Platform Level Data Model (PLDM) for FRU Data Specification

Version: 2.0.0

Document Identifier: DSP0257

Date: 2025-05-19

Version History: https://www.dmtf.org/dsp/DSP0257

Supersedes: 1.0.1

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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 *Platform Level Data Model (PLDM) for FRU Data Specification* (DSP0257) 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:

- Jim Harford — Broadcom Inc. (Editor)
- Phil Chidester — Dell Technologies (past Editor)
- Eliel Louzoun — Intel Corp. (past Editor)
- Alan Berenbaum — SMSC
- Patrick Caporale — Lenovo
- Hoan Do — Broadcom Inc.
- Jason Kilpatrick — Dell Technologies
- Ed Klodnicki — IBM
- John Leung — Intel Corporation
- Bill Scherer — Hewlett Packard Enterprise
- Hemal Shah — Broadcom Inc.
- Tom Slaight — Intel Corporation
- Bob Stevens — Dell Technologies

20 2 Introduction

21 The *Platform Level Data Model (PLDM) FRU Data Specification* defines messages, data structures, and data types used to access FRU (Field Replaceable Unit) Data. Traditional FRU data includes inventory asset information, such as the serial number, part number, and manufacturer for a field replaceable unit. Additional future use of FRU data is envisioned to potentially address more general manageability concerns, such as configuration and maintenance requirements.

22 Version 2.0 and beyond is not backward-compatible with version 1.X of this specification.

23 2.1 Document conventions

24 2.1.1 Typographical conventions

25 The following typographical conventions are used in this document:

- Document titles are marked in *italics*.
    - Important terms that are used for the first time are marked in *italics*.
    - Terms include a link to the term definition in the "Terms and definitions" clause, enabling easy navigation to the term definition.

26 2.1.2 Reserved and unassigned values

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

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

29 2.1.3 Byte ordering

30 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 and higher offsets hold the more significant bytes.

31 2.1.4 Other conventions

32 See ANNEX B of this specification for other conventions.

33 3 Scope

34 DSP0257, *Platform Level Data Model for FRU Data Specification,* defines PLDM messages for reading and writing data associated with field replaceable units, such as add-in cards, local storage, and system motherboards. This specification describes Platform Level Data Model (PLDM) data structures and commands for transferring FRU data between the components of a platform management subsystem.

35 Starting with version 2.0 of this specification, there exist three distinct styles of PLDM access to FRU data:

- FRU data is identified and formatted in the context of a *FRU Record Table*, which was first specified in version 1.0 of this specification. Although the data format within PLDM messages remains the same, FRU data access is not backward-compatible to version 1.X, as a result of migration to PLDM common multipart transfer procedures in version 2.0. The FRU Record Table context is defined in FRU Record Table.
- Each individual atomic unit of FRU data is identified and formatted in the context of a *FRU Data Item*, which is fully defined in FRU Data Item.
- Aggregated FRU data is identified and formatted in the context of a *FRU Data File*. That context, plus the format of DMTF-specified FRU data files, are defined in FRU Data File.

36 3.1 Relationship to other DMTF specifications

37 This specification is closely related to DSP0220 in the support of accessing FRU Data Files. This specification depends upon DSP0220 for the definition of the *FRU Files Record* data structure, which provides a listing of FRU Data Files present on a FRU, plus certain metadata for each FRU Data File. The listing of FRU Data Files and associated per-file metadata are used in the FindFRUFiles and GetFRUFileMetadata commands herein. In turn, DSP0220 depends upon this specification for the data format of FRU Data Files defined by DMTF which are expected to be accessed via PLDM messages. (Some DMTF-defined FRU data is typically needed as part of system initialization before PLDM messaging is available via the platform management subsystem.)

38 This specification leverages DSP0242 for the definition of procedures to read files via PLDM messages. Specifically, FRU Data Files are read from a FRU device via procedures defined in DSP0242.

39 This specification leverages DSP0240 for the definition of PLDM common multipart transfer procedures, which are used in the GetFRURecordTable, SetFRURecordTable, and GetFRURecordByOption commands.

40 This specification relies on DSP0248 for the definition of a FRU Record Set PDR. Note that the addition of a new field, *AccessFlags*, to the FRU Record Set PDR is planned for version 1.4 of DSP0248. The existence of the AccessFlags field is intended to reduce the number of PLDM discovery messages to a FRU device; thus, conformance to the present specification is not strictly conditioned upon version 1.4 of DSP0248. The presence or absence of the AccessFlags field can be determined by checking the length of the FRU Record Set PDR.

41 3.2 Relationship to other non-DMTF specifications

42 In response to market and industry demands for higher performance, there exists a foreseeable increase in hardware

complexity and the burden of properly configuring future hardware. Other standards organizations are likely to define new FRU data requirements to support emerging technologies.

43 This specification maintains a "black box" approach to other standards bodies' FRU data requirements, supporting access to non-DMTF-defined FRU Data Files via the procedures in this specification, while assuming that the data format for such FRU data and certain per-file metadata is defined by the respective standards organizations.

44 4 Normative references

45 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.

46 DMTF DSP0220, *Field Replaceable Unit (FRU) Data Specification, 1.0* https://www.dmtf.org/sites/default/files/standards/documents/DSP0220_1.0.pdf

47 DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification, 1.2* https://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.2.pdf

48 DMTF DSP0242, *Platform Level Data Model (PLDM) for File Transfer Specification, 1.0* https://www.dmtf.org/sites/default/files/standards/documents/DSP0242_1.0.pdf

49 DMTF DSP0245, *Platform Level Data Model (PLDM) IDs and Codes Specification 1.4* https://www.dmtf.org/sites/default/files/standards/documents/DSP0245_1.4.pdf

50 DMTF DSP0248, *Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification 1.3* https://www.dmtf.org/sites/default/files/standards/documents/DSP0248_1.3.pdf

51 ANSI/IEEE Standard 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic* (12 October 1985)

52 ISO/IEC Directives, Part 2, *Principles and rules for the structure and drafting of ISO and IEC documents*, Ninth edition (2021) https://www.iso.org/sites/directives/current/part2/index.xhtml

53 5 Terms and definitions

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

55 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 parenthesis 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.

56 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clauses 22, 6, and 20, respectively.

57 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.

58 Refer to DSP0240 for terms and definitions that are used across the PLDM specifications. For the purposes of this document, the following additional terms and definitions apply.

59 **Downstream FRU**

60 A FRU device that is dependent upon a primary FRU to facilitate access to certain data associated with the downstream FRU.

61 **Field Replaceable Unit (FRU)**

62 A component within a system that is designed to be replaced in the field, if necessary.

63 **Platform Descriptor Record**

64 A set of data that is used to provide semantic information about sensors, effecters, monitored or controller entities, and functions and services within a PLDM implementation.

65 **Primary FRU**

66 A FRU device that functions as a PLDM terminus, supporting the PLDM commands defined herein to facilitate access to certain data associated with the primary FRU and any downstream FRUs attached to the primary FRU.

67 **Record Set Identifier**

68 A unique value within the scope of a primary FRU device that is mapped to a FRU Record Set. This identifier is closely related to both the *FRURecordSetIdentifier* field of a FRU Record Set PDR and the *RecordSetID* field of several PLDM commands. See FRU Record Set PDR for more information.

6 Symbols and abbreviated terms

Refer to DSP0240 for symbols and abbreviated terms that are used across the PLDM specifications. For the purposes of this document, the following additional symbols and abbreviated terms apply.

FRU

Field Replaceable Unit

IANA

Internet Assigned Numbers Authority

PDR

Platform Descriptor Record

RecordSetID

Record Set Identifier

7 PLDM Conventions

Refer to DSP0240 for conventions, notations, and data types that are used across the PLDM specifications.

Within this specification, the PLDM data type ItemName is defined.

82 8 PLDM for FRU Data version

83 The version of this *Platform Level Data Model (PLDM) for FRU Data Specification* shall be 2.0.0 (major version number 2, minor version number 0, update version number 0, and no alpha version).

84 For the GetPLDMVersion command described in DSP0240, the version of this specification is reported using the encoding as 0xF2F0F000.

85 9 FRU Record Set PDR

86 The FRU Record Set PDR is used to describe characteristics of the PLDM FRU Record Set Data. The information can be used to locate a Terminus that holds FRU Record Set Data in order to access that data. The PDR also identifies the particular Entity that is associated with the FRU information.

87 The FRU Record Set PDR is defined in DSP0248.

88 Although not strictly required, this specification generally assumes that FRU Record Set PDRs are supported by a FRU that functions as a PLDM terminus (a "primary FRU"). For scenarios in which additional FRUs whose FRU data is accessed via the same PLDM terminus ("downstream FRUs"), support for FRU Record Set PDRs is mandatory, as these PDRs are necessary to distinguish which FRU is being targeted by a message request to the PLDM terminus shared by those FRUs.

89 Within any FRU Record Set PDR, the FRURecordSetIdentifier field shall be non-zero. For certain commands within this specification, 0 is a special value for the RecordSetID parameter. The GetFRURecordByOption command herein uses RecordSetID = 0 to be a request across all possible FRU data records. The ReadFRUDataItem, WriteFRUDataItem, FindFRUFiles, and GetFRUFileMetadata commands herein use RecordSetID = 0 to request the "default" FRU record set, which is defined as the FRU general record set of the FRU functioning as a PLDM terminus. For additional details, see the individual command descriptions in PLDM Commands for FRU Data.

90 10 FRU Record Table

91 The *FRU Record Table* is an abstract data structure that was originally defined as a means to transfer FRU data associated with multiple FRU PDRs within the same PLDM command. Specifically, the GetFRURecordTable, SetFRURecordTable, and GetFRURecordByOption commands are used to transfer a complete or partial FRU Record Table. The GetFRURecordTableMetadata command is used to query the metadata associated with a FRU device's FRU Record Table.

92 A management controller can discover whether or not a FRU device supports PLDM Access in the context of FRU Record Tables by sending a GetPLDMCommands request (defined in DSP0240) to the FRU device. The response contains a list of PLDM commands supported by the FRU device. If the GetFRURecordTableMetadata and GetFRURecordTable commands are supported, then PLDM access within this context of FRU Record Tables is considered supported by the FRU device. Additionally, if the *AccessFlags* field is present in a FRU Record Set PDR, then the value of the *FRURecordTableAccess* bit specifies whether or not that record set's data can be accessed via FRU Record Tables.

93 Examples of accessing FRU data and associated metadata within this PLDM access style context can be found in ANNEX A (informative) PLDM for FRU Data Examples.

94 The components of a FRU Record Table need not be continuous in the FRU device's memory. The FRU Record Table is defined as "data on the move" (within a PLDM command), not as "data at rest" (memory storage).

95 The format of the FRU Record Table remains unchanged for similarity to previous versions. However, starting with version 2.0, the semantics of various components of the FRU Record Table have been refined or extended to support features introduced in version 2.0:

- The FRU Record Set Identifier component is used within the contexts of FRU Data Item and FRU Data File to determine the physical location associated with the FRU data, which could be either the primary FRU Device or a downstream FRU Device.
- Within the context of a *FRU Data Item*, the FRU Record Type and FRU Field Type components are widened to 16 bits each and become subfields of the 32-bit "ItemName" field.
- Within the context of a *FRU Data Item*, the FRU Field Length component is widened to 16 bits and becomes the "ItemLength" field.
- Within the context of a *FRU Data File*, the analogous length depends on the file format, and some file formats (e.g., JSON) offer unlimited string lengths.

96 The following subclauses define the components of the FRU Record Table data structure.

97 10.1 FRU Record Set Identifier

98 The FRU Record Set Identifier is a unique number that identifies the FRU record set.

99 10.2 FRU Record Type

100 The FRU Record Type identifies the FRU record and is defined in Table 3.

101 10.3 Number of FRU Fields

102 The Number of FRU fields indicates the number of fields that are included in a FRU record.

103 10.4 FRU Encoding Types

104 String value types for a specific FRU Record are defined in the Encoding Type field. The Encoding Type shall apply for all FRU fields in a FRU Record with a string format. The FRU Encoding Types are defined in Table 1.

105 All strings within a FRU Record Table shall be preceded by a length variable where a length of zero indicates that the field is not used in this specific FRU Field. String lengths shall be in bytes. Strings are not null terminated and are limited to a 255-byte size. FRU Record Set Identifiers and their associated record types shall appear contiguous within the FRU Record Table.

106 10.5 FRU Field Type, Length, and Value

107 FRU fields within a FRU Record Table are defined by a Type, Length, and Value (TLV). The Type is defined in Table 4, which also defines the Field format and length ranges.

108 Table 1 specifies the format used for the PLDM FRU Data Format.

109 **Table 1 — PLDM FRU Record Data Format**

Size	Type	Field
2 bytes	uint16	FRU Record Set Identifier
1 byte	uint8	FRU Record Type
1 byte	uint8	Number of FRU fields
1 byte	uint8	Encoding Type for FRU fields 0 = Unspecified 1 = strASCII 2 = strUTF-8 3 = strUTF-16 4 = strUTF-16LE 5 = strUTF-16BE 6–255 = reserved
1 byte	uint8	FRU Field Type #1

Size	Type	Field
1 byte	uint8	FRU Field Length #1
Up to 255 bytes (see Table 4)	Determined by FRU Field Type (see Table 4)	FRU Field #1 Value
1 byte	uint8	FRU Field #2 Type
1 byte	uint8	FRU Field #2 Length
Up to 255 bytes (see Table 4)	Determined by FRU Field Type / Length (see Table 4)	FRU Field #2 Value
...	...	...
1 byte	uint8	FRU Field #n Type
1 byte	uint8	FRU Field #n Length
Up to 255 bytes (see Table 4)	Determined by FRU Field Type / Length (see Table 4)	FRU Field #n Value

110 Table 2 specifies the format used for the PLDM FRU Record Table Format.

111 **Table 2 — PLDM FRU Record Data Table Format**

Field
FRU Record Data #1 (See Table 1)
FRU Record Data #2
FRU Record Data #3
...
FRU Record Data #n

112 Table 3 defines the FRU Record Types.

113 **Table 3 — FRU Record Type Definitions**

Record Type	Description
0	Reserved
1	General FRU Record
2–253	Reserved
254	OEM FRU Record
255	Reserved

114 Table 4 defines the General FRU record field type definitions. Support for General FRU record field types 2–6 (Model, Part Number, Serial Number, Manufacturer, and Manufacture Date) are mandatory; support for all other General FRU record field types is optional.

115 **Table 4 — General FRU Record Field Type Definitions**

Field Type Number	Field Type Description	Field Format	Length
0	Reserved	N/A	N/A
1	Chassis Type	string	1–255 bytes
2	Model	string	1–255 bytes
3	Part Number	string	
4	Serial Number	string	
5	Manufacturer	string	
6	Manufacture Date	timestamp104	13 bytes
7	Vendor	string	
8	Name	string	
9	SKU	string	
10	Version	string	
11	Asset Tag	string	
12	Description	string	
13	Engineering Change Level	string	
14	Other Information	string	
15	Vendor IANA	uint32	4 bytes
16	Spare Part Number	string	
17–255	Reserved	N/A	

116 Table 5 defines the OEM FRU Record field type definitions.

117 When the record type is set to OEM = 254, then that record shall contain one field of field type 1 that contains the Vendor IANA. Other field types 2–254 are defined by the OEM.

118 **Table 5 — OEM FRU Record Field Type Definitions**

Field Type Number	Field Type Description	Field Format
0	Reserved	N/A
1	Vendor IANA	uint32
2–254	OEM specific field types	OEM specific
255	Reserved	N/A

119 10.6 FRU Record Table carriage

120 When a complete or partial FRU Record Table is transferred via the GetFRURecordTable, SetFRURecordTable, or GetFRURecordByOption commands described herein, it is appended by a checksum calculation used to protect the data integrity of the payload, as shown in Table 6.

121 **Table 6 — FRU Record Table Carriage**

Byte	Type	Field
Variable	—	**FRU Record Data** (one or more) See Table 1 for the PLDM representation of PLDM FRU Record Data.
Variable	uint8[ ]	**Pad** 0 to 3 number of pad bytes. The value stored in each pad byte is 0x00. The transmitter can compute the number of pad bytes from the FRU data by using the following algorithm: Let L be the total number of bytes in the FRU Record Data excluding the pad and the integrity checksum. if (L modulo 4 == 0) then NumPadBytes = 0; else NumPadBytes = 4 − (L modulo 4); The receiver can determine the number of pad bytes within an incoming message containing FRU data by using the following method: 1. The receiver first processes all FRU Record Data at the beginning of the message; the remaining bytes of the message contain the pad bytes and four bytes of data integrity checksum. 2. Then the size of the Pad field is the number of remaining bytes of the message minus 4.
	uint32	**FRUDataStructureIntegrityChecksum** Integrity checksum on the FRU data including the pad bytes (if any). It is calculated starting at the first byte of the PLDM representation of FRU data. 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 first.

122 11 FRU Data Item

123 The FRU Data Item is defined as an atomic unit of data that represents specific FRU-related information of interest, such as Part Number, Version, or Asset Tag. Additional examples of FRU data items may be found in Table 4, where they appear as general FRU record field types. Beginning with version 2.0 of this specification, access to individual FRU data items is streamlined via the ReadFRUDataItem and WriteFRUDataItem commands described in PLDM Commands for FRU Data.

124 A management controller can discover whether or not a FRU device supports PLDM Access in the context of FRU Data Items by sending a GetPLDMCommands request (defined in DSP0240) to the FRU device. The response contains a list of PLDM commands supported by the FRU device. If the ReadFRUDataItem and WriteFRUDataItem commands are supported, then PLDM access within this context of FRU Data Items is considered supported by the FRU device. Additionally, if the *AccessFlags* field is present in a FRU Record Set PDR, then the value of the *FRUDataItemAccess* bit specifies whether or not that record set's data can be accessed via FRU Data Items.

125 Examples of accessing FRU data within this PLDM access style context can be found in ANNEX A (informative) PLDM for FRU Data Examples.

126 11.1 ItemName Definition

127 Each FRU Data Item supported by a FRU is uniquely identified within the scope of a related FRU Record Set by an *ItemName*, each of which is composed in the following manner:

- *Item Area* is the most significant 16 bits of an ItemName; it is somewhat analogous to a FRU Record Type, defined in FRU Record Table.
- *Item Index* is the least significant 16 bits of an ItemName; it is somewhat analogous to a FRU Record Field Type, defined in FRU Record Table. Within a given FRU Item Area, each FRU Item Index shall be a number from 1 to N, assigned sequentially.

128 11.2 FRU ItemName Values

129 Initial supported FRU ItemNames have been chosen for similarity with prior versions, wherever possible. The FRU ItemNames are defined as follows:

- FRU Item Area = 1 is equivalent to FRU Record Type = 1 (General FRU Record). Within that FRU Item Area, each FRU Item Index is equivalent to the corresponding FRU Field Type defined in Table 4.
- FRU Item Area = 254 is equivalent to FRU Record Type = 254 (OEM FRU Record). Within that Item Area, Item Index = 1 is assigned "Vendor IANA".
- All other FRU Item Areas are reserved for future definition.

130 12 FRU Data File

131 A significant subset of FRU data may be accessed in aggregations structured as *FRU files*. Such aggregations are defined by DMTF and other standards bodies. Motivations for supporting FRU files include the ability to support other standards bodies' new definitions with a minimum of DMTF document maintenance (a "black box" approach to such FRU data), as well as supporting modern, self-describing data formats (e.g., JSON). Only read access is supported.

132 Each FRU file on a FRU device should be represented by a File Descriptor PDR. A FRU device maintains an internal table of accessible FRU files. When a primary FRU receives a FindFRUFiles PLDM request, the response contains a listing of FRU files that meet the search criteria specified in the request. For each file in the FindFRUFiles response, a *File Identifier* field contains an opaque reference to that file. This File Identifier field contains the same value and carries the same semantics as the FileIdentifier field contained in a File Descriptor PDR for the FRU file, if one exists.

133 The actual read access of the FRU file is performed via PLDM file transfer. A File Identifier discovered either via a FindFRUFiles command or by parsing File Descriptor PDRs associated with the FRU is then provided as a parameter to the DfOpen command specified in DSP0242.

134 The GetFRUFileMetadata PLDM command provides certain metadata associated with each FRU file. This metadata can be useful in file searches; however, the additional metadata gained via this command is often not needed.

135 A management controller can discover whether or not a FRU device supports PLDM Access in the context of FRU Data Files by sending a GetPLDMCommands request (defined in DSP0240) to the FRU device. The response contains a list of PLDM commands supported by the FRU device. If the FindFRUFiles and GetFRUFileMetadata commands are supported, then PLDM access within this context of FRU Data Files is considered supported by the FRU device. Additionally, if the *AccessFlags* field is present in a FRU Record Set PDR, then the value of the *FRUDataFileAccess* bit specifies whether or not that record set's data can be accessed via FRU Data Files.

136 Examples of accessing FRU data and associated metadata within this PLDM access style context can be found in ANNEX A (informative) PLDM for FRU Data Examples.

137 12.1 DMTF General FRU Record File

138 This document specifies a FRU Data File structure designed to aggregate a significant subset of FRU record types listed in Table 4. Not all FRU record types appearing in that table are supported for inclusion into the DMTF General FRU Record File. Many items in the table are considered writable after the final assembly of the FRU, but this specification does not support the writing of FRU data files. Moreover, the writing of FRU data files in the field presents significant implementation challenges. Additionally, any FRU record types that now seem outdated have been removed from the list of supported FRU record types appearing in a DMTF General FRU Record File.

139 The structure of the DMTF General 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:

- "Model"
- "PartNumber"
- "SerialNumber"

- "Manufacturer"
- "ManufactureDate"
- "Vendor"
- "SKU"
- "Version"
- "EngineeringChangeLevel"
- "VendorIANA"
- "SparePartNumber"

140 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 General FRU Record File.

141 13 PLDM Commands for FRU Data

142 Table 7 defines the PLDM command codes defined in the following subclause for the PLDM for FRU data transfer. The PLDM FRU messages have their own PLDM message type, which is defined in DSP0245.

143 **Table 7 — PLDM for FRU Data Command Codes**

Command	Code Value	Requirement	PLDM Access Style
GetFRURecordTableMetadata	0x01	Conditional	FRU Record Table
GetFRURecordTable	0x02	Conditional	FRU Record Table
SetFRURecordTable	0x03	Optional	FRU Record Table
GetFRURecordByOption	0x04	Optional	FRU Record Table
Reserved	0x05..0x0F	Reserved	N/A
ReadFRUDataItem	0x10	Mandatory	FRU Data Item
WriteFRUDataItem	0x11	Mandatory	FRU Data Item
Reserved	0x12..0x17	Reserved	N/A
FindFRUFiles	0x18	Mandatory	FRU Data File
GetFRUFileMetadata	0x19	Mandatory	FRU Data File

144 The requirements specified in Table 7 are relative to the services provided by the PLDM terminus. Specifically, the "Conditional" requirement for the GetFRURecordTableMetadata and GetFRURecordTable commands is that these 2 commands are mandatory when a primary FRU device supports PLDM access via FRU Record Tables (as defined in FRU Record Table) for at least one FRU Record Set.

145 When a FRU device receives a PLDM command request with data formatted per the relevant Request Data section, and the request is successfully processed, then the FRU device shall respond with data formatted per the relevant Response Data section. For a non-SUCCESS CompletionCode, only the CompletionCode field of the Response Data shall be returned.

146 13.1 GetFRURecordTableMetadata

147 The GetFRURecordTableMetadata command, described in Table 8, is used to get the FRU Record Table metadata information that includes the FRU Record major version, the FRU Record minor version, the size of the largest FRU Record data, the total length of the FRU Record Table, the total number of FRU Record Data structures, and the integrity checksum on the FRU Record Table data.

148 This command mechanism exists within the PLDM access style context herein labeled FRU Record Table.

149 **Table 8 — GetFRUTableMetadata Command Format**

Byte	Type	Request Data
—	—	No Request Data

Byte	Type	Response Data

0	enum8	**CompletionCode**
		Possible values:
		{
		PLDM_BASE_CODES,
		NO_FRU_DATA_STRUCTURE_TABLE_METADATA=0x83
		}
		If there exists no valid metadata table, the FRU device shall return CompletionCode NO_FRU_DATA_STRUCTURE_TABLE_METADATA.

1	uint8	**FRUDataMajorVersion** The major version of the PLDM FRU Data specification with which the FRU Record Table is compliant. For an implementation compliant with this specification, the FRUDataMajorVersion shall be set to 0x02.
2	uint8	**FRUDataMinorVersion** The minor version of the PLDM FRU Data specification with which the FRU Record Table is compliant. For an implementation compliant with this specification, the FRUDataMinorVersion shall be set to 0x00.
3:6	uint32	**FRUTableMaximumSize** The maximum number of data bytes that can be stored in the FRU Record Table using the SetFRURecordTable command. A value of 0x00000000 in this field means that SetFRURecordTable command is not supported. A value of 0xffffffff in this field means unknown and cannot be specified.
7:10	uint32	**FRUTableLength** Total length of the FRU table in bytes
11:12	uint16	**Total number of Record Set Identifiers in table**
13:14	uint16	**Total number of records in table**
15:18	uint32	**FRUDataStructureTableIntegrityChecksum (CRC-32)** Integrity checksum shall be computed on the FRU Record Table data as shown in Table 6 excluding pad bytes. See Table 6 for more information about this integrity checksum.

150 13.2 GetFRURecordTable

151 The GetFRURecordTable command is used to read the FRU Record Table data. This command is a specific implementation of the DSP0240 PLDM Base Specification MultipartReceive command, with identical request and response formats. When more than one message is used to transfer the FRU Record Table, the response messages contain the non-overlapping contiguous portions of the FRU Record Table as defined in Table 6. By combining the portions of the FRU Record Table from the response messages, the entire FRU Record Table can be reconstructed.

152 This command mechanism exists within the PLDM access style context herein labeled FRU Record Table.

153 The following requirements apply:

- All transfers shall consist of a single section, beginning at offset zero and transferring the entire table.
- Handling of aborted transfers, which is defined in DSP0240 to be protocol specific, shall be that the requester discards all received portions of the FRU Record Table data.
- Values of the DataTransferHandle field are implementation-specific.
- The TransferContext field, which is defined in DSP0240 to be protocol specific, shall be set to 0 as further explained in the paragraph immediately below.

154 The 32-bit TransferContext field carries identical semantics across the GetFRURecordTable, SetFRURecordTable, and GetFRURecordByOption commands: the most significant 16 bits are the Record Set Identifier, the next 8 bits are the Record Type, and the least significant 8 bits are the Field Type. A value 0 for any of the three subfields acts as a wildcard. Thus, a TransferContext field of 0 means that all Record Set Identifiers, all Record Types, and all Field Types shall be included in the response, which is behavior equivalent to the original version of this command.

155 13.3 SetFRURecordTable

156 The SetFRURecordTable command is used to write the FRU Record Table data. This command is a specific implementation of the DSP0240 PLDM Base Specification MultipartSend command, with identical request and response formats. When more than one message is used to transfer the FRU Record Table, the request messages contain the non-overlapping contiguous portions of FRU Record Table as defined in Table 6. By combining the portions of FRU Record Table from the request messages, the entire FRU Record Table can be reconstructed.

157 This command mechanism exists within the PLDM access style context herein labeled FRU Record Table.

158 The following requirements apply:

- All transfers shall consist of a single section, beginning at offset zero and transferring the entire table.
- No portion of the updated FRU Record Table shall take effect before all portions of the updated FRU Record Table are received by the responder.
- Handling of aborted transfers, which is defined in DSP0240 to be protocol specific, shall be that the responder discards all received portions of the FRU Record Table data.
- Values of the DataTransferHandle field are implementation-specific.
- The TransferContext field, which is defined in DSP0240 to be protocol specific, shall be set to 0 as further explained in the paragraph immediately below.

159 The 32-bit TransferContext field carries identical semantics across the GetFRURecordTable, SetFRURecordTable, and GetFRURecordByOption commands: the most significant 16 bits are the Record Set Identifier, the next 8 bits are the Record Type, and the least significant 8 bits are the Field Type. A value 0 for any of the three subfields acts as a wildcard. Thus, a TransferContext field of 0 means that all Record Set Identifiers, all Record Types, and all Field Types shall be included in the request, which is behavior equivalent to the original version of this command.

160 13.4 GetFRURecordByOption

161 The GetFRURecordByOption command is used to read a subset of the FRU Record Table data. This command is a

specific implementation of the DSP0240 PLDM Base Specification MultipartReceive command, with identical request and response formats. When more than one message is used to transfer the FRU Record Table, the response messages contain the non-overlapping contiguous portions of FRU Record Table as defined in Table 6. By combining the portions of FRU Record Table from the response messages, the requested subset of the FRU Record Table can be reconstructed.

162 This command mechanism exists within the PLDM access style context herein labeled FRU Record Table.

163 The following requirements apply:

- All transfers shall consist of a single section, beginning at offset zero and transferring the requested subset of the entire table.
- Handling of aborted transfers, which is defined in DSP0240 to be protocol specific, shall be that the requester discards all received portions of the FRU Record Table data.
- Values of the DataTransferHandle field are implementation-specific.
- The TransferContext field, which is defined in DSP0240 to be protocol specific, shall be set as further explained in the paragraph immediately below.

164 The 32-bit TransferContext field carries identical semantics across the GetFRURecordTable, SetFRURecordTable, and GetFRURecordByOption commands: the most significant 16 bits are the Record Set Identifier, the next 8 bits are the Record Type, and the least significant 8 bits are the Field Type. The values for each of the 3 subfields are defined as follows:

- For the Record Set Identifier subfield: a value of 0 means all record sets, and the values 0x0001–0xffff are used to indicate a specific record set, which is behavior equivalent to the original version of this command.
- For the Record Type subfield: a value of 0 means all record types, and the values 0x01–0xff are used to indicate a specific record type, which is behavior equivalent to the original version of this command. If the Field Type subfield is non-zero, then the Record Type subfield shall also be non-zero.
- For the Field Type subfield: a value of 0 means all field types, and the values 0x01–0xff are used to indicate a specific field type, which is behavior equivalent to the original version of this command. If the Field Type subfield is non-zero, then the Record Type subfield shall also be non-zero.

165 13.5 ReadFRUDataItem

166 The ReadFRUDataItem command, described in Table 9, is used to query a single FRU data item. This command mechanism exists within the PLDM access style context herein labeled FRU Data Item.

167 **Table 9 — ReadFRUDataItem Command Format**

Byte	Type	Request Data
0:1	uint16	**RecordSetID** Possible values: {Default record set=0x0000, Specific record set=0x0001–0xffff}
2:5	ItemName	**FRUItemName** A structured field that identifies a single FRU data item (e.g., Part Number, Version, Asset Tag)

Byte	Type	Response Data
0	enum8	**CompletionCode** Possible values: { PLDM_BASE_CODES }
1:2	uint16	**ItemLength** The length, in bytes, of the ItemValue field within this PLDM message response
variable	—	**ItemValue** The current value of the single FRU data item being read, encoded in strUTF-8 format, without null termination

168 13.6 WriteFRUDataItem

169 The WriteFRUDataItem command, described in Table 10, is used to modify a single FRU data item. This command mechanism exists within the PLDM access style context herein labeled FRU Data Item.

170 **Table 10 — WriteFRUDataItem Command Format**

Byte	Type	Request Data
0:1	uint16	**RecordSetID** Possible values: {Default record set=0x0000, Specific record set=0x0001–0xffff}
2:5	ItemName	**FRUItemName** A structured field that identifies a single FRU data item (e.g., Part Number, Version, Asset Tag)
6:7	uint16	**ItemLength** The length, in bytes, of the ItemValue field within this PLDM message request
variable	—	**ItemValue** The update value of the single FRU data item being written, encoded in strUTF-8 format, without null termination

Byte	Type	Response Data
0	enum8	**CompletionCode** Possible values: { PLDM_BASE_CODES }

171 13.7 FindFRUFiles

172 The FindFRUFiles command, described in Table 11, is used to find FRU files that either match a specified FormatID or all files within a single RecordSetID. The command response contains a list of files that match the search criteria specified in the command request. For each such file referenced in the command response, the file attributes listed in Table 12 are reported in the response.

173 This command mechanism exists within the PLDM access style context herein labeled FRU Data File.

174 **Table 11 — FindFRUFiles Command Format**

Byte	Type	Request Data
0:1	uint16	**RecordSetID** Possible values: {Default record set=0x0000, Specific record set=0x0001–0xffff}
2:17	UUID	**FormatID** A 16-byte field that identifies the type and format of a FRU file. A special FormatID value of all zeros is used as a wildcard match, meaning all FRU files within a single RecordSetID will be listed in the response.

Byte	Type	Response Data
0	enum8	**CompletionCode** Possible values: { PLDM_BASE_CODES }
1		**Reserved** This field shall be set to 0.
2:3	uint16	**NumMatchingFiles** The number of files matching the request
variable	FRUFileAttributes[]	**FRUFileList** An array in which each element contains various attributes, for all FRU files matching the command request

175 For each of the matching files referenced in the command response, the file attributes listed in Table 12 are reported.

176 **Table 12 — FRU File Attributes in FindFRUFiles Response**

Byte	Type	Field
0:1	uint16	**File Identifier** An opaque handle that is used to reference a FRU file in subsequent DfOpen commands
2:3		**Reserved** This field shall be set to 0.
4:19	UUID	**FRU File FormatID** A 16-byte field that identifies the type and format of a FRU file
20:23	uint32	**FRU File Version** The version of FRU file, within a FormatID-specific format and release history
24:27	uint32	**FRU File Length** The length, in bytes, of the FRU file content
28:31	uint32	**FRU File Context** The definition of this field is FormatID-specific; for DMTF-specified FormatIDs, the value is 0.
32:33	bitfield16	**FRU File Flags** The definition of this field is FormatID-specific; for DMTF-specified FormatIDs, the value is 0.
34:35	uint16	**FRUFileIndex** An index, associated with this FRU File, to an internal table within the overall FRU

177 13.8 GetFRUFileMetadata

178 The GetFRUFileMetadata command, described in Table 13, is used to obtain all metadata associated with a FRU file. Note that the use of this command is somewhat rare; typically the FindFRUFiles command yields sufficient information to discover, select, and specify a FRU file in subsequent ReadFRUFile commands.

179 This command mechanism exists within the PLDM access style context herein labeled FRU Data File.

Table 13 — GetFRUFileMetadata Command Format

Byte	Type	Request Data
0:1	uint16	**RecordSetID** Possible values: {Default record set=0x0000, Specific record set=0x0001–0xffff}
2:3	uint16	**FRUFileIndex** An index, associated with this FRU File, to an internal table within the overall FRU

Byte	Type	Response Data
0	enum8	**CompletionCode** Possible values: { PLDM_BASE_CODES }
1		**Reserved** This field shall be set to 0.
2:3	uint16	**File Identifier** An opaque handle that is used to reference a FRU file in subsequent DfOpen commands
4:19	UUID	**FRU File FormatID** A 16-byte field that identifies the type and format of a FRU file
20:23	uint32	**FRU File Version** The version of FRU file, within a FormatID-specific format and release history
24:27	uint32	**FRU File Length** The length, in bytes, of the FRU file content
28:31	uint32	**FRU File Context** The definition of this field is FormatID-specific; for DMTF-specified FormatIDs, the value is 0.
32:33	bitfield16	**FRU File Flags** The definition of this field is FormatID-specific; for DMTF-specified FormatIDs, the value is 0.

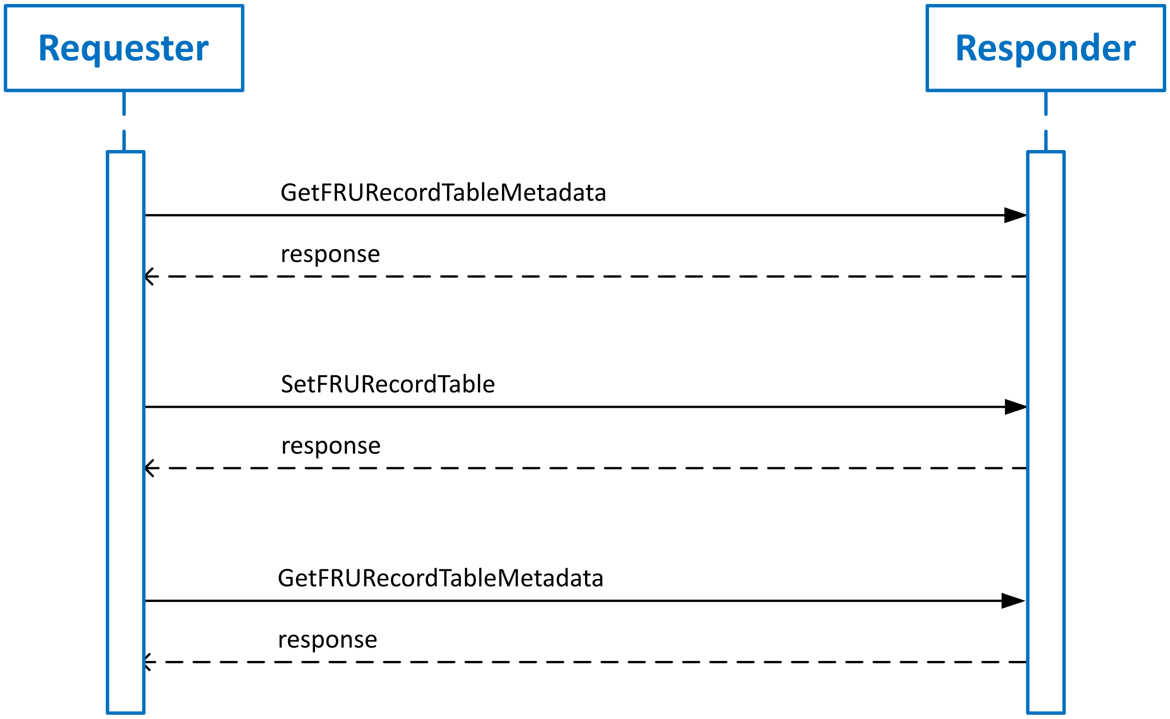
181
14 ANNEX A (informative) PLDM for FRU Data Examples

182 This annex provides examples of PLDM communications using the PLDM commands defined in this specification.

183
14.1 FRU Record Table Transfer between Endpoints Example

184 In this example, Requester sets the FRU Record Table on Responder. Requester first queries the FRU Record Table metadata by using the GetFRURecordTableMetadata command. The command response indicates that Responder does not have the latest FRU Record Table. Upon finding that Responder does not have the latest FRU Record Table, Requester transfers the FRU Record Table to Responder by using the SetFRURecordTable command. Requester can confirm the update was successful by sending a GetFRURecordTableMetadata command to Responder. Figure 1 shows the interaction.

185



186 **Figure 1 — Example of FRU Record Table Transfer Using the SetFRURecordTable Command**

187 14.2 GetFRURecordByOption Examples

188 The following examples show three **GetFRURecordByOption** commands and their respective responses. Table 14 describes a sample FRU Record Table that has two FRU record set identifiers that include two FRU fields with part number and serial number FRU field types. The second record set identifier also contains an OEM FRU Record.

189 **Table 14 — Sample FRU Record Table**

Field	Value
FRU Record Set Identifier #1	1030
FRU Record Type #1	1 = General FRU Record
Number of FRU fields	2
Encoding Type for FRU fields	1 = strASCII
FRU Field #1 Type	3 = Part Number
FRU Field #1 Length	6
FRU Field #1 Value	"123456"
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	"SN12345"
FRU Record Set Identifier #2	2040
FRU Record Type #2	1 = General FRU Record
Number of FRU fields	2
Encoding Type for FRU fields	1 = strASCII
FRU Field #1 Type	3 = Part Number
FRU Field #1 Length	6
FRU Field #1 Value	"345678"
FRU Field #2 Type	0x04 = Serial Number
FRU Field #2 Length	0x07
FRU Field #2 Value	"SN34567"
FRU Record Set Identifier #2	2040
FRU Record Type #3	254 = OEM FRU Record

Field	Value
Number of FRU fields	2
Encoding Type for FRU fields	1 = strASCII
FRU Field #1 Type	1 = VendorIANA
FRU Field #1 Length	4
FRU Field #1 Value	412
FRU Field #2 Type	2 = OEM
FRU Field #2 Length	6
FRU Field #2 Value	"Fusion"

EXAMPLE 1: This example returns all data for FRU record set identifier #1 = 1030.

In the **GetFRURecordByOption** command:

Record Set Identifier = 1030, **Record Type** = 0, and **Field Type** = 0

This results in the data shown in Table 15.

Table 15 — Get FRU Record Set Identifier Response Data (Example 1)

Field	Value
FRU Record Set Identifier #1	1030
FRU Record Type #1	1 = General FRU Record
Number of FRU fields	2
Encoding Type for FRU fields	1 = strASCII
FRU Field #1 Type	3 = Part Number
FRU Field #1 Length	6
FRU Field #1 Value	"123456"
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	"SN12345"

EXAMPLE 2: This example returns all FRU Record type 1 records (get all General FRU Records).

In the **GetFRURecordByOption** command:

Record Set Identifier = 0, **Record Type** = 1, and **Field Type** = 0

This results in the data shown in Table 16.

Table 16 — Get FRU Record Type Response Data (Example 2)

Field	Value
FRU Record Set Identifier #1	1030
FRU Record Type #1	1 = General FRU Record
Number of FRU fields	2
Encoding Type for FRU fields	1 = strASCII
FRU Field #1 Type	3 = Part Number
FRU Field #1 Length	6
FRU Field #1 Value	"123456"
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	"SN12345"
FRU Record Set Identifier #2	2040
FRU Record Type #2	1 = General FRU Record
Number of FRU fields	2
Encoding Type for FRU fields	1 = strASCII
FRU Field #1 Type	3 = Part Number
FRU Field #1 Length	6
FRU Field #1 Value	"345678"
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	"SN34567"

201 **EXAMPLE 3**: This example returns all FRU Record type / FRU field type = 4 fields (all General FRU Record serial number fields).

202 In the **GetFRURecordByOption** command:

203 **Record Set Identifier** = 0, **Record Type** = 1, and **Field Type** = 4

204 This results in the data shown in Table 17.

205 **Table 17 — Get FRU Field Type Response Data (Example 3)**

Field	Value
FRU Record Set Identifier #1	1030
FRU Record Type #1	1 = General FRU Record
Number of FRU fields	1
Encoding Type for FRU fields	1 = strASCII
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	"SN12345"
FRU Record Set Identifier #2	2040
FRU Record Type #2	1 = General FRU Record
Number of FRU fields	1
Encoding Type for FRU fields	1 = strASCII
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	"SN34567"

206 14.3 Accessing Nested FRU Record Sets via ReadFRUDataItem Example

207 In the following example, an add-in network interface card (NIC) exists. In addition, there exist 2 SFP modules, each of which is plugged into a NIC network port. The NIC and the 2 SFP modules all have FRU data records. The NIC controller chip acts as a PLDM terminus, but the 2 SFP modules are considered downstream FRUs, because their respective FRU data is accessed via the NIC PLDM terminus.

208 The NIC has a FRU serial number of "SN12345" and an associated FRU Record Set PDR which includes the following fields:

209 **Partial FRU Record Set PDR for NIC**

Field	Value
FRURecordSetIdentifier	1
EntityType	Physical, network controller
EntityInstanceNumber	1
EntityContainerID	100

210 For the **ReadFRUDataItem** command containing **RecordSetID** = 0, **FRUItemName** = 0x00010004 (Item Area = 1, Item Index = 4), the returned response contains **ItemValue** = "SN12345". Note that RecordSetID = 0 specifies the default record set, which in this case is the FRU data of the NIC network controller chip.

211 One of the SFP modules has a FRU serial number of "SN1357" and an associated FRU Record Set PDR which includes the following fields:

212 **Partial FRU Record Set PDR for SFP (Port 1)**

Field	Value
FRURecordSetIdentifier	2
EntityType	Physical, SFP
EntityInstanceNumber	1
EntityContainerID	200

213 For the **ReadFRUDataItem** command containing **RecordSetID** = 2, **FRUItemName** = 0x00010004 (Item Area = 1, Item Index = 4), the returned response contains **ItemValue** = "SN1357".

214 The other SFP module has a FRU serial number of "SN2468" and an associated FRU Record Set PDR which includes the following fields:

215 **Partial FRU Record Set PDR for SFP (Port 2)**

Field	Value
FRURecordSetIdentifier	3
EntityType	Physical, SFP
EntityInstanceNumber	2
EntityContainerID	200

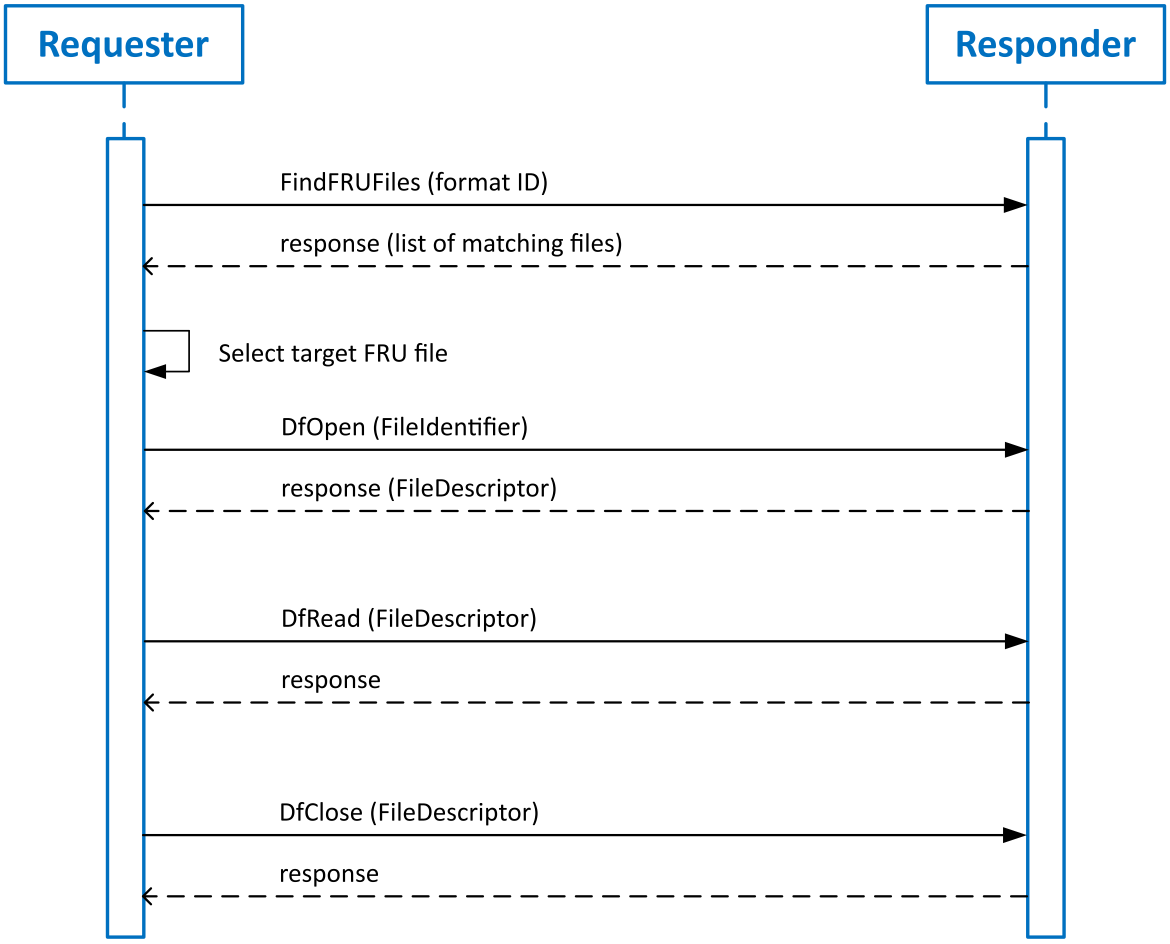
216 For the **ReadFRUDataItem** command containing **RecordSetID** = 3, **FRUItemName** = 0x00010004 (Item Area = 1, Item Index = 4), the returned response contains **ItemValue** = "SN2468".

217 14.4 Reading FRU Files Examples

218 14.4.1 Basic Example

219 In this example, Requester reads a FRU data file from Responder. Requester first queries Responder for a list of files matching some search criteria specified via the FormatID parameter of the FindFRUFiles command. The response contains a list of such files. Next, Requester must select a target FRU file to read from the list supplied by Responder. The target FRU file's FRU File Identifier is contained within the response to the FindFRUFiles command. Requester then reads the selected file by performing the sequence of DfOpen, DfRead, and DfClose, as specified in DSP0242. Figure 2 shows the interaction.

220

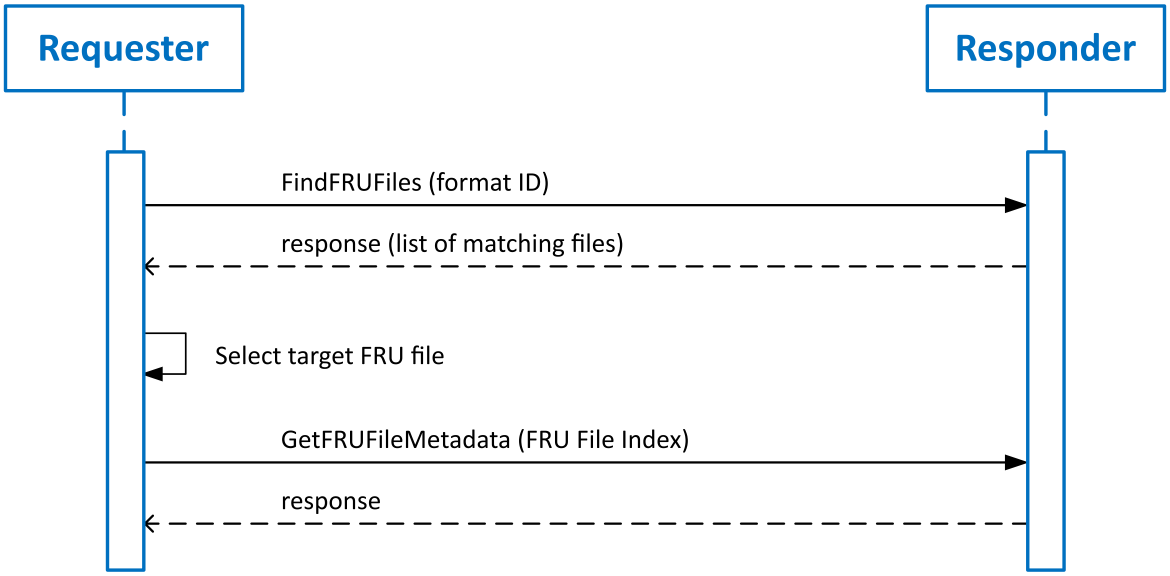


221 **Figure 2 — Example of basic FRU file read access**

222 14.4.2 Query File Metadata Example

223 In this example, Requester reads a FRU data file's metadata. Requester first queries Responder for a list of files matching some search criteria specified via the FormatID parameter of the FindFRUFiles command. The response contains a list of such files. Next, Requester must select a target FRU file from the list supplied by Responder. The target FRU file's FRU File Index is contained within the response to the FindFRUFiles command. Requester then reads the target File's metadata by sending a GetFRUFileMetadata, with the FRU File Index as a parameter of the request. Figure 3 shows the interaction.

224



225 **Figure 3 — Example of querying FRU file metadata**

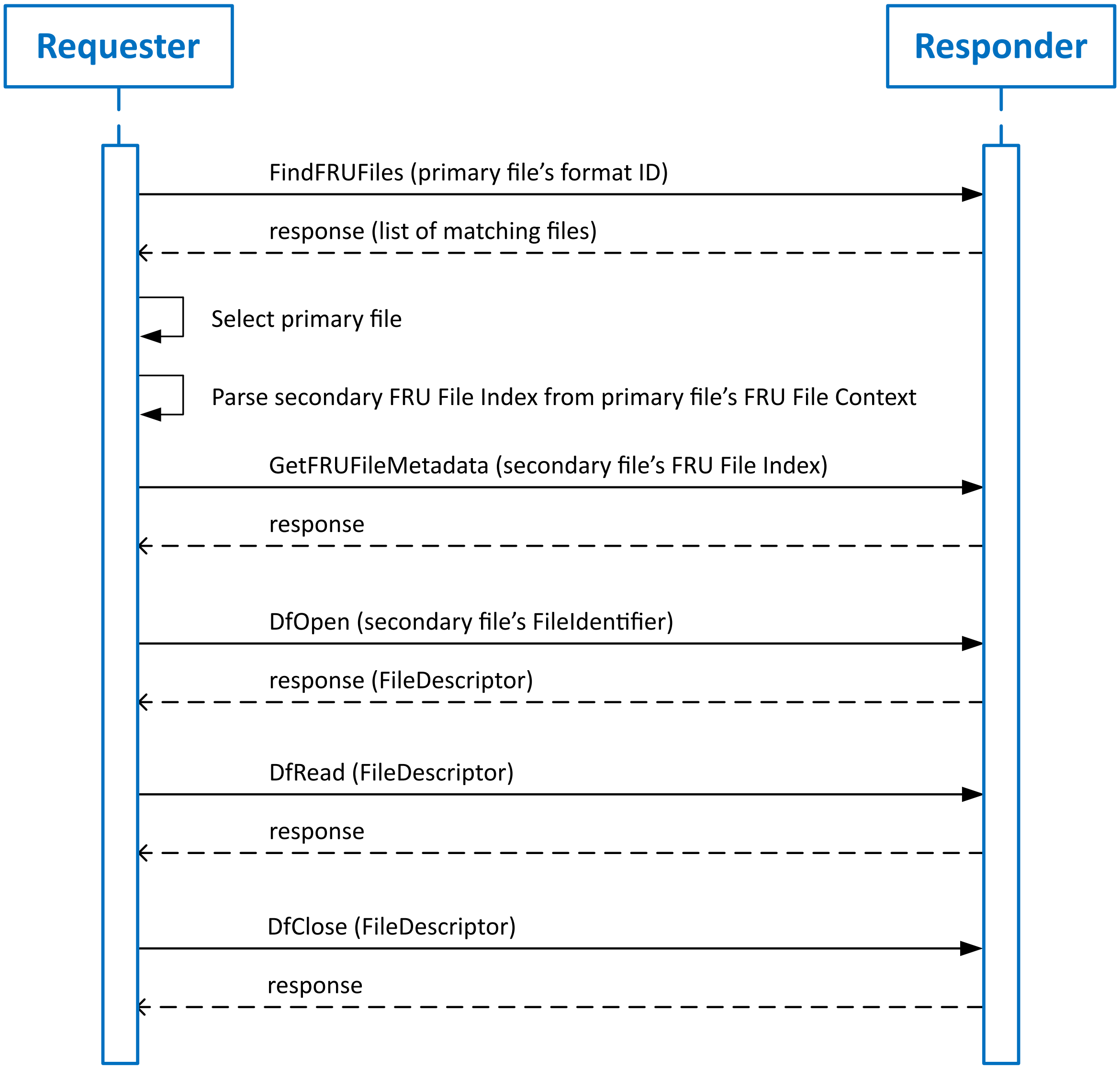
226 14.4.3 Follow a Reference to Read a Linked File Example

227 This example combines elements of the previous two examples. Assume for the purposes of this example that some FRU data file (the "primary" FRU data file) includes a link to a secondary FRU data file. More specifically, this link is encoded as the secondary file's FRU File Index, contained within the primary file's FRU File Context information. FRU File Context information is contained within the response to the FindFRUFiles command.

228 Requester first queries Responder for a list of files matching some search criteria specified via the FormatID parameter of the FindFRUFiles command. The response contains a list of such files. Next, Requester must select a primary FRU file from the list supplied by Responder. The primary FRU file's FRU File Context is contained within the response to the FindFRUFiles command. Requester then parses the secondary file's FRU File Index from the primary file's FRU File Context. Requester then reads the secondary file's metadata by sending a GetFRUFileMetadata, with secondary file's FRU File Index as a parameter of the request. The secondary file's FRU

File Identifier is included in the GetFRUFileMetadata response message. Requester then reads the secondary file by performing the sequence of DfOpen, DfRead, and DfClose, as specified in DSP0242. Figure 4 shows the interaction.

229


Requester
Responder
FindFRUFiles (primary file's format ID)
response (list of matching files)
Select primary file
Parse secondary FRU File Index from primary file's FRU File Context
GetFRUFileMetadata (secondary file's FRU File Index)
response
DfOpen (secondary file's FileIdentifier)
response (FileDescriptor)
DfRead (FileDescriptor)
response
DfClose (FileDescriptor)
response

230 **Figure 4 — Example of reading a linked file**

231

15 ANNEX B (informative) Notation

232 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, 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 indicates the entire range may be present or absent. The lowest offset is on the left, the highest is on the right.
DSP0240	Underlined, blue text is typically used to indicate a reference to a document or specification called out 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, the least significant bit is on the right.
1b	The lower case "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.

233

16 ANNEX C (informative) Change Log

Version	Date	Description
1.0.0	2011-10-26	DMTF Standard
1.0.1	2022-01-01	Fixed footer (Mantis 2977). Added Annex A. [Note: in version 2.0.0, this Annex A was renamed to Annex B]
2.0.0	2025-04-24	Major update to PLDM for FRU data transfer: version 2.0 differs significantly from the 1.0 spec and is not backward-compatible. Added support for FRU Data Items and FRU Files.