



Field Replaceable Unit (FRU) Data Specification

Version: 1.0.0

Document Identifier: DSP0220

Date: 2025-05-19

Version History: <https://www.dmtf.org/dsp/DSP0220>

Supersedes: None

Document Class: Normative

Document Status: Published

Document Language: en-US

Copyright Notice

Copyright © 2025 DMTF. All rights reserved.

- 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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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	4
1.1 Acknowledgments	4
2 Introduction	5
2.1 Document conventions	5
2.1.1 Typographical conventions	5
2.1.2 Reserved and unassigned values	5
2.1.3 Byte ordering	5
2.1.4 Data types	5
3 Scope	6
4 Normative references	7
5 Terms and definitions	8
6 Symbols and abbreviated terms	9
7 FRU Content Storage	10
8 FRU Content Structure	11
8.1 FRU Files Record	11
8.1.1 Header	11
8.1.2 File Descriptor Table	11
8.1.3 FRU File(s)	12
8.1.4 File Descriptor for FRU Files	13
8.1.5 FRU File Format and Encoding	14
8.1.5.1 Compression header	15
8.2 DMTF FRU Files	15
8.2.1 DMTF Format Identifiers	15
8.2.2 DMTF General FRU Record File Content	15
8.2.3 DMTF Hardware Descriptor FRU Record File Content	15
8.3 IPMI FRU Content Compatibility	16
8.3.1 IPMI and DMTF FRU Combined Structure	17
8.3.2 DMTF Defined IPMI OEM Record	17
9 ANNEX A (informative) Notation	19
10 ANNEX B (informative) Example General FRU Record File in JSON	20
10.1 Data compression	20
10.1.1 MessagePack	20
10.1.2 Gzip	21
11 ANNEX C (informative) Change log	22

1 Foreword

The *Field Replaceable Unit (FRU) Data Specification* (DSP0220) was prepared by the Platform Management Communications Infrastructure (PMCI) Working Group of DMTF.

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:

- Jason Kilpatrick — Dell Technologies (Co-editor)
- Bill Scherer — Hewlett Packard Enterprise (Co-editor)
- Patrick Caporale — Lenovo
- Jim Harford — Broadcom Inc.
- Yuval Itkin — NVIDIA Corporation
- Phillip Leech — Hewlett Packard Enterprise
- Myron Loewen — Solidigm
- David Rudy — Oracle
- Patrick Schoeller — Intel
- Hemal Shah — Broadcom Inc.
- Xiaobing Zhang — Broadcom Inc.

2 Introduction

The *Field Replaceable Unit (FRU) Data Specification* defines the data structures and model for FRU data content.

2.1 Document conventions

2.1.1 Typographical conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.

2.1.2 Reserved and unassigned values

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

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

2.1.3 Byte ordering

Unless otherwise specified, the byte ordering of multibyte numeric fields or multibyte bit fields in this specification shall be "little endian": The lowest byte offset holds the least significant byte (LSB) and higher offsets hold the more significant bytes.

2.1.4 Data types

Unless otherwise specified, the data types provided for fields of structures in this specification shall be as defined in the *Platform Level Data Model (PLDM) Base Specification* [DSP0240](#).

32 **3 Scope**

- 33 The *Field Replaceable Unit (FRU) Data Specification* defines a data content structure for vital product data that describes a platform asset or component. This specification leverages asset information defined in [DSP0257](#), *Platform Level Data Model (PLDM) for FRU Data Specification*. Additional asset information outside of [DSP0257](#) is also defined within this specification.
- 34 This specification defines the data content structure for access in raw format. Access of FRU data via PLDM is defined in [DSP0257](#).

35

4 Normative references

- 36 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.
- 37 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification* v1.3 https://www.dmtf.org/standards/published_documents/DSP0236_1.3.pdf
- 38 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes* v1.12 https://www.dmtf.org/standards/published_documents/DSP0239_1.12.pdf
- 39 DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification* v1.2 https://www.dmtf.org/standards/published_documents/DSP0240_1.2.pdf
- 40 DMTF DSP0242, *Platform Level Data Model (PLDM) for File Transfer Specification* v1.0 https://www.dmtf.org/standards/published_documents/DSP0242_1.0.pdf
- 41 DMTF DSP0257, *Platform Level Data Model (PLDM) for FRU Data Specification* v2.0 https://www.dmtf.org/standards/published_documents/DSP0257_2.0.pdf
- 42 DMTF DSP4014, *DMTF Process for Working Bodies* v2.14 https://www.dmtf.org/sites/default/files/standards/documents/DSP4014_2.14.pdf
- 43 ECMA-404, *The JSON Data Interchange Syntax*, 2nd edition, December 2017 https://ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf
- 44 GNU Gzip (data compression program) <https://www.gnu.org/software/gzip/>
- 45 IPMI, *Platform Management FRU Information Storage Definition* v1.0, Document Revision 1.3 (March 24, 2015) <https://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/ipmi-platform-mgt-fru-info-storage-def-v1-0-rev-1-3-spec-update.pdf>
- 46 ISO/IEC Directives, Part 2, *Principles and rules for the structure and drafting of ISO and IEC documents* <https://www.iso.org/sites/directives/current/part2/index.xhtml>
- 47 MessagePack (compression format for JSON data) <https://msgpack.org/index.html>

48 5 Terms and definitions

49 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined
in this clause.

50 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"), "may",
"need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC
Directives, Part 2, Clause 7. The terms in parentheses are alternatives for the preceding term, for use in exceptional
cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Clause 7
specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal
English meaning.

51 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as described in
ISO/IEC Directives, Part 2, Clause 6.

52 The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives,
Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do not contain normative
content. Notes and examples are always informative elements.

53 For the purposes of this document, the following additional terms and definitions apply.

54 **Expanded format JSON**

55 UTF-8 formatted JSON data that has been optimized for human readability by inserting whitespace between
syntactic elements (this is the normal manner in which JSON data is presented). Contrast with *Minified format JSON*.

56 **Field Replaceable Unit (FRU)**

57 A component within a system that is made to be replaced in the field.

58 **FRU Files Record**

59 A data structure containing metadata that points to a FRU File.

60 **FRU File**

61 A file or storage object that contains the properties and values that describe the FRU.

62 **FRU Bundle**

63 A collection of FRU Files.

64 **Minified format JSON**

65 UTF-8 formatted JSON data that has had all optional whitespace between syntactic elements removed. Contrast with
Expanded format JSON.

66 **Resident FRU file**

67 A FRU file that is located on the same device as the FRU Files Record and can be accessed either directly by
reading the device or via [PLDM for FRU Data \(DSP0257\)](#). A FRU file that is not resident may only be accessible via
PLDM for FRU Data.

68 **6 Symbols and abbreviated terms**

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

70 **CRC-32**

71 Cyclic redundancy check, 32-bit version

72 **FRU**

73 Field Replaceable Unit

74 **IPMI**

75 Intelligent Platform Management Interface

76 **JSON**

77 JavaScript Object Notation

78 **LSb**

79 Least significant bit

80 **LSB**

81 Least significant byte

82 **MSb**

83 Most significant bit

84 **UUID**

85 Universally Unique Identifier

86

7 FRU Content Storage

87

Storage devices that provide direct, raw access through a management interface or bus shall store FRU data content as defined in the [FRU Content Structure clause](#). A storage device, such as a 24C02 compatible EEPROM or a device that can emulate one, is an example of a device that could be used to store the FRU data content. Emulated devices that provide a compatible interface for raw access could also be used for storage. If FRU content data is stored on a device with raw access, it is recommended that the FRU data be stored at the beginning of the storage device (i.e., offset 0x0).

88

The structure of the FRU data content that is defined in this specification does not represent how the data is stored at rest for all implementations. For MCTP capable devices that implement PLDM for FRU and/or PLDM for File Transfer access methods, the data may be stored at the discretion of the implementer—or even dynamically generated—as long as the returned data matches the structure defined by the access methods.

8 FRU Content Structure

This specification defines two methods for storing FRU data content:

1. The FRU Files Record enables one or more FRU Files to be collected together into a FRU Bundle. The individual FRU Files can be stored in any of several formats, including JSON and ENUM. The preferred format for a FRU File is JSON.
2. For compatibility with the [IPMI FRU format](#), a DMTF IPMI OEM Record is also defined in this specification. The DMTF IPMI OEM Record contains an offset that links to the FRU Files Record. This enables bridging from legacy IPMI FRU representations to the DMTF format.

8.1 FRU Files Record

The FRU Files Record is a data structure that holds information about the FRU files. The FRU Files Record contains the following components: Header, File Descriptor Table, and one or more FRU Files. [Figure 1](#) depicts the structure of the FRU Files Record.

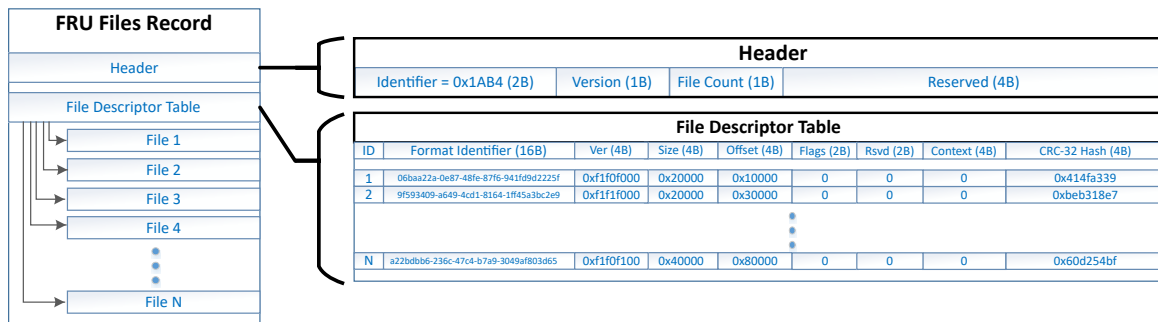


Figure 1 — FRU Files Record

8.1.1 Header

The Header defines the DMTF FRU identifier, the version of the FRU content specification, and the file count for the number of FRU files in the File Descriptor Table. See [Table 1](#) for the Header definition.

8.1.2 File Descriptor Table

The File Descriptor Table is a list of FRU files and associated metadata about those files. See [Table 1](#) for the File Descriptor Table definition.

The ID column in the File Descriptor Table that is displayed in [Figure 1](#) is not literally present in the FRU Files Record data structure. Rather, the value of the ID column is implicit and represents the one-based row number in the File Descriptor Table. The first row in the table has an ID of 0x01.

8.1.3 FRU File(s)

The FRU file contains data content that describes the Field Replaceable Unit. Any entity or organization may define information about FRU. This specification includes a set of properties as defined by DMTF. Some properties in this specification are leveraged from the general FRU record field types defined in [DSP0257](#).

Table 1 — DMTF FRU Files Record Data Format

Component	Field Name	Field Size	Description
Header	Identifier	2 bytes	uint16. DMTF Vendor Identifier that indicates the DMTF FRU is present: 0x1AB4
	Version	1 byte	uint8. Version of the FRU content structure: 0x01 for specification version 1.0
	File Count	1 byte	uint8. The number of files N defined in the File Descriptor Table.
	Reserved	4 bytes	—
File Descriptor Table	File 1 Descriptor	40 bytes	File Descriptor for the first FRU File in the FRU Bundle. The format of the File Descriptor is found in Table 2 below.
	File 2 Descriptor	40 bytes	File Descriptor for the second FRU File in the FRU Bundle. The format of the File Descriptor is found in Table 2 below.
	...	...	...
	File N Descriptor	40 bytes	File Descriptor for the last FRU File in the FRU Bundle. The format of the File Descriptor is found in Table 2 below.
FRU File(s)	—	—	Data content that describes the Field Replaceable Unit. For this specification, see the DMTF General FRU Record File Content clause .

8.1.4 File Descriptor for FRU Files

The format of each entry in the FRU Files Record's File Descriptor Table is as described in Table 2 below.

Table 2 — File Descriptor Data Format

Field Name	Field Size	Description
FRU File Identifier	0 bytes	The FRU File Identifier corresponds to the one-based row number of the file within the File Descriptor Table. This value is stored implicitly by the location of the file's entry; no storage bytes are used to represent it.
Format Identifier	16 bytes	UUID. Format Identifier of the entity or organization that has defined the FRU File properties. The Format Identifier also defines the format and encoding of the FRU File. The Format Identifier is a 16-byte UUID-style value without dashes or hyphens. A Format Identifier is not necessarily unique among all FRU Files in the File Descriptor Table. A FRU may contain multiple FRU Files with the same Format Identifier. These multiple files may be differentiated by other fields in the File Descriptor Table.
Version	4 bytes	ver32. Version of the FRU file definition. This value is formatted in standard PLDM encoding, e.g., 0xF1F0F000.
Size	4 bytes	uint32. Size in the bytes of the FRU file. This value shall be 0xffff ffff if the dynamic flag bit is set to 0.
Offset	4 bytes	sint32. Offset to the start of the FRU file. Specifically, this is the number of bytes between the first byte of the FRU Files Record Header and the first byte of the FRU File, with a negative value indicating that the FRU File appears earlier than the FRU Files Record Header. This value shall be -1 (0xffff ffff) if the resident flag bit is set to 0.

Field Name	Field Size	Description
Flags	2 bytes	bitfield[16]. Bit field definition for Flags field. [15:6] - Reserved [5] - resident 0b - The FRU file is not resident on the FRU device. 1b - The FRU file is resident on the FRU device. [4] - dynamic 0b - The FRU file is static. 1b - The FRU file is dynamically generated on read request. [3] - Reserved for future use. [2] - checksum 0b - No CRC-32 checksum is stored for the FRU File. 1b - A CRC-32 checksum is stored for the FRU File. [1:0] - compression 00b - FRU File is not compressed. 01b - FRU File is compressed with MessagePack. 10b - FRU File is compressed with Gzip. 11b - FRU File is compressed with another algorithm, and a compression header is present at the beginning of the file. See Compression header below.
Reserved	2 bytes	Reserved for future use.
Context	4 bytes	Context is Format Identifier specific and shall be defined by the organization that defines the Format Identifier.
Integrity Checksum	4 bytes	uint32. CRC-32 checksum for the FRU File, including the compression header if present. The checksum shall be stored as zero (0x0000 0000) if the checksum flag bit is set to zero or if the dynamic flag bit is set to one. For this specification, the CRC-32 algorithm with the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (the same as the one used by IEEE 802.3) shall be used for the integrity checksum computation. The CRC computation involves processing a byte at a time with the least significant bit (LSb) first.

112 8.1.5 FRU File Format and Encoding

113 FRU File format and encoding is defined by the Format Identifier in the File Descriptor Table. JSON is the recommended format for FRU Files. When JSON is used for the FRU File format, the uncompressed JSON data shall be encoded using UTF-8. (Note that the JSON standard [ECMA-404](#) requires conformant JSON to use Unicode.) Any entity that defines a Format Identifier shall specify the format of the FRU File in the supporting documentation.

8.1.5.1 Compression header

If and only if the entry for a FRU File contains the value 11b for the compression flags field, a 16-byte compression header shall be prepended to the FRU File. These 16 bytes are a UUID-style Compression Identifier that indicates the specific compression algorithm that was used to compress the FRU File. As with FRU Format identifiers, the specific meaning of a Compression Identifier is defined by the organization that issued it.

8.2 DMTF FRU Files

This clause lists the FRU Files defined by DMTF and their respective Format Identifiers. For each of these Format Identifiers, the Context field is not used and shall be stored as zero (0x0000 0000).

8.2.1 DMTF Format Identifiers

Table 3 — FRU File Format Identifiers

Format Identifier	Description	Data Format
a4f59e2cb8a94fadb09241332c90e65b	DMTF General FRU Record File	JSON; see DMTF General FRU Record File Content
bb04e8da642c11ef937a088e90d7ef14	DMTF Hardware Descriptor FRU Record File	JSON; see DMTF Hardware Descriptor FRU Record File Content

8.2.2 DMTF General FRU Record File Content

The content of the DMTF General FRU Record File is a set of property names and values that describe the field replaceable unit. For compatibility with [PLDM for FRU Data Specification \(DSP0257\)](#), the record fields defined in General FRU Record in [DSP0257](#) are used as the property names of the DMTF General FRU Record File.

See the "DMTF General FRU Record File" clause in [DSP0257](#) for a supported list of property names and for the JSON schema for this FRU File type.

8.2.3 DMTF Hardware Descriptor FRU Record File Content

The DMTF Hardware Descriptor FRU Record File contains additional properties about the FRU that are not defined in [DSP0257](#). This record contains hardware-centric information that may assist in discovering the FRU. Information in this record may be independent of information discovered from an MCTP endpoint.

Table 4 — DMTF Hardware Descriptor FRU Record

Property ID	Property Name	Requirement	Description
0x01	SMBusAddress	Conditional	I2C/SMBus address of the FRU's MCTP endpoint. Required if FRU is accessible via an MCTP endpoint using SMBus/I2C.
0x02	SMBusClock	Conditional	A list of SMBus/I2C clock speeds as supported by the FRU. The fastest clock speed is listed first. Required if FRU is accessible via an MCTP endpoint using SMBus/I2C. Valid values include 1M, 400k, 100k.
0x03	FRUType	Required	FRU type identifier value as defined in DSP0239 , MCTP device type identifier table.
0x04	UUID	Optional	Same UUID value returned by MCTP control command, Get Endpoint UUID.
0x05–0xFF	Reserved	—	Reserved for future use.

The structure of the DMTF Hardware Descriptor FRU Record File is a JSON object, containing a flat (i.e., non-hierarchical) list of string-value pairs. The following string-value pair identifiers are supported:

- "SMBusAddress"
- "SMBusClock"
- "FRUType"
- "UUID"

The above list follows the same ordering as [Table 4](#) for ease of comparison with that table; however, the string-value pairs may appear in any order within a DMTF Hardware Descriptor FRU Record File.

8.3 IPMI FRU Content Compatibility

In addition to serving as a standalone structure, the FRU Files Record may be integrated with an IPMI FRU via the use of a DMTF OEM IPMI Record. This enables bridging between legacy formats and the formats detailed in this specification. In addition to the normal fields required for an IPMI OEM Record, the DMTF OEM record provides an offset pointer to the DMTF FRU Header.

8.3.1 IPMI and DMTF FRU Combined Structure

The DMTF OEM IPMI Record enables bridging from IPMI FRU data to the DMTF FRU Record. It may be placed anywhere within the IPMI FRU records. The DMTF FRU Record should immediately follow the IPMI record data on the FRU device.

Figure 2 details the structure of the DMTF OEM IPMI Record and the manner in which it bridges to the DMTF FRU Record.

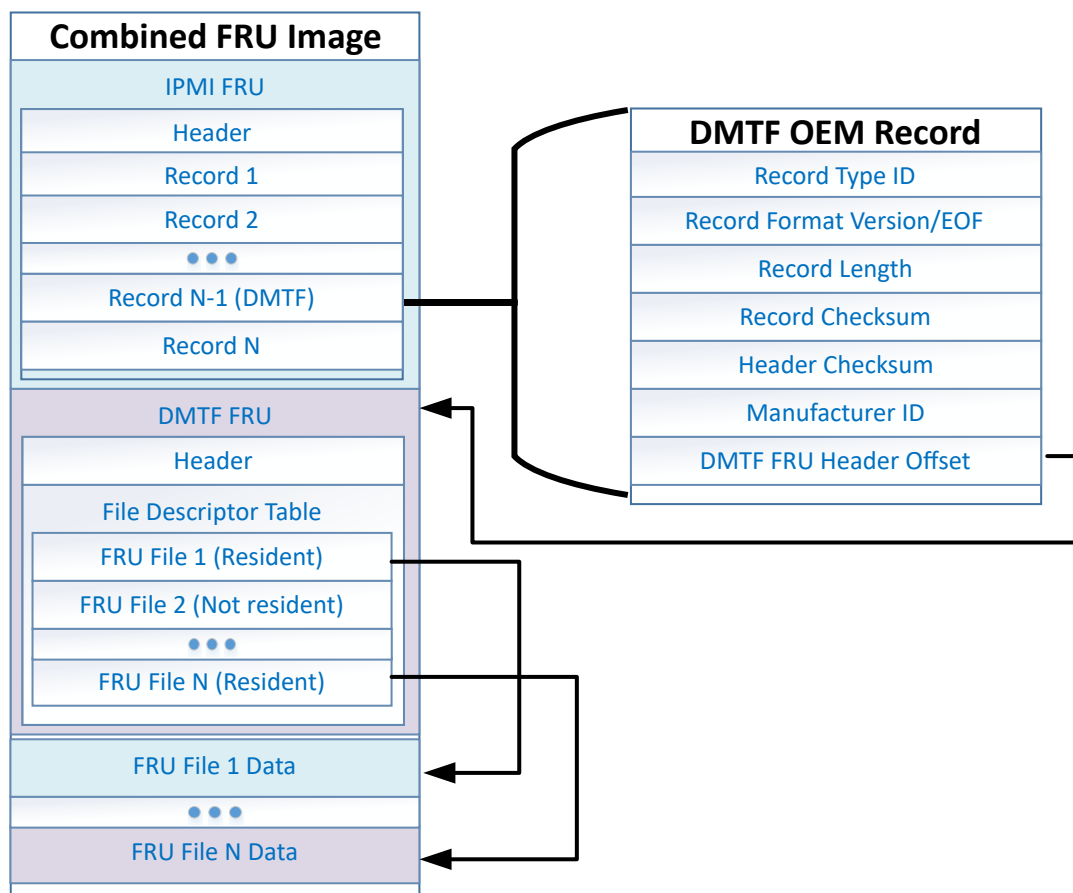


Figure 2 — Combining IPMI and DMTF FRU records

8.3.2 DMTF Defined IPMI OEM Record

The fields of the DMTF OEM IPMI Record are defined in Table 5 below.

143
144
145

Table 5 — DMTF OEM IPMI Record Header

Length	Field	Value
1 byte	Record Type ID The IPMI FRU specification reserves record types IDs 0xC0–0xFF for OEM usage.	0xC0
1 byte	[7:7] – End of Field [6:4] – Reserved [3:0] – Record Format Version (0x02)	0x82 – End of Fields set, DMTF Record is last Record
1 byte	Record Length	Varies
1 byte	Record Checksum	Varies
1 byte	Header Checksum	Varies
3 bytes	Manufacturer ID DMTF Identifier. Always set to 0x1AB4, matching DMTF Vendor ID as registered in PCI-SIG™. The IPMI FRU specification defines a Manufacturer ID as 3 bytes, least significant byte (LSB) first.	0xB41A00
4 bytes	DMTF FRU Header Offset Pointer to the start of the DMTF FRU Files Record Header	—

9 ANNEX A (informative) Notation

Examples of notations used in this document are as follows:

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.
<u>DSP0240</u>	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 (MSb) is on the left, and the least significant bit (LSb) 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.

10 ANNEX B (informative) Example General FRU Record File in JSON

The following is an example DMTF General FRU Record File in [expanded format JSON](#).

```
{
  "Model": "NOA1",
  "PartNumber": "12345",
  "SerialNumber": "SN1234567",
  "Manufacturer": "DMTF",
  "ManufactureDate": "2017-04-01T14:55:33+03:00",
  "Vendor": "DMTF",
  "Version": "A1",
  "EngineeringChangeLevel": "R1",
  "VendorIANA": "3704"
}
```

Figure 3 — DMTF General FRU Record File – Expanded format JSON

This is the same data in [minified format JSON](#).

```
{"Model":"NOA1","PartNumber":"12345","SerialNumber":"SN1234567","Manufacturer":"DMTF","ManufactureDate":"2017-04-01T14:55:33+03:00","Vendor":"DMTF","Version":"A1","EngineeringChangeLevel":"R1","VendorIANA":"3704"}
```

Figure 4 — DMTF General FRU Record File – Minified format JSON

10.1 Data compression

To reduce the size of record data, JSON data may be compressed. This specification recommends using [MessagePack](#) or [Gzip](#) when compressing FRU content data.

10.1.1 MessagePack

The following is compressed data from the General FRU File example. Using [MessagePack](#) to compress the minified version of the JSON data decreased its size from 213 bytes to 177 bytes.

```
00000000 89a5 4d6f 6465 6ca4 4e4f 4131 aa50 6172
00000010 744e 756d 6265 72a5 3132 3334 35ac 5365
```

```

00000020 7269 616c 4e75 6d62 6572 a953 4e31 3233
00000030 3435 3637 ac4d 616e 7566 6163 7475 7265
00000040 72a4 444d 5446 af4d 616e 7566 6163 7475
00000050 7265 4461 7465 b932 3031 372d 3034 2d30
00000060 3154 3134 3a35 353a 3333 2b30 333a 3030
00000070 a656 656e 646f 72a4 444d 5446 a756 6572
00000080 7369 6f6e a241 31b6 456e 6769 6e65 6572
00000090 696e 6743 6861 6e67 654c 6576 656c a252
000000a0 31aa 5665 6e64 6f72 4941 4e41 a433 3730
000000b0 34

```

Figure 5 — DMTF General FRU Record File – MessagePack compression

10.1.2 Gzip

The following is compressed data from the General FRU File example. Using [Gzip](#) to compress the minified version of the JSON data decreased its size from 213 bytes to 179 bytes.

```

00000000 8b1f 0808 88e0 674e 0300 3166 6a2e 6f73
00000010 006e 8e55 0ac1 4082 4514 7df7 6cc6 784b
00000020 8ce3 b309 2c93 7208 148a 53f7 4cbe 27b0
00000030 da4c fa26 1ef7 90b3 9eda 2f73 b79c 8628
00000040 7b06 8461 663d ac52 c9c4 d1f9 8f4e 7a0b
00000050 3266 3a56 5861 efa2 3f5c d2e3 b106 594e
00000060 8e15 9ba6 8ebb 0f93 2f2a ddaa cd3f 88dd
00000070 622c 6990 3a04 5902 6d49 c492 b528 6504
00000080 7800 235d c335 bbcf ff46 06ec 1062 b692
00000090 76d4 9c84 ed40 eee6 c5a8 be03 f242 ce59
000000a0 7de7 3366 2a26 2d05 8b3e 682f e6c7 d6dd
000000b0 0000 0000
000000b3

```

Figure 6 — DMTF General FRU Record File – Gzip compression

170

11 ANNEX C (informative) Change log

Version	Date	Description
1.0.0	2025-03-26	Initial release