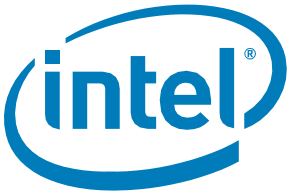


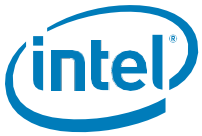
Firmware Interface Table

BIOS Specification

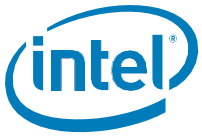
April 2020

Revision 1.2

Document 599500

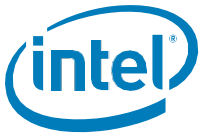


Contents



Revision History

Revision	Description	Revision Date
1.2	• Updated CSE Secure Boot Rules in section 4.12	April 2020
1.1	• Merged 338505 content into new ID 599500 • Added FIT Entry for Diagnostic ACM Rules in section 4.5 • Updated base formatting and stylesheet	January 2020
001	• Updated Microcode Update Rules in section 4.3 • Initial publication	November 2018
0.70	• Initial release	October 2016



1.0 Firmware Interface Table Introduction

This document provides a high-level overview of the Firmware Interface Table.

A Firmware Interface Table (FIT) is a data structure inside BIOS flash and consists of multiple entries. Each entry defines the starting address and attributes of different components in the BIOS. FIT resides in the BIOS Flash area and is located by a FIT pointer at physical address (4GB - 40h), refer to Figure below. The FIT is generated at build time, based on the size and location of the firmware components.

The CPU processes the FIT before executing the first BIOS instruction located at the reset vector (0FFFFFFF0h). If a microcode update for the BSP is pointed by a FIT type 1 entry, it is loaded before executing the BIOS code at the reset vector, and applied to all threads within the package.

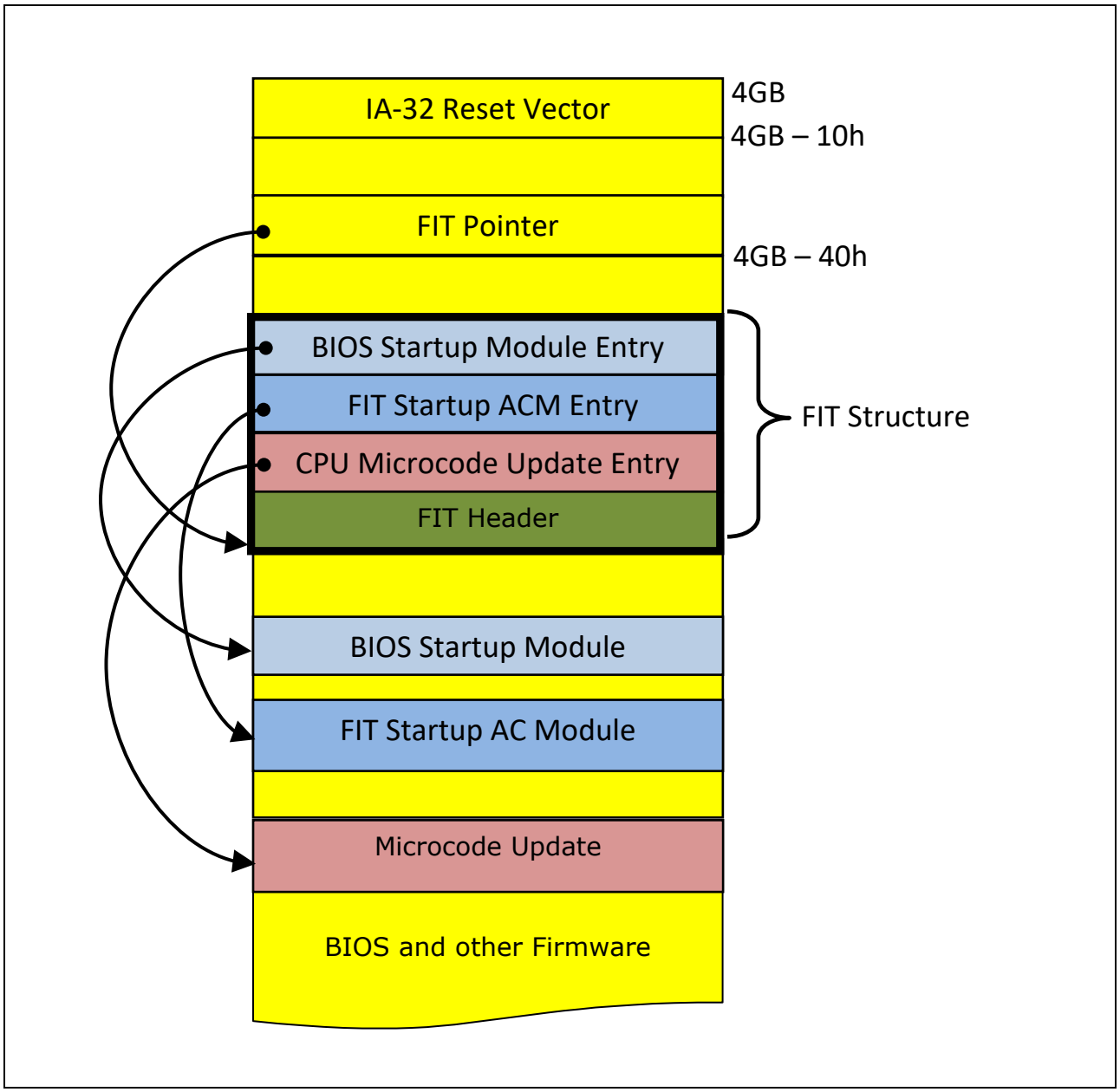
Refer to the CPU BIOS specification for model specific Microcode Update Loading guidance.

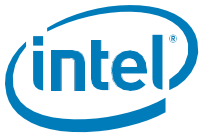
The FIT boot is a method the processors use to establish a root of trust for the BIOS. If a valid type 2 entry is found, then that startup ACM is executed. For detailed information regarding the ACM, refer to the Intel® Trusted Execution Technology BIOS Specification BIOS Writer's Guide.

2.0 Summary of Key Requirements:

- The BIOS flash must include a Firmware Interface Table (FIT) with Type 0 (FIT Header) and Type 1 (Microcode Update) entries.
- A microcode update must exist for every processor stepping supported by the platform.

Figure 2-1. FIT Layout in Flash / ROM





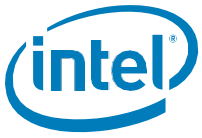
3.0 FIT Pointer

The processor locates the FIT by following the FIT Pointer, which is located at the fixed address 4 GB - 40h (i.e., 0FFFFFFC0h). The FIT pointer points to the first byte of the Header (type 0) entry in the FIT.

3.1 FIT Pointer Rules

The physical location of FIT in firmware address space must satisfy the following requirements:

1. The entire FIT table must reside within the firmware address range of (4 GB to 16 MB) to (4 GB-40h). If the FIT is located outside this region, the processor will invoke a legacy boot process and a root of trust will not be established using FIT.
2. FIT must also reside within the firmware address region that is accessible by the hardware upon CPU reset. Any initialization done by system service processors present on the platform prior to invocation of CPU reset is permitted.
3. With introduction of Boot Guard Technology, FIT table is no longer optional element of architecture. CPU behavior, if FIT table is absent or invalid depends on platform provisioning and will range from immediate unbreakable shutdown and up to fallback to time-limited legacy boot followed by a platform reset, when timeout expires.
4. If FIT table is used by a platform, and Top Swap is used as a method of platform recovery, the need to keep top and swap blocks updated can be avoided, if FIT resides outside of boot and recovery areas. It has to be noted, such that flash layout is suitable only for platform not employing Boot Guard, which has own rules of Top Swap implementation.



4.0 Firmware Interface Table

Each entry in the Firmware Interface Table is 16 bytes in length. A valid entry will contain all the required fields as defined below.

Note: It is recommended to place FIT at a fixed address in the BIOS. This will help making FIT Pointer static.

Table 1. FIT Entry Format

Byte Offsets	15	14	13:12	11	10:8	7:0
Meaning	Chksum	Bit 7 - C_V Bits 6:0 - Type	Version	Reserved	Size	Address

ADDRESS - Address is the base address of the firmware component and must be aligned on 16-byte boundary.

SIZE - Size is the span of the component in multiple of 16 bytes.

VERSION - Version contains the component's version number in binary coded decimal (BCD) format. For the FIT header entry, the value in this field will indicate the revision number of the FIT data structure. The upper byte of the revision field indicates the major revision and the lower byte indicates the minor revision. The format 0x1234 conveys the major number encoded in the first two digits and the minor number in the last two with a fixed point assumed in between.

C_V - Checksum Valid bit. This is a one bit field that indicates, whether component has a valid checksum. CPU must ignore CHKSUM field, if C_V bit is not set.

TYPE - 7 bit field containing the type code for the component registered in the FIT table. The type field encoding is defined in Table below.

CHKSUM - 1 byte field containing the component's checksum. The modulo sum of all the bytes in the component and the value in this field (CHKSUM) must add up to zero. This field is only valid, if the C_V flag is non-zero. Support for checksum is optional.

RESERVED: All reserved bit fields must be set to 0.

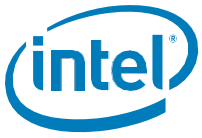


Table 2. FIT Entry Type Field Definitions

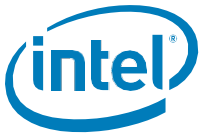
FIT Entry Type	Description (Intel® 64 and IA-32 Architectures)
0x00	FIT Header Entry
0x01	Microcode Update Entry
0x02	Startup AC Module Entry
0x03	Diagnostic AC Module Entry
0x04 to 0x06	Intel Reserved
0x07	BIOS Startup Module Entry
0x08	TPM Policy Record
0x09	BIOS Policy Record
0x0A	TXT Policy Record
0x0B	Key Manifest Record
0x0C	Boot Policy Manifest
0x0D - 0x0F	Intel Reserved
0x10	CSE Secure Boot
0x11 - 0x2C	Intel Reserved
0x2D	Feature Policy Delivery Record
0x2E	Intel Reserved
0x2F	JMP $ Debug Policy
0x30 - 0x70	Reserved for Platform Manufacturer Use
0x71 - 0x7E	Intel Reserved
0x7F	Unused Entry (skip)

Intel Reserved Entries are reserved for Intel usage only.

Platform Manufacturer Use Reserved Entries are reserved for Platform Manufacturer specific usage. These entries are not checked by the processor.

Unused Entry (Type 0x7F) - FIT Type "0x7F" means "Unused entry". "Unused entry" refers to FIT entry that exists but has no meaningful contents. It serves as a "reserved" or an "invalid" entry, which may be updated later or had been invalidated without having to change the total number of FIT entries (like a deleted record).

The FIT processing code always skips the unused entry and moves on to the next record.



4.1 FIT Ordering Rules

The following rules must be satisfied by the firmware, while populating FIT with various component records.

1. The records must always be arranged in the ascending order of their type attribute in the FIT. (Example: The first record must be of Type 0).
2. For certain types, it is acceptable to have multiple records in the FIT of that Type. Refer to the rules regarding individual record types.

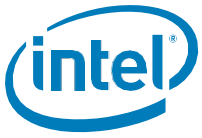
Note: The ACM is NOT required to check or ensure that the entry types are in order.

4.2 FIT Header (Type 0) Rules

1. The very first entry in FIT must be a Type 0 entry, FIT Header Entry. There can be exactly one Type 0 entry in FIT.
2. The address field must contain the ASCII value of "_FIT_<sp> <sp> <sp>", where
3. <sp> is space character (20h).
4. If C_V bit in this entry is set, FIT table must checksum to 0.
5. Size field represents the size of the FIT table in multiple of 16 bytes.
6. Version field should be set to 0x0100.

4.3 Microcode Update (Type 1) Rules

1. At least one Microcode Update (Type 1) Entry is required. There can be one or more Microcode Update Entries in the FIT.
2. BIOS may carry multiple Microcode Updates for multiple processor stepping support. Each Type 1 entry points to a distinct Microcode Update. Each Microcode Update includes a header followed by update data, which may be followed by Extension Signature Table. The address field in Type 1 entry points to the first byte of the Microcode Update Header.
3. Each Type 1 entry must point to an address that is accessible by the processor at reset (i.e., requires no chipset configuration to reach that address in the flash).
4. BIOS may have some empty Microcode Update slots. These slots are set aside by BIOS to store future Microcode Updates. It is suitable for a Type 1 entry to point to these empty slots as long as the first dword in the empty slot is 0xFFFF_FFFF.
5. For a given processor stepping, multiple revisions of Microcode Updates may be released over time. The FIT can contain more than one Type 1 entry for a processor signature and Platform ID combination for recovery considerations. The processor will load the latest available microcode update by choosing the one that has higher revision ID. To comply with the microcode update requirement that BIOS must ensure the latest Microcode Update is loaded after a recovery.
6. Microcode updates pointed to by a type 1 entry must be aligned on a 16-byte address.
7. Microcode updates pointed to by a type 1 entry must not be compressed, encoded or encrypted by the BIOS.
8. The C_V bit in this entry should be clear to 0.
9. The Size field is not used. BIOS should clear this field to 0.



4.4 Startup ACM (Type 2) Rules

1. At least one Startup ACM (Type 2) Entry in the FIT is required for FIT boot support.
2. The address field points to a Startup ACM. Specifically, the address field in the type 2 record points to the first byte of the ACM header.
3. Type 2 entry must point to an address that is accessible by the processor at reset vector.
4. Internal to the processor, one MTRR base/limit pair is used to map Startup AC module. This places alignment restrictions on the Startup ACM. The MTRR size (called MTRR_Size) must be a power of 2 and the base (MTRR_Base) must be a multiple of MTRR size. The following equation defines MTRR_Size.

Equation 1 .MTRR Size Calculation

```
MTRR_Size = 2**(ceiling(log2( Startup_ACM_size )))
```

Note: Ceiling is a mathematical function. Ceiling(x) returns the smallest integer x larger than x

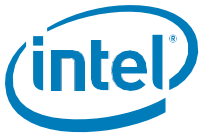
Example: If the size of Startup ACM is 13 k, MTRR_Size is 16k (the next power of 2), and Startup ACM must be aligned on 16k boundary.

5. ACM may be smaller than size of allocated Authenticated Code Execution Area (ACEA) computed by the above formula. ACEA completely obscures flash part at addresses occupied by itself, therefore no objects that ACM needs to reach must be located in this obscured area. This includes FIT and all objects pointed to by FIT records.
6. The C_V bit in this entry should be clear.
7. The Size field is not used. BIOS should set this field to 0.
8. The Version field should be set to 0x0100.

4.5 Diagnostic ACM (Type 3) Rules

Diagnostic ACM (Type 3) entry in the FIT is required only for platforms which support functional safety.

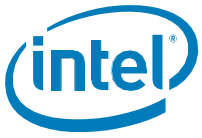
1. The address field points to a Diagnostic ACM. Specifically, the address field in the type 3 record points to the first byte of the ACM header.
2. Type 3 entry must point to an address that is accessible by the processor at reset vector and should be 4Kb aligned.
3. The C_V bit in this entry should be clear.
4. The Size field is not used. BIOS should set this field to 0.
5. The Version field should be set to 0x0100.



4.6 BIOS Startup Module (Type 7) Rules

Record Types 7 is used by legacy Intel® TXT FIT boot only and is not needed, if latter is not used.

1. There can be zero or more BIOS Startup Module Entries in the FIT. For FIT boot, support with BPT do not have to include Type 7 entry. Otherwise, at least one BIOS Startup Module Entry in the FIT is required for FIT boot support.
2. For purpose of this specification, BIOS Startup code is defined as the code that gets control at reset vector and continues the chain of trust in TCG compliant fashion. In addition, this code may also configure memory and SMRAM.
3. In order to enable more flexible flash layout, BIOS Init code can be split in multiple BIOS Startup Modules. Each BIOS Startup Module will have one corresponding Type 7 entry. Each Type 7 entry describes address and size of the corresponding BIOS Startup Module.
4. Each Type 7 entry must point to an address that is accessible by the processor at reset vector. The address should be within the low 4 GB of address space.
5. At least one BIOS Startup Module must encompass Reset vector.
6. At least one BIOS Startup Module must encompass FIT pointer.
7. BIOS Startup Module should not encompass Type 0x9 record, if signature verification mechanism is used.
8. Various BIOS Startup Modules cannot overlap with each other.
9. None of the BIOS Startup Module can overlap with Startup ACM (refer to Section 4.4).
10. The C_V bit in this entry should be clear to 0.
11. The Size field indicates the size of the BIOS startup module in 16-byte multiples. Example: Value of 0x1000 indicates a 64 KB BIOS Startup Module.
12. The Version field should be set to 0x0100.



4.7 TPM Policy Record (Type 8) Rules

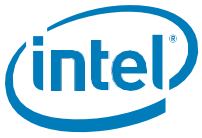
Record Types 8 is used by legacy Intel® TXT FIT boot only and is not needed, if latter is not used.

1. There can be zero or one TPM Policy Record in the FIT.
2. Each Type 8 entry is at an address that is accessible by the processor at reset vector.
3. The address field contains the TPM_POLICY_PTR structure. This structure contains the address, where the TPM Policy information resided.
4. The version field is set to 0, if TPM_POLICY_PTR describes an Indexed IO type pointer. The version field is set to 1, if TPM_POLICY_PTR describes a flat memory pointer.
5. If indexed IO type pointer is used, the Address field holds a structure of the type INDEX_IO_ADDRESS. This structure contains the IO addresses of the index and data register, access width and position of the bit that holds the TPM policy.
6. If flat memory type pointer is used, the Address field holds a 64-bit memory address. The memory address should be within the low 4 GB of address space. Bit 0 at this address holds the TPM Policy.
7. The TPM policy says whether TPM should be enabled or disabled. If TPM Policy = 0, the TPM should be disabled. If TPM Policy is 1, the TPM should be enabled.
8. The default setting is 1. In other words, if this structure is not present or is invalid, the Startup ACM will behave as if TPM Policy = 1.
9. The C_V bit in this entry should be clear to 0.
10. The Size field is not used. BIOS should set this field to 0.

4.8 BIOS Policy Data Record (Type 9) Rules

Record Types 9 is used by legacy Intel® TXT FIT boot only and is not needed, if latter is not used. The BIOS policy is stored in the TPM.

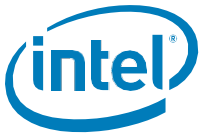
1. There can be zero or one type 9 Record in FIT. A Type 9 entry contains the BIOS policy data. If the platform uses Hash Comparison method and employs fail-safe bootblock, one Type 9 entry is needed, and it contains the fail-safe hash. If the platform uses Signature verification method, one Type 9 entry is needed. In this case, Type 9 entry contains the OEM key, hash of the BIOS and signature over the hash using OEM key. In all other cases, Type 9 entry is not required and should not be implemented.
2. Type 9 entry must point to an address that is accessible by the processor at reset vector. The memory address should be within the low 4 GB of address space.
3. The address must point to the LCP_POLICY_DATA structure.
4. The Version field of Type 9 entry should be set to 0x0100.
5. The C_V bit in this entry should be clear.
6. The Checksum field is set to 0.
7. LCP_POLICY_DATA is a variable length data structure. The size field in a Type 9 entry specifies the size of LCP_POLICY_DATA data structure. Elements of LCP_POLICY_DATA data structure contains enough information to compute the length of LCP_POLICY_DATA data structure. The length of LCP_POLICY_DATA computed using a Type 9 entry must match the length computed using fields within LCP_POLICY_DATA



4.9 Intel® TXT Policy Data Record (Type 0x0A) Rules

There can be zero or one Intel® TXT Configuration Policy Record in the FIT.

1. If there are zero records of this type Intel® TXT state defaults to be in ENABLED state. In other words, this record needs to be provided only, if OEM needs to offer Intel® TXT configuration disable feature without changing any bits in FIT, FIT pointer and/or BIOS ACM.
2. Each Type 0x0A entry in an address that is accessible by the processor at reset vector.
3. The address field contains the TXT_CONFIG_POLICY_PTR structure. This structure contains the address, where the TXT Configuration Policy information resides. (Refer section 4.9.1)
4. The version field is set to 0, if TXT_CONFIG_POLICY_PTR describes an Indexed IO type pointer. The version field is set to 1, if TXT_CONFIG_POLICY_PTR describes a flat memory pointer.
5. If indexed IO type pointer is used, the Address field holds a structure of the type INDEX_IO_ADDRESS. This structure contains the IO addresses of the index and data register, access width and position of the bit that holds the Intel® TXT policy (refer to Table 3).
6. The indexed IO location must be accessible at reset without any hardware initialization.
7. If flat memory type pointer is used, the Address field holds a 64-bit memory address. The memory address should be under 4 GB. Bit0 at this address holds the Intel® TXT Configuration Policy (refer to Table 4).
8. The Intel® TXT Config policy says, whether Intel® TXT should be enabled or disabled. If Intel® TXT Policy =0, Intel® TXT should be disabled. If Intel® TXT Configuration Policy is 1, Intel® TXT should be enabled.
9. The default setting is 1. In other words, if this structure is not present or is invalid, the Startup ACM will behave, as if TXT Config Policy = 1.
10. The C_V bit in this entry should be cleared to 0.
11. The Size field is not used. BIOS should set this field to 0.



4.9.1 Intel® TXT Enable Disable

```
typedef struct {
    UINT16
    IndexRegisterAddress;
    UINT16
    DataRegisterAddress;
    UINT8 AccessWidthInBytes; // 1=1 byte acceess, 2=2 byte
    access UINT8 BitPosition; // Bit number, 15=> Bit 15
    UINT16 Index;
    } INDEX_IO_ADDRESS;
typedef union {
    UINT64
    FlatMemoryAddress;
    INDEX_IO_ADDRESSIndexI
    o;
    } TXT_POLICY_PTR;
```

Table 3. Intel® TXT Configuration Policy Entry Version = 0, Indexed IO TypePointer

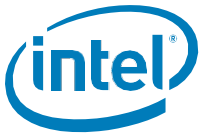
CHKESUM 1 byte	C_V=0 1 bit	Type 7 bits	Version=0 2 bytes	Reserved 1 byte	Size 3 bytes
Index 2 bytes	Bit Position 1 byte		Access Width In Bytes 1 byte	Data Register Address 2 bytes	Index Register Address 2 bytes

Table 4. Intel® TXT Configuration Policy Entry Version = 1, Flat Memory Type Pointer

CHKESUM 1 byte	C_V=0 1 bit	Type 7 bits	Version=1 2 bytes	Reserved 1 byte	Size 3 bytes
64 bit Physical Address 8 bytes Bit 0 - Intel® Configuration Policy 0 = Intel® TXT disabled 1 = Intel® TXT enabled					

4.10 Key Manifest Record (Type 0x0B) Rules

1. There can be more than one Key Manifest Record in the FIT. The multiple Key Manifest Record Entries must be in contiguous FIT Entries (i.e., all Key Manifest Entry Types must be together. This does not require that the Key Manifest themselves to be contiguous). This is to allow common BIOS images, which support multiple platforms. The ACM will compare each one with the Key Manifest ID in the Boot Policy Register and use the one, which matches.
2. The Version field should be set to 0x0100.
3. The C_V bit in this entry should be clear.
4. The Checksum field is set to 0.
5. The Size field must be set to the value not less than the size of the Key Manifest Structure.



4.11 Boot Policy Manifest (Type 0x0C) Rules

It is required that all elements of the Boot Policy Manifest be in the specific sequence and in contiguous memory.

1. There can be more than one Boot Policy Manifest entries in the FIT. The ACM will only use the first one found and will ignore subsequent entries of this type.
2. The entry must follow the Key Manifest Entry (Type 0x0B).
3. The Version field should be set to 0x0100.
4. The C_V bit in this entry should be clear.
5. The Checksum field is set to 0.
6. The Size field must be set to the value not less than the size of the Boot Policy Manifest Structure (the contiguous memory starting from the first byte of the Boot Policy Manifest Header through the last byte of the Boot Policy Manifest Signature Element).

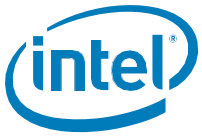
4.12 Intel® CSE Secure Boot (Type 0x10) Rules

1. There can be more than one Intel® CSE Secure Boot entries in the FIT, the order of these entries in the FIT table is not important.
2. The CSE created FIT table would have the OEM Key Manifest and OEM Boot Policy Manifest entries in it.
3. The Reserved field in the FIT table (refer to Table 1) will be used to further distinguish the type:

   0 = Reserved
   1 = Key Hash 1
   2 = CSE Measurement Hash
   3 = Boot Policy
   4 = Other Boot Policy
   5 = OEM SMIP
   6 = MRC Training Data
   7 = IBBL Hash
   8 = IBB Hash
   9 = OEM ID
   10 = OEM SKU ID
   11 = Boot Device Indicator
   (1= SPI, 2= eMMC, 3 = UFS, else are reserved)
   12 = FIT Patch Manifest (FPM)
   13 = AC Module Manifest (ACMM)
   14 onwards = Reserved

The OEM SMIP, MRC Training Data and IBB Hash, are not present in the initial SRAM map, but will be placed in the shared SRAM later (after Ring Buffer protocol is done) for IBBL to consume. This use of the Reserved field does not interfere in any way with the CPU microcode operation.

4. The Version field should be set to 0x0100.
5. The C_V bit in this entry should be clear.
6. The Checksum field is set to 0.



4.13 Feature Policy Record (Type 0x2D) Rules

1. There can be zero or more Feature Policy Records in the FIT. If there are more than one, the meaning of each successive one's bits is different from the others and will need to be defined.
2. This policy record is used for ACM feature control purposes and will be built into both debug signed ACMs and production signed ACMs. This Record is used for communicating configuration information to the startup ACMs, which is unlikely to be changed by BIOS setup settings. For this reason, it is only allowed to be an MMIO record.
3. The Feature Policy byte specifies changes in the normal operation of the ACMs according to the table below. (Refer to Table 5)
4. The offsets in the table are hard-coded and the value of the bit position field in the Feature Policy Record is ignored.
5. The default setting is as, if 0 is read for the byte. In other words, if this structure is not present or is invalid, the ACMs will behave as if Feature Policy byte = 0.
6. The Version field should be set to 0x0100.
7. The C_V bit in this entry should be clear.

Table 5. Feature Policy Bit Definition

Bit(s)	BIOS ACM	Function
32-2	N/A	Reserved. These bits are currently ignored by the ACM.
1	BIOSACM	PCR policy: 0 – means ACM does hashing. Example: Do not use TPM2_PcrEvent command 1 – means TPM does hashing. Example: Do use TPM2_PcrEvent command
0	BIOSACM	FEATURE_POLICY_ALLOW_SMB_WRT. If set, skips the SMB_DIS_WRT check in the BIOSACM LockConfig calls.