Firmware Interface Table

BIOS Specification

Revision 1.6

March 2025

Document Number: 599500

Contents

Figures

Tables

Revision History

Revision Number	Description	Revision Date
0.70	• Initial release	October 2016
001	• Updated Microcode Update Rules in section 4.3 • Initial publication	November 2018
1.1	• Merged 338505 content into new ID 599500 • Added FIT Entry for Diagnostic ACM Rules in section 4.4.2 • Updated base formatting and stylesheet	January 2020
1.2	• Updated CSE Secure Boot Rules in section	April 2020
1.3	• Internal only - maintenance updates	June 2021
1.4	• Added VAB Provisioning Table record • Added VAB Boot Key Manifest record • Added VAB Boot Image Manifest record • Added VAB Boot Image Descriptors record • Added StartUp ACM record • Added Platform Boot Policy record • Added FIT Reset State	October 2022
1.5	• Added Memory Microcontroller Firmware Image record • Added FSP Boot Manifest record	February 2024
1.6	• Updated section on FIT type 0xA to add the index TPM address option. • Added section for FIT type 0x1E., VAB Authorization Code Patch Image Manifest record. • Deprecated FIT type 0x1D	March 2025

§§

1 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.

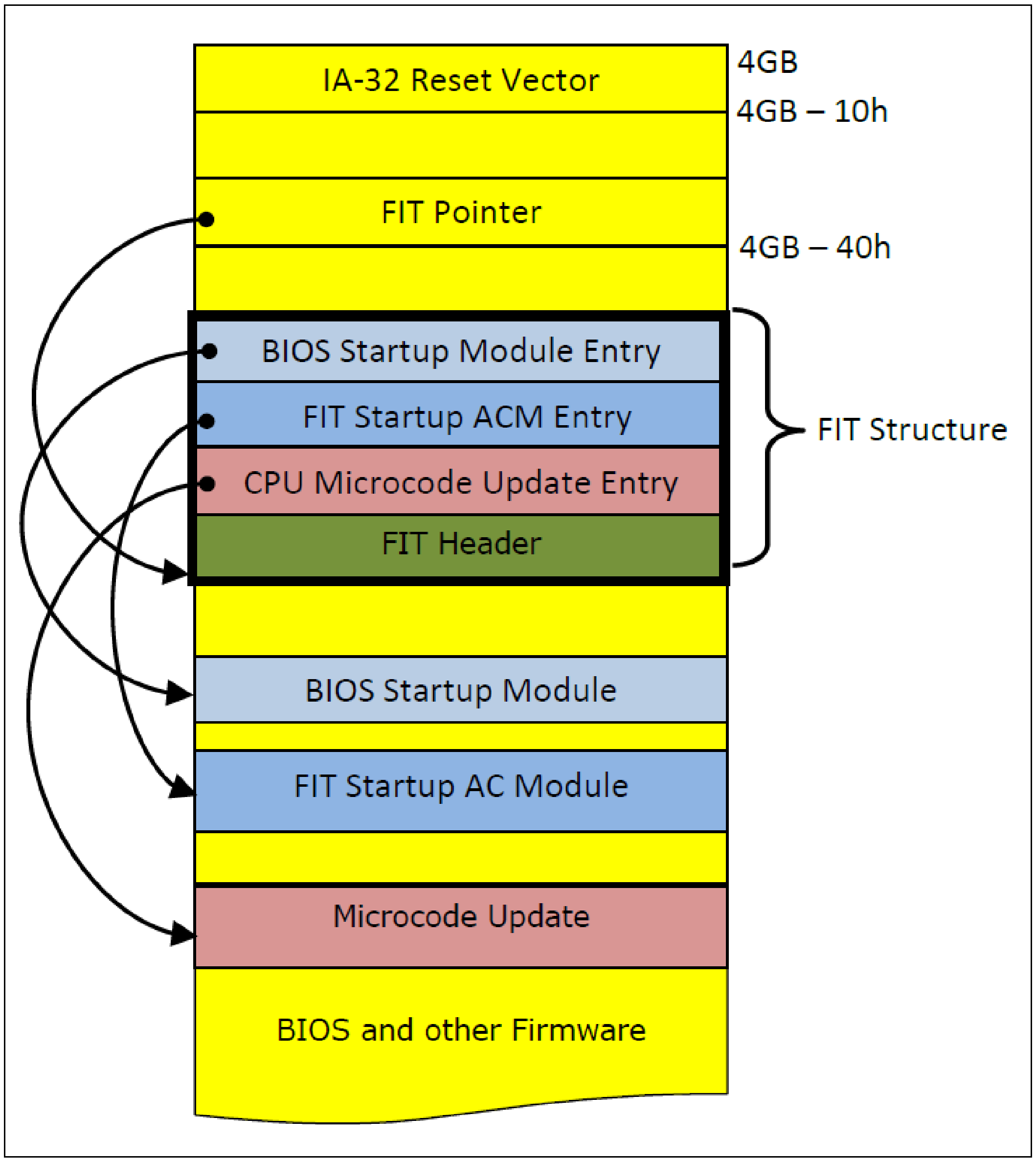
For detailed information regarding the ACM, refer to the Intel® Trusted Execution Technology BIOS Specification BIOS Writer's Guide.

§§

2 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 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 Intel® 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 must be noted, such that flash layout is suitable only for a platform not employing Intel® Boot Guard, which has own rules of Top Swap implementation.

§§

4 Firmware Interface Table

Each entry in the Firmware Interface Table is 16 bytes in length. Generic FIT entry format is described below.

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

Table 4-1: Generic FIT Entry Format

Byte- Offsets	15	14	13:12	11	10:8	7:0
Meaning	Chksu m	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 1-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.

Note: FIT Entries are not required to support the above generic format. Custom formats are permitted and are intensively used. Custom formats are described in dedicated sections.

Table 4-2: Generic FIT Entry Format

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

FIT Entry Type	Description (Intel® 64 and IA-32 Architectures)
0x04	Platform Boot Policy Entry
0x05	Memory Microcontroller Firmware Entry
0x06	FIT Reset State Entry
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	FSP Boot Manifest
0x0E - 0x0F	Intel Reserved
0x10	CSE Secure Boot
0x11 – 0x19	Intel Reserved
0x1A	Vendor Authorized Boot Provisioning Table
0x1B	Vendor Authorized Boot Key Manifest
0x1C	Vendor Authorized Boot Image Manifest
0x1D	Vendor Authorized Boot Image Hash (Deprecated)
0x1E	VAB Authorization Code Patch Image Manifest
0x1F - 0x2B	Intel Reserved
0x2C	SACM Debug Record
0x2D	Feature Policy Delivery Record. Deprecated in CBnT platforms.
0x2E	Granular SCRTM Error Record
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 processors 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 if 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

There are two supported versions of Type entries legacy 0x100 and updated (modern) 0x200

4.4.1 Legacy Record

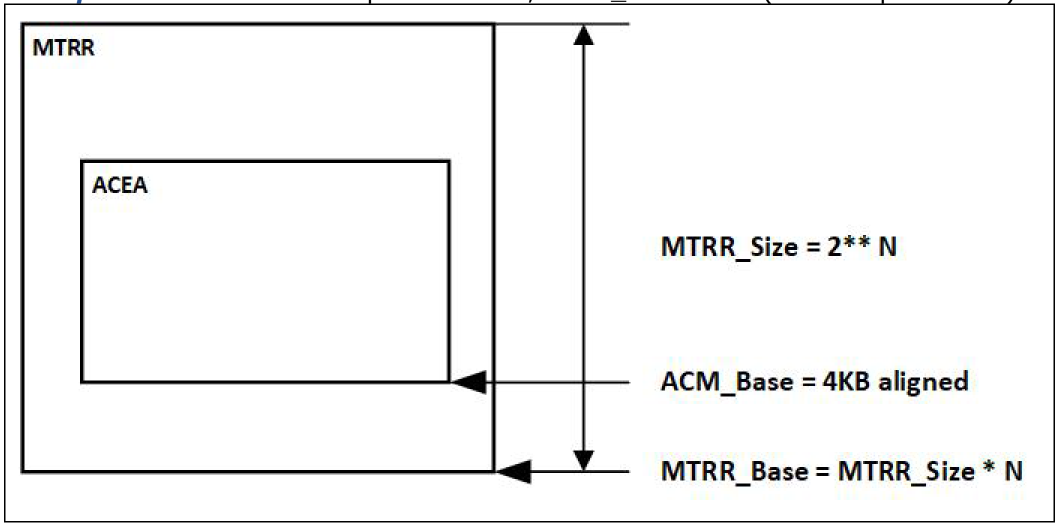
The following rules apply only to the record version 0x100:

1. One Startup ACM (Type 2) Entry in the FIT is required for FIT boot support.
2. The address field must point to a Startup ACM. Specifically, the address field in the Type 2 record must point 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.

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

Where ceiling (X) is a mathematical function returning the smallest integer Y larger than X

Example: If the size of Startup ACM is 13 k, MTRR_Size is 16k (the next power of 2).



NOTE: Legacy processor logic for the above MTRR programming was: attempt to program MTRR_Base equal to the ACM_Base. If attempt failed processor was failing ACM launch. This logic required IFWI to place ACM on the boundary meeting the above requirements for MTRR_Base. Recently for client processors requirement that MTRR_Base and ACM+Base must coincide has been relaxed. ACM must be on 4KB boundary, and it may "float" inside of MTRR covered window under condition that ACEA is completely covered by an MTRR. Server processor are still enforcing MTRR_Base == ACM_Base requirement.

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 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.

4.4.2 Modern Record

Legacy definition of Type 2 record does not allow loading alternative ACMs via FIT table. Modern Type 2 record overcomes this limitation.

FIT table may contain multiple modern Type 2 records pointing to different ACMs.

Table 3 below contains layout of the new record.

Table 4-3: Type 2 Record Version 0x200 Structure

Byte Offsets	**15**	**14**	**13:12**	**11**	**10**	**9**	**8**	**7:0**
Meaning	Ext Family Mask [7:4] Ext Family [3:0]	Bit 7 - C_V Bits 6:0 - Type	0x200	Ext Model Mask [7:4] Type Mask [3:0]	Family Mask [7:4] Model Mask [3:0]	Ext Model [7:4] Type [3:0]	Family [7:4] Model [3:0]	Address

The following rules apply only to the record version 0x200:

1. At least one Startup ACM (Type 2) Entry in the FIT is required for FIT boot support.
2. The address field must point to a Startup ACM. Specifically, the address field in the type 2 record must point 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.
5. ACM may be smaller than size of allocated Authenticated Code Execution Area (ACEA).
6. The C_V bit in this entry should be clear.
7. The Version field should be set to 0x200.
8. Checksum, Reserved, and size fields are redefined to deliver CPU FMS and CPU FMS Mask values as specified below.
9. Extended Family byte 15 is split onto two nibbles. High order nibble contains Extended Family Mask, while low order nibble contains Extended Family value.

10. Bytes 11 and 10 deliver in 4 consecutive nibbles Extended Model, Type, Family, and Model Mask value.
11. Bytes 9 & 8 deliver in 4 consecutive nibbles Extended Model, Type, Family and Model values.
12. FIT table may contain mixture of type 2 records version 0x100 and 0x200 but record version 0x100 must precede all records version 0x200.

4.4.3 CPU Processing Rules

CPU will scan iteratively Type 2 records:

1. Legacy CPU supporting only record version 0x100 will process it and will load ACM pointed by the address field.
2. Modern CPU supporting only records version 0x200 will skip record version 0x100 and will continue scanning.
3. CPU will read Extended Family, Extended Model, Type, Family and Model values from record version 0x200 and will construct full FMS_Target value supported by this record. It will assume Stepping value equal 0.
4. CPU will read Extended Family, Extended Model, Type, Family and Model Mask values from record version 0x200 and will construct full FMS_Mask value supported by this record. It will assume SteppingMask value equal 0
5. CPU will read own FMS value and apply the check:

```
IF (bitwise (FMS & FMS_Mask) equals FMS_Target)
        it will load ACM pointed to by the address field.

ELSE
        it will continue scanning.
```

Note: It might occur that one single ACM is designed to support two or more essentially different CPU microarchitectures such that creation of a Mask value is onerous or even impossible. In such a case BIOS shall create separate version 0x200 records each supporting one CPU microarchitecture but pointing to the same ACM.

4.5 Diagnostic ACM (Type 3) Rules

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

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:

1. Type 3 entry must point to an address that is accessible by the processor at reset vector and should be 4Kb aligned.
2. The C_V bit in this entry should be clear.
3. The Size field is not used. BIOS should set this field to 0.
4. The Version field should be set to 0x0100.

4.6 Platform Boot Policy (Type 4) Rules

Platform Boot Policy (Type 4) entry in the FIT is required for Servers executing CBnT and/or PFR without a PCH. Type 04 payload is owned by OEM. Details will be defined in upcoming Platform Boot Policy specification:

1. The address field points to the Platform Boot Policy Data.
2. Type 4 entry must point to an address that is accessible by the processor at reset vector and should be 16b 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.

4.7 Memory Microcontroller Firmware Image (Type 5) Rules

1. The Memory Microcontroller (MMC) firmware image (Type 5) entry in the FIT is applicable to specific platforms, such as some server Xeon platforms. Please check your specific platform for applicability.
2. The FIT table may contain more than one Type 5 entry. The IFWI may carry multiple MMC images for multiple processors and steppings. Each Type 5 entry points to a distinct MMC FW image. The address field in Type 5 entry points to the first byte of the MMC FW image.
3. Each Type 5 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. The IFWI may have some empty Type 5 slots. These slots are set aside to store future MMC FW images. It is suitable for a Type 5 entry to point to these empty slots if the first dword in the empty slot is 0xFFFF_FFFF.
5. MMC FW images pointed to by a type 5 entry must be aligned on a 16-byte address.
6. The IFWI stitching and construction process must not perform any additional compression, encoding, or encryption on the MMC FW image beyond what was already present in the delivered MMC FW image.
7. The C_V bit in this entry should be clear to 0.
8. The Size field is not used. This field should be cleared to 0.
9. The version field is not used. This field should be cleared to 0.

4.8 FIT Reset State (Type 6) Rules

FIT Reset (Type 6) entry is used on platforms which require certain CPU states to be initialized at reset vector:

1. At least one Type 6 Entry must be specified on such platforms. Otherwise, this entry is not required, and will be ignored if present.
2. The address field points to a FIT reset state table. Specifically, the address field in the type 6 record points to the first byte of the FIT reset state table.
3. Each Type 6 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. The FIT reset state table pointed to by a type 6 entry must be aligned on a 16-byte address.
5. The FIT reset state table pointed to by a type 6 entry must not be compressed, encoded, or encrypted by the BIOS.
6. The Size field is not used. BIOS should clear this field to 0.
7. The Version field should be set to 0x0100.
8. The C_V bit in this entry should be clear to 0.

4.9 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.

4.10 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. Type 8 entry address field must contain an address accessible by the processor at reset vector.
3. The address field contains the TPM_POLICY_PTR structure (see section 4.10.1. 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 indexed IO location must be accessible at reset without any hardware initialization.
8. The TPM policy says whether TPM should be enabled or disabled.

   If TPM Policy == 0 the TPM should be disabled.
   If TPM Policy == 1 the TPM 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 TPM Policy = 1.
10. The C_V bit in this entry should be clear to 0.
11. The Size field is not used. BIOS should set this field to 0.

Table 4-4: Type 8 Record Structure

Byte Offsets	**15**	**14**	**13:12**	**11**	**10:8**	**7:0**
Meaning	Checksum Must be 0	Bit 7 - C_V Must be 0 Bits 6:0 – Type Must be 8	Version Must be 0 or	Reserved Must be 0	Size Must be 0	Address TPM_POLICY_PTR. Must be INDEX_IO_ADDRESS If Version == 0 or FlatMemoryAddress if Version == 1

4.10.1 TPM Enabling Policy

```
typedef struct {

        UINT16        IndexRegisterAddress;

        UINT16        DataRegisterAddress;

        UINT8         AccessWidthInBytes;   // = 1 - 1-byte access;

                                            // = 2 - 2-byte access

        UINT8         BitPosition;          // e.g. = 15 - Bit15

        UINT16        Index;
} INDEX_IO_ADDRESS;

typedef union {

       UINT64        FlatMemoryAddress;

       INDEX_IO_ADDRESS      IndexIo;

} TPM_POLICY_PTR;
```

4.11 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

4.12 Intel® TXT Policy Data Record (Type 0xA) Rules

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

1. If TXT is supported by a platform, single instance of Type 0xA record must exist.
2. Type 0xA entry address field must contain an address 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 4.12.1)
4. The version field must be set to 0 if TXT_CONFIG_POLICY_PTR describes an Indexed IO type pointer.
5. The version field must be set to 1 if TXT_CONFIG_POLICY_PTR describes a flat memory pointer.

6. The version field must be set to 2 if TXT_CONFIG_POLICY_PTR describes a TPM NV index.
7. If indexed IO type pointer is used, the Address field holds a structure of the type INDEX_IO_ADDRESS – see section 4.10.1. 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.
8. The indexed IO location must be accessible at reset without any hardware initialization.
9. 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.
10. If TPM NV pointer is used, the Address field holds a structure of the type INDEX_TPM_ADDRESS. This structure contains the location within TPM NV index that holds the Intel® TXT Configuration Policy.
11. The Intel® TXT Configuration policy says whether Intel® TXT should be enabled or disabled.

    If TXT Configuration Policy == 0 the Intel® TXT should be disabled.
    If TXT Configuration Policy == 1 the Intel® TXT should be enabled.

12. 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 Configuration Policy == 1 but this may create a mismatch between BIOS and ACM TXT processing.
13. The C_V bit in this entry should be cleared to 0.
14. The Size field is not used. BIOS should set this field to 0.

Table 4-5: Type 0xA Address Field Content

FIT_A Version	FIT_A Address Field Content
0	INDEX_IO_ADDRESS
1	FLAT_MEMORY_POINTER
2	INDEX_TPM_ADDRESS

Table 4-6: Type 0xA Record Structure

Byte Offsets	15	14	13:12	11	10:8	7:0
Meaning	Checksum Must be 0	Bit 7 - C_V Must be 0 Bits 6:0 – Type Must be 0xA	Version Must be 0, 1, or 2	Reserved Must be 0	Size Must be 0	Address TXT_CONFIG_POLICY_PTR. Version 0 - INDEX_IO_ADDRESS 1 – FLAT_MEMORY_POINTER 2 - INDEX_TPM_ADDRESS

4.12.1 TXT Configuration Policy

```
INDEX_IO_ADDRESS
{
    UINT16    Index Register Address  // 0x70
    UINT16    Data Register Address   // 0x71
    UINT8     Field Width In Bytes    // = 1 - 1-byte width
                                      // = 2 - 2-byte width
    UINT8     Starting Bit Position   // e.g. = 15 - BIT15
    UINT16    CMOS Index              // CMOS Index offset
                                      // e.g. = 50 – byte 50
}

FLAT_MEMORY_POINTER
{
    UINT64*   VarPtr                  // *VarPtr[0] = 1 – enabled
                                      //            = 0 - disabled
}

INDEX_TPM_ADDRESS
{
    UINT32    TPM NV index handle
    UINT8     Field Width In Bytes    // = 1 - 1-byte width
                                      // = 2 - 2-byte width
    UINT8     Starting Bit Position   // e.g. = 7 - BIT7
    UINT16    Byte Offset             // Offset within NV index
                                      // e.g. = 1 - byte 1
}
```

```
typedef union {

        FLAT_MEMORY_POINTER                FlatMemoryAddress;

        INDEX_IO_ADDRESS                           IndexIo;
        // see section 4.10.1

        INDEX_TPM_ADDRESS
        IndexTpm;

} TXT_CONFIG_POLICY_PTR;
```

4.13 Key Manifest Record (Type 0xB) 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.

4.14 Boot Policy Manifest (Type 0xC) 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).

4.15 FSP Boot Manifest (Type 0xD) Rules

It is required that all elements of the "FSP Boot Manifest" (FBM) be in the specific sequence and in contiguous memory.

FBM record 0xD must follow "Generic FIT Entry Format" as shown in Table 1:

1. There can be more than one FBM record entry in the FIT. The ACM will only use the first one found and will ignore subsequent entries of this type.
2. The entry must be located after the Boot Policy Manifest Entry (Type 0x0C).
3. The Version field should be set to 0x0100.
4. The C_V bit in this entry should be clear.
5. The Checksum field should be set to 0.
6. The Size field should be set to the value not less than the size of the FBM Structure (the contiguous memory starting from the first byte of the FBM Structure through the last byte of the FBM Signature Element).

4.16 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 Table1) 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:

1. The Version field should be set to 0x0100.
2. The C_V bit in this entry should be clear.
3. The Checksum field is set to 0.

4.17 Vendor Authorized Boot Provisioning Table (Type 0x1A) Rules

VAB Provisioning Table Manifest (Type 0x1A) entry in the FIT is required only for platforms which support Vendor Authorized Boot. The VAB Provisioning Manifest is used to initially program the Key Authentication Keys (KAKs). There can only be zero or one Type 0x1A record present in the FIT:

1. The Version field should be set to 0x0100.
2. The C_V bit in this entry should be clear.
3. The Checksum field is set to 0.
4. The entry must be 64 byte aligned.
5. Address + size must be under (4GB – 1) and above (4GB – 16MB).

4.18 Vendor Authorized Boot Key Manifest (Type 0x1B) Rules

VAB Key Manifest (Type 0x1B) entry in the FIT is required only for platforms which support Vendor Authorized Boot. It is signed by the Key Authorization Key (KAK). The VAB Key Manifest contains keys that sign VAB Image Manifest (Type 0x1C). There can only be zero or one Type 0x1B record present in the FIT:

1. The Version field should be set to 0x0100.
2. The C_V bit in this entry should be clear
3. The Checksum field is set to 0.
4. The entry must be 64 byte aligned.
5. Address + size must be under (4GB – 1) and above (4GB – 16MB)

4.19 Vendor Authorized Boot Image Manifest (Type 0x1C) Rules

VAB Image Manifest (Type 0x1C) entry in the FIT is required only for platforms which support Vendor Authorized Boot. It contains an Image Manifest Header and an Image Manifest. The VAB Image Manifest (Type 0x1C) is signed by a key in the VAB Key Manifest (Type 0x1B). The Image Manifest Header defines the number of Image Manifests. The Image Manifest contains hashes of firmware to be authorized. There can only be zero or one Type 0x1C record present in the FIT:

1. The Version field should be set to 0x0100.
2. The C_V bit in this entry should be clear
3. The Checksum field is set to 0.
4. The entry must be 64 byte aligned.
5. Address + size must be under (4GB – 1) and above (4GB – 16MB).

4.20 Vendor Authorized Boot Image Descriptors (Type 0x1D) Rules (Deprecated)

Note: This FIT Type is deprecated starting in version 1.6. Any implementation of this field conforming to version 1.5 or prior may not be supported.

Boot Image Descriptors (Type 0x1D) entry in the FIT is required only for platforms which support Vendor Authorized Boot. The Boot Image Descriptors Manifest is an optional header only required on SPI flash for the BIOS as a structure to define number of Image Descriptors that follow it. There can only be zero or one Type 0x1D record present in the FIT:

1. The Version field should be set to 0x0100.
2. The C_V bit in this entry should be clear
3. The Checksum field is set to 0.
4. The entry must be 64 byte aligned.
5. Address + size must be under (4GB – 1) and above (4GB – 16MB).

4.21 VAB Authorization Code Patch Image Manifest (Type 0x1E) Rules

Type 0x1E entry in the FIT is required for BIOS to locate the VAB Authorization Code Patch Image Manifest and request security controller to authorize and update. It contains a Manifest Header and one or more Authorization Patch Image Manifests. The VAB Authorization Code Patch Image Manifest (0x1E) is signed by a key in the VAB Key Manifest (Type 0x1B). The Manifest Header defines the number of Authorization Code Patch Image Manifests. The Authorization Code Patch Image Manifest contains authorization code to update Quantum Safe Cryptography algorithms used for authorization during the products life cycle:

1. There can only be zero or one Type 0x1E record present in the FIT.
2. The Version field should be set to 0x0100.
3. The C_V bit in this entry should be clear to 0.
4. The Checksum field is set to 0.
5. The entry must be 64 byte aligned.
6. Address + size must be under (4GB – 1) and above (4GB – 16MB).

4.22 SACM Debug Record (Type 0x2C) Rules

Engineering Record used by certain Debug ACMs. Ignored by Production and Release ACMs:

1. There can be zero or one Type 02C record present in the FIT. If there are more than one, subsequent entries will be ignored.
2. Release / production signed ACMs will ignore this record.

4.23 Feature Policy Record (Type 0x2D) Rules

Feature Policy Record is deprecated in CBnT ACMs and is ignored if present.

For legacy platforms, usage is documented in Intel (R) Trusted Execution Technology and BootGuard Server BIOS specification, available here: https://cdrdv2.intel.com/v1/dl/getContent/558294

4.24 Granular SCRTM Error Record (Type 0x2E) Rules

Granular SCRTM Error Record allows OEMs enabling CBnT to do “best effort” to measure the KM, BPM, IBB and policy data into PCR0 in the case of an integrity error. This record is only valid for BTGP0 / TXT enabled, as integrity errors will result in BTG enforcement. This record is optional; default behavior for integrity errors results in CBnT ACMs not starting the TPM.

ACM will verify the contents of this record only when an integrity error is found. If this record cannot be verified, ACM will fall back to default PCR0 handling (not starting the TPM at all):

1. There can be zero or one Granular SCRTM Error Records in the FIT. Additional type 0x2E entries will be ignored by the ACM.
2. The address field points to the Backup IBB address. This address must be 16b aligned, and within accessible range for ACM. Current allowed ranges are (4GB-16MB) to (4GB – 1).
3. The size field indicates the size of the Backup IBB address. This size field must be within reasonable values. Current allowed ranges as (4KB-16MB).
4. Address + size must be under (4GB –1) and above (4GB – 16MB).
5. The Version field should be set to 0x0100.
6. The C_V bit in this entry should be clear.

§§