Management Component Transport Protocol (MCTP) Host Interface Specification

Version: 2.0.0

Document Identifier: DSP0256

Date: 2025-03-04

Version History: https://www.dmtf.org/dsp/DSP0256

Supersedes: 1.0.0

Document Class: Normative

Document Status: Published

Document Language: en-US

10 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

11 Implementation of certain elements of this standard or proposed standard may be subject to third-party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights and is not responsible to recognize, disclose, or identify any or all such third-party patent right owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners, or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third-party patent rights, or for such party's reliance on the standard or incorporation thereof in its products, protocols, or testing procedures. DMTF shall have no liability to any party implementing such standards, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

12 For information about patents held by third-parties which have notified DMTF that, in their opinion, such patents may relate to or impact implementations of DMTF standards, visit https://www.dmtf.org/about/policies/disclosures.

13 PCI-SIG, PCIe, and the PCI HOT PLUG design mark are registered trademarks or service marks of PCI-SIG. All other marks and brands are the property of their respective owners.

14 This document's normative language is English. Translation into other languages is permitted.

CONTENTS

1 Foreword

The *Management Component Transport Protocol (MCTP) Host Interface Specification* (DSP0256) was prepared by the PMCI Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about DMTF, see https://www.dmtf.org.

1.1 Acknowledgments

DMTF acknowledges the following individuals for their contributions to this document:

Editors:

- Philip Chidester — Dell Inc.
- Samer El-Haj-Mahmoud — Arm Limited
- Jose Marinho — Arm Limited
- Hemal Shah — Broadcom Inc.

Contributors:

- Harb Abdulhamid — Ampere Computing
- Patrick Caporale — Lenovo
- Robert Gough — Ampere Computing, Intel Corporation
- Darren Hart — Ampere Computing
- Yuval Itkin — NVIDIA Corporation
- Janusz Jurski — Intel Corporation
- Eliel Louzoun — Intel Corporation
- Thanu Rangarajan — Arm Limited
- David Rudy — Oracle

35 2 Introduction

36 The Management Component Transport Protocol (MCTP) defines a communication model intended to facilitate communication between:

- 37 Management controllers and other management controllers
- 38 Management controllers and management devices

39 The communication model includes a message format, transport description, message exchange patterns, and configuration and initialization messages. The *MCTP Base Specification* (DSP0236) describes the protocol and commands used for communication within and initialization of an MCTP network

40 This *MCTP Host Interface Specification* (DSP0256) describes how MCTP Host Interfaces are discovered (interfaces between a management controller and host software).

41 2.1 Document conventions

42 The conventions described in the following clauses apply to this specification.

43 2.2 Reserved and unassigned values

44 Unless otherwise specified, any reserved, unspecified, or unassigned values in enumerations or other numeric ranges are reserved for future definition by DMTF.

45 Unless otherwise specified, numeric or bit fields that are designated as reserved shall be written as 0 (zero) and ignored when read.

46 2.3 Byte ordering

47 Unless otherwise specified, byte ordering of multi-byte numeric fields or bit fields is "Big Endian" (that is, the lower byte offset holds the most significant byte, and higher offsets hold lesser significant bytes).

3 Scope

This document provides the specifications for the Management Component Transport Protocol (MCTP) Host Interface.

50 4 Normative references

51 The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

52 DMTF DSP0134, *SMBIOS Specification* (System Management BIOS reference specification) https://www.dmtf.org/dsp/DSP0134

53 DMTF DSP0233, *Management Component Transport Protocol (MCTP) I3C Transport Binding Specification* https://www.dmtf.org/standards/published_documents/DSP0233_1.0.pdf

54 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification* https://www.dmtf.org/standards/published_documents/DSP0236_1.3.pdf

55 DMTF DSP0237, *Management Component Transport Protocol (MCTP) SMBus/I2C Transport Binding Specification* https://www.dmtf.org/standards/published_documents/DSP0237_1.2.pdf

56 DMTF DSP0238, *Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification* https://www.dmtf.org/standards/published_documents/DSP0238_1.2.pdf

57 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes* https://www.dmtf.org/standards/published_documents/DSP0239_1.9.pdf

58 DMTF DSP0282, *Memory-Mapped Buffer Interface (MMBI) Specification* https://www.dmtf.org/standards/published_documents/DSP0282_1.0.pdf

59 DMTF DSP0283, *Management Component Transport Protocol (MCTP) Universal Serial Bus (USB) Transport Binding Specification* https://www.dmtf.org/standards/published_documents/DSP0283_1.0.pdf

60 DMTF DSP0284, *Management Component Transport Protocol (MCTP) Memory-Mapped Buffer Interface (MMBI) Transport Binding Specification* https://www.dmtf.org/standards/published_documents/DSP0284_1.0.pdf

61 DMTF DSP0292, *Management Component Transport Protocol (MCTP) Platform Communications Channel (PCC) Transport Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0292_1.0.pdf

62 IPMI Consortium, *Intelligent Platform Management Interface Specification: Second Generation* v2.0 Revision 1.1, October 1, 2013 https://web.archive.org/web/20191124115026/https://www.intel.com/content/www/us/en/products/docs/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html

63 MIPI Alliance, *MIPI I3C Specification* https://www.mipi.org/mipi-i3c-basic-download

64 MIPI Alliance, *MIPI I3C Device Characteristics Register* https://www.mipi.org/mipi_i3c_device_characteristics_register

65 PCI-SIG, *PCI Code and ID Assignment Specification* Revision 1.11 (24 Jan. 2019) https://pcisig.com/sites/default/files/files/PCI_Code-ID_r_1_11__v24_Jan_2019.pdf

66 UEFI Forum, *Advanced Configuration and Power Interface (ACPI) Specification* https://uefi.org/specifications

67 5 Terms and definitions

68 Refer to DSP0236 for terms and definitions that are used across the MCTP specifications. For the purposes of this document, the following terms and definitions apply.

69 This document defines these additional terms:

70 **MCTP Host Interface**
a Host Interface that enables host software to locally access an MCTP Network in the managed system.

71 6 Symbols and abbreviated terms

72 Refer to DSP0236 for symbols and abbreviated terms that are used across the MCTP specifications. For the purposes of this document, the following additional symbols and abbreviated terms apply.

73 The following symbols and abbreviations are used in this document.

74 **ASL**
ACPI Source Language

75 **DSDT**
Differentiated System Description Table

76 **MCTP HI**
MCTP Host Interface

7 MCTP Host Interface

The MCTP Host Interface Specification describes how MCTP Host Interfaces are discovered. This specification describes the Host Interface discovery and commands for registering software endpoints such as BIOS, UEFI, or system software.

The MCTP HI discovery is defined for the following interface types:

- I2C/SMBus
- I3C
- MMBI
- PCC
- Serial
- USB

7.1 Host Interface discovery

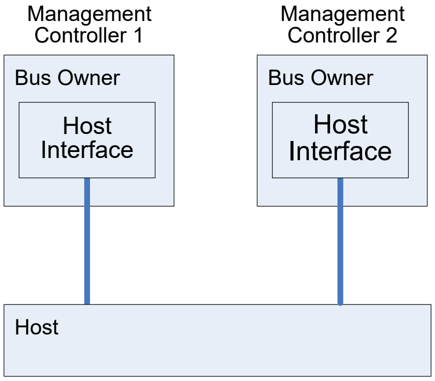

Figure 1 — MCTP Host Interface Discovery

A Management Controller implements a single Host Interface. There can be multiple Management Controllers in a system, but each Management Controller is in a separate MCTP network that connects to the Host.

Each MCTP Host Interface can be discovered using one or more discovery methods described in this specification.

7.2 Multiple Host Interfaces

Multiple Host Interfaces may exist in a given system implementation. Refer to the "MCTP Overview" clause in DSP0236 for details on identifying whether multiple Host Interfaces connect to common or separate MCTP networks.

A Host is permitted to interact with different Host Interfaces in order to identify the Host Interface that connects to the management controller that the Host intends to interact with. That Host can use the "Get Endpoint UUID" MCTP command to identify the management controller on each Host Interface.

94 7.3 Transport-specific commands

95 For correct Host Interface operation, the MCTP Host Interface may require transport-specific commands to be implemented.

96 An example of a required transport-specific command is the SMBus Get UUID command. The SMBus Get UUID command is used to verify that the SMBus endpoint implements MCTP support. For more information see DSP0237.

97 Other transports may require transport-specific commands. The details of these commands are described in the transport-binding specifications.

98 8 Locating MCTP Host Interfaces with ACPI

99 This specification introduces the option of describing the presence of the MCTP Host Interface using ACPI namespace.

100 8.1 Locating MCTP Host Interfaces in the ACPI namespace

101 MCTP Host Interfaces may be described in the ACPI namespace. This makes it possible for the OSPM (an ACPI term for "OS Operating System-directed configuration and Power Management" essentially meaning an ACPI-aware OS or OS loader) to enumerate the MCTP Host Interface as a device.

102 Note that to be ACPI compatible, the fixed resources for MCTP Host Interfaces shall still be accounted for in accordance with the ACPI Specification. If the device is not formally described in the ACPI namespace, its resources shall be described as fixed system resources or the resources appended to some other fixed resource system device in order to ensure that the OS does not attempt to allocate those resources to some other device.

103 To formally describe the MCTP Host Interface in the ACPI namespace, an MCTP device is created using the named device object. The MCTP device object can have the elements as described in Table 1.

104 **Table 1 — MCTP Device Object Control Methods**

Object	Description	Support Level
_ADR	Named object that evaluates to the interface's address on its parent bus. _ADR is a standard device configuration control method defined in the ACPI Specification. A device object shall contain either an _HID object or an _ADR object, but shall not contain both.	Required only for devices on a bus that has standard enumeration mechanism.
_HID	Named object that provides the interface's Plug and Play identifier. This value can be vendor specific but shall be set to DMT0001[1] if no CID object is provided. _HID is a standard device configuration control method defined in the ACPI Specification. According to the ACPI Specification, an ACPI device object may contain either an _HID object or an _ADR object, but shall not contain both.	Required only for devices that do not use standard bus enumeration mechanism.
_CID	Named object that provides the interface's compatible Plug and Play identifier. This object is required and contains the value of DMT0001 if _HID is present and contains vendor specific identifier. Otherwise, this object is optional.	See the "Description" column for this object.

Object	Description	Support Level
_STR	Named object that evaluates to a Unicode string that may be used by an OS to provide information to an end user describing the device. _STR is a standard device configuration control method defined in the ACPI Specification. The strings "MCTP_Serial", "MCTP_I2C", "MCTP_MMBI_eSPI", and "MCTP_PCC" are recommended for identifying the Serial, I^{2}C/SMBus, MMBI over eSPI, and PCC Host Interfaces, respectively. Note: the strings defined above refer to interface types that can be described via an ACPI DSDT device. Other interface types, such as I3C or USB, are discovered via bus-specific schemes and so will never be made discoverable via an ACPI DSDT device. For such interface types, a string definition is not required.	Required
_UID	Named object that specifies a device's unique persistent ID, or a control method that generates it. _UID is a standard device configuration control method defined in the ACPI Specification. The value 0 shall be used to identify the primary MCTP Host Interface.	Required if more than one device is present.
_CRS	Named object that returns the interface's current resource settings. System Processor Management Interfaces are considered static resources; hence only return their defined resources. The address region definition is interface type/subtype dependent. _CRS is a standard device configuration control method defined in the ACPI Specification.	Recommended
_STA	Object that returns the status of the device: enabled, disabled or removed, as defined in the ACPI Specification. If this method is not present, the device is assumed to be enabled.	Recommended
_IFT	Object that specifies the MCTP Host interface type, as defined in DSP0239. The OS driver uses the _IFT object to determine the link-layer type. Note: _IFT has been reserved in ACPI Specification as the control method name defined by the SPMI table in the IPMI Specification and by this specification.	Required
_SRV	Object that specifies the MCTP Base specification revision. (Note: _SRV has been reserved in ACPI Specification as the control method name defined by the SPMI table in the IPMI Specification and by this specification.)	Recommended

105 1. DMTF has registered the DMT PNP ID with https://uefi.org/PNP_ACPI_Registry for describing all DMTF-related devices. DMTF has granted the use of DMT0001 to describe the generic Management Controller Device as defined in this specification.

106 In a multi-node system where there may be more than one MCTP device in an OS domain, it is highly recommended that all MCTP devices be described in the ACPI namespace with the _STA returning enabled for the active MCTP device(s).

107 Multiple MCTP Host Interfaces can exist. There shall be a DSDT ACPI table for each interface. Each such table has a unique _UID value. The _UID value for this device shall be an integer. One of the tables shall have _UID=0. The model defined in the ACPI Specification allows for a sparse _UID instance number set, e.g. an Instance number set of {0, 2, 10, ...} representing a collection of devices with _UID=0, _UID=2 and _UID=10, ... is permitted by the ACPI model.

108 If present, the MCTP HI ACPI DSDT description shall conform to the following interface-type-specific requirements:

109 - I^{2}C/SMBus DSDT shall:
110   - return 0x09 from the _IFT method,

    - contain an I2CSerialBusV2 resource, representing the SMBus Host Controller. In the _CRS, the I2CSerialBusV2 shall:
        - contain an address that is 7 bits long, placed in bits [6:0] of the I2C/SMBus bus target address, and likewise set the addressing mode to AddressingMode7Bit (because SMBus addresses are 7 bits long),
        - set resource usage to ResourceConsumer.
- MMBI on eSPI DSDT shall:
    - return 0xC from the _IFT method,
    - contain a pointer to the base address of the MMBI Capability Descriptor.
- PCC DSDT shall:
    - contain the PCC space identifiers, of both the type 3 and type 4 PCC channel, in a _CRS object,
    - return 0x0D from the _IFT method.

8.2 Example MCTP definition ASL code

Example ASL code that describes MCTP Host Interfaces is provided in the following subclauses.

122 8.2.1 Example 1: I^{2}C/SMBus interface

123 Example ASL for describing the presence of the MCTP Host Interface using the I^{2}C/SMBus MCTP binding is shown in the following listing:

```
Device(SMB0) // example SMBus host controller
{
  Name(_HID, "<Vendor-Specific HID>") // Vendor-specific HID
  Name(_UID, 0) // Unique ID of particular host controller
  :
  :
} // end Device SMB0

Device (MHI0)
{
  Name(_HID, "DMT0001") // MCTP Host Interface
  Name(_STR, Unicode("MCTP_I2C")) // String to identify MCTP Host Interface
  Name(_UID, 0) // UID for the primary MCTP Host Interface in the system

  // Current Resource Settings of the SMBus interface
  Name(_CRS, ResourceTemplate() {
    I2CSerialBusV2(
      0x0010,                // I2C/SMBus bus target address
      ControllerInitiated,   // Target mode
      400000,                // Maximum connection speed
      AddressingMode7Bit,    // Addressing mode
      "\\_SB.SMB0",          // Resource source (I2C/SMBus bus controller)
      0x00,                  // Resource Source index
      ResourceConsumer,      // Resource usage
      MCTP_HI0,              // Descriptor name
      Exclusive,             // Shared or Exclusive
                             // Vendor specific data
    )
  }

  Method (_STA, 0, NotSerialized) {
    Return (0x0F)
  }

  // Returns the interface type
  Method(_IFT) {
    Return (0x09) // MCTP I2C/SMBus (DSP0239, Table 4)
  }

  // Returns the interface specification version
  Method(_SRV) {
    Return(0x0130) // MCTP Base Specification Revision 1.3
  }
} // end Device MHI0
```

124 8.2.2 Example 2: MMBI interface on eSPI bus

125 Example ASL for describing the presence of the MCTP Host Interface using the MMBI MCTP binding on an eSPI bus is shown in the following listing:

```
Device(MHI0)
{
  Name(_HID, "DMT0001") // MCTP Host Interface
  Name(_STR, Unicode("MCTP_MMBI_eSPI")) // String to identify MCTP Host Interface
  Name(_UID, 0)   // UID for the primary MCTP Host Interface in the system

  // Returns the "Current Resources"
  Name(_CRS,
  ResourceTemplate() {
    QWordMemory(
    ResourceConsumer, //
    PosDecode, //
    MinFixed, //
    MaxFixed, //
    NonCacheable, //
    ReadWrite, //
      0x0000000000000000,   // \_GRA, Address granularity.
                            // E.g. All 64-bits decoded.
      0x80000FFFFC020000,   // \_MIN, Address of the MMBI Capability Descriptor.
      0x80000FFFFC02003F,   // \_MAX, Highest address in the MMBI Capability Descriptor.
      0x0000000000000000,   // \_TRA, Translation.
      0x0000000000000040,   // \_LEN, 64 byte MMBI Capability Descriptor.
    )
  }

  // Returns the interface type
  Method _IFT {
    Return (0x0C) // MCTP MMBI (DSP0239, Table 4)
  }

  // Returns the interface specification version
  Method(_SRV) {
    Return(0x0130) // MCTP Base Specification Revision 1.3
  }
}
```

126 8.2.3 Example 3: PCC interface

127 Example ASL for describing the presence of the MCTP Host Interface using the PCC MCTP binding is shown in the following listing:

```
Device(PCC0)
{
  Name(_HID, "DMT0001") // MCTP Host Interface
  Name(_STR, Unicode("MCTP_PCC")) // String to identify MCTP Host Interface
  Name(_UID, 0)   // UID for the primary MCTP Host Interface in the system

  // Returns the "Current Resources"
  Name(_CRS,
  ResourceTemplate() {
    WordPCC (
      1, // PccChannel, the index of the type 3 PCC subspace in the PCCT
      , // ResourceSourceIndex
      , // ResourceSource
      , // DescriptorName
    ),
    WordPCC (
      2, // PccChannel, the index of the type 4 PCC subspace in the PCCT
      , // ResourceSourceIndex
      , // ResourceSource
      , // DescriptorName
    )
  }

  // Returns the interface type
  Method _IFT {
    Return (0x0D) // MCTP PCC (DSP0239, Table 4)
  }

  // Returns the interface specification version
  Method(_SRV) {
    Return(0x0130) // MCTP Base Specification Revision 1.3
  }
}
```

9 Locating MCTP Host Interfaces via SMBIOS tables

The SMBIOS DSP0134 specification defines the Management Controller Host Interface table (with type set to 42). The SMBIOS Type 42 table describes the interface and protocol types. For each Protocol type, there is a Protocol Type specific data entry.

9.1 Protocol Type specific data

For an MCTP HI, the SMBIOS Type 42 has the Interface Type set to a value from the "MCTP Host Interfaces" range. The values in the "MCTP Host Interfaces" range are defined in the "MCTP Host Interface type identifiers" clause of DSP0239. There shall be at least one Protocol Record entry. That mandatory Protocol Record shall have Protocol Type set to MCTP (03h). The Protocol Type specific data for the Protocol Record with 03h Protocol Type is defined in Table 2.

Table 2 — Protocol Type Specific Data

Offset	Field Name	Size	Description
00h	MCTP protocol version	WORD	The version of the MCTP base specification DSP0236 implemented by the management controller. Bits[15:8] contain the major version of the MCTP base specification, Bits[7:0] contain the minor version of the MCTP base specification. For example, for a MCTP HI that implements version 1.3.1 of the MCTP base specification, this field is set to 0x0103.
02h	MCTP link-layer type	BYTE	Integer describing the type of link or protocol layer that connects the Host to the management controller. The value is as defined in the MCTP Host Interface type identifier table in DSP0239.
03h	Reserved	BYTE	Reserved.
04h	Instance number	DWORD	The instance number of the MCTP Host Interface. If this MCTP HI has an ACPI DSDT device (Characteristics[0]==1), then the instance number matches with the _UID method of that ACPI DSDT device.
08h	Characteristics	DWORD	Characteristics is a bit field describing properties of this interface. Characteristics[0]: ACPI DSDT description: • 0: This MCTP HI does not have an ACPI DSDT device. • 1: This MCTP HI has an ACPI DSDT device. All other bits are reserved.

9.2 Interface Type specific data

The Interface Type specific data is defined for the MMBI interface type. This data is reserved for the remaining interface types, occupying the minimum required size of 4 bytes, as specified by the field definition defined by Table 3, which is a DSP0134 Type 42 structure.

137 **Table 3 — Interface Type Specific Data**

Interface Type(s)	Interface Type specific data definition			
	Offset	**Field name**	**Size**	**Description**
MMBI	00h	MMBI Capability Descriptor pointer	QWORD	A pointer to the MMBI capability descriptor, as defined in DSP0282. This field is used if MMBI is on an eSPI bus. If the MMBI device is not on an eSPI bus, this field shall be set to 0 (zero).
I2C/SMBus, I3C, PCC, USB	00h	Reserved.	DWORD	Reserved.

138 10 MCTP HI bus-specific discovery

139 Some buses have sufficient abstractions defined to enable bus-specific MCTP HI endpoint discovery. This clause documents such discovery processes for the given bus examples.

140 10.1 MCTP HI discovery on I3C

141 An MCTP HI endpoint on an I3C bus complies with the MIPI I3C Specification and implements MCTP support as defined in I3C MCTP binding. The association between the MCTP Host Interface and a particular I3C device is done by matching the I3C endpoint Device Characteristics Register (DCR). I3C devices implementing an MCTP HI shall have the Device Characteristics Register set to 0xCC, as defined in the MIPI I3C Device Characteristics Register.

142 10.2 MCTP HI discovery on USB

143 An MCTP HI endpoint on a USB bus is discovered as specified in the "MCTP support and capabilities discovery" clause of the MCTP over USB binding specification (DSP0283).

144 10.3 MCTP HI discovery on MMBI over PCIe

145 A PCIe function that exposes an MMBI interface implementing an MCTP HI must have its PCIe Base Class, Sub-class, and Programming interface set to {0xC, 0xC, 0x0}, respectively, as defined in the PCI Code and ID Assignment Specification. The MMBI Capability Descriptor is located within a BAR of the function as defined by DSP0282. Additional information on MMBI discovery is found in DSP0282.

11 ANNEX A (informative) Notation

Notations used in this document include the following:

2:N	In field descriptions, this will typically be used to represent a range of byte offsets starting from byte two and continuing to and including byte N. The lowest offset is on the left, and the highest is on the right.
(6)	Parentheses around a single number can be used in message field descriptions to indicate a byte field that may be present or absent.
(3:6)	Parentheses around a field consisting of a range of bytes indicate the entire range may be present or absent. The lowest offset is on the left, and the highest is on the right.
PCIe	Underlined blue text is typically used to indicate a reference to a document or specification listed in the "Normative references" clause or to items hyperlinked within the document.
rsvd	Abbreviation for "reserved." Case insensitive.
[4]	Square brackets around a number are typically used to indicate a bit offset. Bit offsets are given as zero-based values (that is, the least significant bit (LSb) offset = 0).
[7:5]	A range of bit offsets. The most significant bit is on the left, and the least significant bit is on the right.
1b	A lowercase "b" following a number consisting of 0s and 1s is used to indicate the number is being given in binary format.
0x12	A leading "0x" is used to indicate a number given in hexadecimal format.

12 ANNEX B (informative) Change log

Version	Date	Description
1.0.0	2010-07-21	Released as DMTF Standard
2.0.0	2025-03-04	Fixed document title Added acknowledgments list Updated references versions and links Major update to MCTP host interface discovery for current MCTP bindings - differs significantly from the 1.0.0 spec and is not backward compatible Removed the SMBIOS, PCI/PCIe, and static ACPI MCHI table methods for MCTP Host Interface discovery Updated the ACPI namespace discovery method to cover newer MCTP bindings (including I^{2}C/SMBus and I3C) Updated ACPI ASL examples to add I^{2}C/SMBus and I3C