
Procedure:	1. send machine to G3 state 2. boot to OS 3. run *-MeInfo* command SPHY version should be in format:MM.xxx.yyy.ZZZZ 4. connect type-C device 5. verify type-C device is connected 6. copy a file to connected type-C device 7. Disconnect type-C device 8. verify type-C device is disconnected 9. connect PCIe device 10. verify PCIe device is connected 11. copy a file to connected PCIe device 12. Disconnect PCIe device 13. verify PCIe device is disconnected

18.10 Test SPHY_06

Test ID:	SPHY_06
Test Case Title:	Boot from UFS
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	success boot from UFS
Test Pass Criteria:	successful boot
Description:	the test confirm boot from UFS functionality
Procedure:	1. boot the machine a UFS device 2. with run *-MeInfo* command SPHY version should be in format:MM.xxx.yyy.ZZZZ

§ §



(This page intentionally left blank.)

19 Embedded Controller Lite FW Compliancy

19.1 Introduction

EC-less design shall provide a solution to the system implementer to remove the dependency of a discrete EC on a platform. The solution is enabled by offloading the EC functionalities (Battery management, thermal management, Type C policy manager) on to the SoC (here ISH as an offloading engine) and the FW which implements the EC functionality is referred as EC Lite.

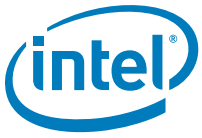
The implementation of an EC-less design will help to reduce the BOM area/cost for the overall solution. This solution is ideal for the smaller form factors which has constraint on the BOM and the area.

19.2 Test Environment Setup

The System Under Test (SUT) is to be configured in manual configuration mode a with wired LAN or wireless LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured so that the wired LAN and wireless LAN addresses can reside on same or separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

19.3 Tools for Testing

- **Intel® Platform Enablement Test Suite** - Latest version of the tool is available in the Intel® CSE compliancy kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.



19.4 EC LiteFW Compliancy Test Coverage Summary

Operating System Support and How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Support	How?
EC 1.0.0	PM - System power state S0 - Warm reboot	N/A	Yes	M
EC 1.0.1	PM - System power state S5 – shutdown	N/A	YES	M
EC 1.0.2	PM - S4 state entry and exit	N/A	Yes	M
EC 1.0.3	PM - Power Standby - Sleep and Wake	N/A	Yes	M
EC 1.0.4	TCSS Wake Support from Sx/S0ix	N/A	Yes	M
EC 1.0.5	TCSS Functionality after Sx/S0ix/Reset	N/A	Yes	M
EC 2.0.0	USB Type-C Devices Re-enumeration after system power state transition.	N/A	Yes	M
EC 3.0.0	Boot – verify firmware information	N/A	Yes	M

19.5 Test EC 1.0.0

Test ID:	EC 1.0.0
Test Case Title:	PM - System power state S0 - Warm reboot
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	To check out the warm reboot flow
Test Pass Criteria:	After Successful restart, all the USB-C peripherals should work fine, and battery and thermal manager should be brought up as expected.
Description:	The purpose of this test is to ascertain the EC lite controlled components such as battery and thermal manager, and TCSS are brought up normally following a warm reboot.
Procedure:	Perform a system restart by the different possible ways: 1. OS Interface 2. Reset Button on the board. Check whether battery is connected and showing correct a valid battery % remaining. Also check whether all USB-C devices are coming up as expected. Check for a valid CPU temperature in the Windows* Performance Monitor application.

19.6 Test EC 1.0.1

Test ID:	EC 1.0.1
Test Case Title:	PM - System power state S5 – shutdown
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	To check out the EC Lite component shutdown flow
Test Pass Criteria:	After successful restart, all the peripherals should work fine. Ensure that all USB-C connected devices, thermal monitor and battery manager are coming up correctly after power on.
Description:	This test makes sure the system is already booted to OS, and initiates system shutdown.
Procedure:	Initiate a system shutdown via any of the shutdown entry methods as mentioned below: a. System shutdown gracefully using OS UI. b. System shutdown using power button. c. System shutdown abruptly after removing the power supply in system active condition. Some variations of this use case which are OS specific must also be verified. For example, in Windows*: 1. With Connected Standby and 10sec Power Button Override enabled in BIOS, check for graceful shutdown using slider UI. 2. Shutdown with Fast Startup enabled (default)/Disabled in Windows* Power option. After shutdown, system should be able to startup correctly using power button press. Ensure all USB-C devices, thermal manager and battery manager come up correctly after system resumes from shutdown (S5).

19.7 Test EC 1.0.2

Test ID:	EC 1.0.1
Test Case Title:	PM - System power state G3 – shutdown
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	To check out the EC Lite component shutdown flow
Test Pass Criteria:	After successful restart, all the peripherals should work fine. Ensure that all USB-C connected devices, thermal monitor and battery manager are coming up correctly after power on.
Description:	This test makes sure the system is already booted to OS, and initiates system shutdown.
Procedure:	Initiate a system shutdown via shutting down the system abruptly after removing the power supply in system active condition. Some variations of this use case which are OS specific must also be verified. For example, in Windows*: 1. With Connected Standby and 10sec Power Button Override enabled in BIOS, check for graceful shutdown using slider UI. 2. Shutdown with Fast Startup enabled (default)/Disabled in Windows* Power option. After shutdown, system should be able to startup correctly using power button press. Ensure all USB-C devices, thermal manager and battery manager come up correctly after system resumes from shutdown (G3).

19.8 Test EC 1.0.3

Test ID:	EC 1.0.2
Test Case Title:	PM - S4 state entry and exit
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	To check out the S4 state entry and exit flow
Test Pass Criteria:	After successful OS bring up from hibernate, all the peripherals should work fine.
Description:	This test makes sure the system is already booted to OS, initiates system hibernate, and ensures system is brought up as expected post hibernate.
Procedure:	Perform a system hibernate via the OS and ensure system boots up to OS after resuming from hibernate, all USB-C connected devices are enumerated in device manager as expected, and thermal and battery manager are brought up in OS as expected.

19.9 Test EC 1.0.4

Test ID:	**EC 1.0.3**
Test Case Title:	PM - Power Standby - Sleep and Wake
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	To check out the system sleep and wake flow
Test Pass Criteria:	After successful wake from sleep state, all the USB-C peripherals, battery and thermal manager apps should function as expected.
Description:	The purpose of this test is to ensure that the system can enter low power mode covering all the ways possible and once in low power mode the system can still process external events that require it to come back to active mode via all possible wake-up events.
Procedure:	The system should be transitioned into a Low Power Sleep State in one of the following ways: 1. User performs a short press of the power button to force the platform to go to low power Sleep state if the power button press is configured appropriately. 2. System is inactive and goes to low power state after a pre-configured time as decided in the Power options 3. User closes the LID of the System Upon Sleep, the System can either go into Standby (S3) or Modern Standby (CMS, DMS) based on the bios configurations, platform BOM etc. Once the platform is in low power mode, ensure that it can wake up and boot to OS as expected. Ensure that all USB-C connected devices are enumerated correctly once system resumes from S5.

19.10 Test EC 1.0.5

Test ID:	EC 1.0.4
Test Case Title:	TCSS Wake Support from Sx/S0ix
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	To ensure that the system should support wake from Sx or S0ix through devices which are connected via Type-C Port.
Test Pass Criteria:	After successful wake from Sx or S0ix state, all the peripherals connected via USB Type-C port should be enumerated as expected.
Description:	The purpose of this test is to ensure that the system should support wake from Sx or S0ix through devices which are connected via USB-C Port.
Procedure:	Transition the system into a Low Power Sleep State in one of the following ways: 1. User performs a short press of the power button to force the platform to go to low power Sleep state if the power button press is configured appropriately. 2. System is inactive and goes to low power state after a pre-configured time as decided in the Power options 3. User closes the LID of the System Upon Sleep, the System can either go into Standby (S3) or Modern Standby (CMS, DMS) based on the BIOS configurations, platform BOM etc. Once the platform is in low power mode, ensure that it can wake up via a selected set of asynchronous events.

19.11 Test EC 1.0.6

Test ID:	EC 1.0.5
Test Case Title:	TCSS Functionality after Sx/S0ix/Reset
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	The objective of this test is to verify Type-C Device functionality before and after S3, S4, S5, Deep S4, Deep S5, S0ix, warm reset and cold reset.
Test Pass Criteria:	Before and after successful wake from S4, S5, deep S4, deep S5, S0ix, warm reset, cold reset, all the peripherals connected via USB Type-C port should work as expected.
Description:	The purpose of this test is to ensure that the USB type-C devices should function as expected before and after wake from S4, S5, deep S4, deep S5, S0ix, warm reset and cold reset.
Procedure:	Ensure that all peripherals connected to USB type-C port are working fine before and after the system is transitioned into S4, S5, deep S4, deep S5, S0ix, warm reset and cold reset in one of the following ways: 1. User performs a short press of the power button to force the platform to go to low power Sleep state if the power button press is configured appropriately. 2. System is inactive and goes to low power state after a pre-configured time as decided in the Power options 3. User closes the LID of the System Upon Sleep, the System can either go into Standby (S3) or Modern Standby (CMS, DMS) based on the bios configuration, platform BOM etc. Once the platform is in low power mode, ensure that it can wake up via a selected set of asynchronous events, such as moving the mouse pointer or a keyboard strike.

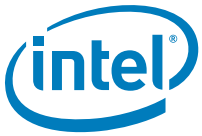
19.12 Test EC 2.0.0

Test ID:	EC 2.0.0
Test Case Title:	USB Type-C Devices Re-enumeration after system power state transition.
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	The objective of this test is to verify enumeration of USB type-C devices after system wakes from S4, S5, deep S4, deep S5 and S0ix.
Test Pass Criteria:	Before and after successful wake from S4, S5, deep S4, deep S5, S0ix, warm reset, cold reset, all the peripherals connected via USB Type-C port should enumerate as expected.
Description:	The purpose of this test is to ensure that the USB type-C devices should enumerate correctly after wake from S4, S5, deep S4, deep S5, S0ix, warm reset and cold reset.
Procedure:	Ensure that all peripherals connected to USB type-C port are enumerated after the system is transitioned into S4, S5, deep S4, deep S5, S0ix, warm reset and cold reset in one of the following ways: 1. User performs a short press of the power button to force the platform to go to low power Sleep state if the power button press is configured appropriately. 2. System is inactive and goes to low power state after a pre-configured time as decided in the Power options 3. User closes the LID of the System Upon Sleep, the System can either go into Standby (S3) or Modern Standby (CMS, DMS) based on the bios configurations, platform BOM etc. Once the platform is in low power mode, ensure that it can wake up via a selected set of asynchronous events.

19.13 Test EC 3.0.0

Test ID:	EC 3.0.0
Test Case Title:	Boot – verify firmware information
Platform:	LKF
Mandatory/Optional:	Mandatory
Objective:	The goal of this test case is to validate firmware is able to report information about platform configuration at startup.
Test Pass Criteria:	The platform configuration reported at startup is as expected.
Description:	The purpose of this test is to ensure that the firmware is able to report information about platform configuration at startup.
Procedure:	Navigate to the BIOS menu and ensure the firmware (IFWI) version is reported correctly in the platform information menu.

§ §



(This page is intentionally left blank)