



1
2
3
4

Document Number: DSP0257

Date: 2011-10-26

Version: 1.0.0

5
6

Platform Level Data Model (PLDM) for FRU Data Specification

7
8
9
10

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

11 Copyright notice

12 Copyright © 2011 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

30 For information about patents held by third-parties which have notified the DMTF that, in their opinion,
31 such patent may relate to or impact implementations of DMTF standards, visit
32 <http://www.dmtf.org/about/policies/disclosures.php>.

33 PCI-SIG, PCIe, and the PCI HOT PLUG design mark are registered trademarks or service marks of PCI-
34 SIG.

35 All other marks and brands are the property of their respective owners.

36

37

CONTENTS

38	Foreword	5
39	Introduction	6
40	1 Scope	7
41	2 Normative references	7
42	3 Terms and definitions	7
43	4 Symbols and abbreviated terms.....	8
44	5 Conventions	9
45	6 PLDM for FRU Data version	9
46	7 PLDM for FRU record data format	9
47	7.1 FRU Record Set Identifier.....	9
48	7.2 FRU Record Type.....	9
49	7.3 Number of FRU Fields	9
50	7.4 FRU Encoding Types.....	9
51	7.5 FRU Field Type, Length and Value	10
52	8 FRU Record Set PDR	12
53	9 PLDM for FRU Data Transfer.....	12
54	9.1 PLDM Representation of FRU Record Data.....	12
55	9.2 PLDM Commands for FRU Data Transfer.....	14
56	10 PLDM for FRU Data Transfer Examples.....	17
57	10.1 Multipart Transfers	17
58	10.2 FRU Record Table Transfer between Endpoints Example.....	18
59	10.3 GetFRURecordByOption Examples	19
60	ANNEX A (informative) Change Log	23
61		

62 Tables

63	Table 1 – PLDM FRU Data Types	9
64	Table 2 – PLDM FRU Record Data Format.....	10
65	Table 3 – PLDM FRU Record Data Table Format.....	11
66	Table 4 – FRU Record Type Definitions	11
67	Table 5 – General FRU Record Field Type Definitions	11
68	Table 6 – OEM FRU Record Field Type Definitions	12
69	Table 7 – PLDM Representation of FRU Record Data.....	13
70	Table 8 – PLDM for FRU Data Transfer Command Codes	14
71	Table 9 – GetFRURecordMetadata Command Format.....	14
72	Table 10 – GetFRURecordTable Command Format.....	15
73	Table 11 – SetFRURecordTable Command Format	16
74	Table 12 – GetFRURecordByOption Command Format	17
75	Table 13 – Sample FRU Record Table.....	19
76	Table 14 – Get FRU Record Set Identifier Response Data (Example 1)	20
77	Table 15 – Get FRU Record Type Response Data (Example 2).....	21
78	Table 16 – Get FRU Field Type Response Data (Example 3).....	22
79		

80 **Figures**

81 Figure 1 – Multipart FRU Record Table Transfer Using the GetFRURecordTable Command 18

82 Figure 2 – Example of FRU Record Table Transfer Using the SetFRURecordTable Command..... 19

83

84

Foreword

85 The *Platform Level Data Model (PLDM) for FRU Data Specification* (DSP0257) was prepared by the
86 Platform Management Components Intercommunications Working Group of the DMTF.

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

89 **Acknowledgments**

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

- 91 • Alan Berenbaum – SMSC
- 92 • Phil Chidester – Dell (Editor)
- 93 • Hoan Do – Broadcom Corporation
- 94 • Ed Klodnicki – IBM
- 95 • John Leung – Intel
- 96 • Hemal Shah – Broadcom Corporation
- 97 • Tom Slaight – Intel

98

Introduction

99 The *Platform Level Data Model (PLDM) FRU Data Specification* defines messages, data structures, and
100 data types used for FRU (Field Replaceable Unit) Data access and representation. FRU Data typically
101 includes the serial number, part number and manufacturer for a field replaceable unit.

102 Platform Level Data Model (PLDM) for FRU Data Specification

103 1 Scope

104 DSP0257, *Platform Level Data Model for FRU Data Specification*, defines a FRU data format that
105 provides platform asset information including part number, serial number and manufacturer. The FRU
106 Record Table typically resides in a non-volatile memory accessible by the management controller and
107 contains one or more FRU records. This document describes Platform Level Data Model (PLDM) data
108 structures and commands for transferring FRU data between the components of a platform management
109 subsystem.

110 This document meets the following objectives:

- 111 • Specifies PLDM representations of FRU Record Table and FRU record data format
- 112 • Specifies a set of commands for transferring FRU record data information

113 2 Normative references

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

118 ANSI/IEEE Standard 754-1985, *Standard for Binary Floating Point Arithmetic*

119 DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification 1.0*,
120 http://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.0.pdf

121 DMTF DSP0245, *Platform Level Data Model (PLDM) IDs and Codes Specification 1.0*,
122 http://www.dmtf.org/standards/published_documents/DSP0245_1.0.pdf

123 DMTF DSP0248, *Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification*
124 *1.0*, http://www.dmtf.org/sites/default/files/standards/documents/DSP0248_1.0.pdf

125 IETF RFC2781, *UTF-16, an encoding of ISO 10646*, February 2000, <http://www.ietf.org/rfc/rfc2781.txt>

126 IETF RFC3629, *UTF-8, a transformation format of ISO 10646*, November 2003,
127 <http://www.ietf.org/rfc/rfc3629.txt>

128 IETF RFC4122, *A Universally Unique Identifier (UUID) URN Namespace*, July 2005,
129 <http://www.ietf.org/rfc/rfc4122.txt>

130 IETF RFC4646, *Tags for Identifying Languages*, September 2006, <http://www.ietf.org/rfc/rfc4646.txt>

131 ISO 8859-1, *Final Text of DIS 8859-1, 8-bit single-byte coded graphic character sets -- Part 1: Latin*
132 *alphabet No. 1*, February 1998

133 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
134 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>

135 3 Terms and definitions

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

138 The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"),
139 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
140 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
141 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
142 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
143 alternatives shall be interpreted in their normal English meaning.

144 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as
145 described in [ISO/IEC Directives, Part 2](#), Clause 5.

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

149 Refer to [DSP0240](#) for terms and definitions that are used across the PLDM specifications. For the
150 purposes of this document, the following additional terms and definitions apply.

151 3.1

152 Platform Descriptor Record

153 PDR

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

156 PDRs are mostly used to support PLDM monitoring and control and platform events. This information also
157 describes the relationships (associations) between sensor and control functions, the physical or logical
158 entities that are being monitored or controlled, and the semantic information associated with those
159 elements.

160 4 Symbols and abbreviated terms

161 Refer to [DSP0240](#) for symbols and abbreviated terms that are used across the PLDM specifications. For
162 the purposes of this document, the following additional symbols and abbreviated terms apply.

163 4.1

164 CIM

165 Common Information Model

166 4.2

167 EID

168 Endpoint ID

169 4.3

170 FRU

171 Field Replaceable Unit

172 4.4

173 IANA

174 Internet Assigned Numbers Authority

175 5 Conventions

176 Refer to [DSP0240](#) for conventions, notations, and data types that are used across the PLDM
177 specifications. The data types listed in Table 1 are also defined for use in this specification.

178 **Table 1 – PLDM FRU Data Types**

Data Type	Interpretation
ASCII	Characters are encoded using the 8-bit ISO8859-1 "ASCII + Latin1" character set encoding. ASCII strings are limited to a maximum of 255 bytes.
UTF-8	UTF-8 encoded string per RFC3629. UTF-8 defines a variable length for Unicode encoded characters where each individual character may require one to four bytes. UTF-8 encoded unicode strings are limited to a maximum of 255 bytes.
UTF-16	UTF-16 encoded string with Byte Order Mark (BOM) per RFC2781. UTF-16 defines an encoding for Unicode characters where each individual character requires two bytes. UTF-16 encoded unicode strings are limited to a maximum of 255 bytes.
UTF-16LE	UTF-16, "little endian" encoded string per RFC2781. UTF-16LE defines an encoding for Unicode characters where each individual character requires two bytes. UTF16LE encoded unicode strings are limited to a maximum of 255 bytes.
UTF-16BE	UTF-16, "big-endian" encoded string per RFC2781. UTF-16BE defines an encoding for Unicode characters where each individual character requires two bytes. UTF16BE encoded unicode strings are limited to a maximum of 255 bytes.

179 6 PLDM for FRU Data version

180 The version of this *Platform Level Data Model (PLDM) for FRU Data Specification* shall be 1.0.0 (major
181 version number 1, minor version number 0, update version number 0, and no alpha version).

182 For the GetPLDMVersion command described in [DSP0240](#), the version of this specification is reported
183 using the encoding as 0xF1F0F000.

184

185 7 PLDM for FRU record data format

186 All PLDM FRU record data is represented by the fields in the following subclauses.

187 7.1 FRU Record Set Identifier

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

189 7.2 FRU Record Type

190 The FRU Record Type identifies the FRU record and is defined in Table 4.

191 7.3 Number of FRU Fields

192 The Number of FRU fields indicated the number of fields that are included in a FRU record.

193 7.4 FRU Encoding Types

194 String values types for a specific FRU Record are defined in the Encoding Type field of the FRU. The
195 Encoding Type shall apply for all FRU fields in a FRU Record with a sting format. The FRU Encoding
196 Types are defined in Table 2.

197 All strings shall be preceded by a length variable where a length of zero indicates that the field is not used
 198 in this specific FRU. String lengths shall be in bytes. Strings are not null terminated and are limited to a
 199 255-byte size. FRU record set identifiers and their associated record types shall be contiguous in the
 200 table.

201 7.5 FRU Field Type, Length and Value

202 All FRU fields are defined by a Type, Length and Value (TLV). The Type is defined in Table 5, which also
 203 defines the Field format and length ranges. The Field Value is defined by the manufacturer.

204 Table 2 specifies the format used for the PLDM FRU Data Format.

205

Table 2 – 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 = ASCII 2 = UTF8 3 = UTF16 4 = UTF16-LE 5 = UTF16-BE 6-255 = reserved
1 byte	uint8	FRU Field Type #1
1 byte	uint8	FRU Field Length #1
Up to 255 bytes (see Table 5)	Determined by FRU Field Type (see Table 5)	FRU Field #1 Value
1 byte	uint8	FRU Field #2 Type
1 byte	uint8	FRU Field #2 Length
Up to 255 bytes (see Table 5)	Determined by FRU Field Type / Length (see Table 5)	FRU Field #2 Value
....
1 byte	uint8	FRU Field #n Type
1 byte	uint8	FRU Field #n Length
Up to 255 bytes (see Table 5)	Determined by FRU Field Type / Length (see Table 5)	FRU Field #n Value

206

207 Table 3 specifies the format used for the PLDM FRU Record Table Format.

208 **Table 3 – PLDM FRU Record Data Table Format**

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

209 Table 4 defines the FRU Record Types.

210 **Table 4 – FRU Record Type Definitions**

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

211 Table 5 defines the General FRU record field type definitions.

212 **Table 5 – 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	

Field Type Number	Field Type Description	Field Format	Length
15	Vendor IANA	uint32	4 bytes
16 – 255	Reserved	N/A	

213 Table 6 defines the OEM FRU Record field type definitions.

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

216

Table 6 – 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

217 8 FRU Record Set PDR

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

221 The FRU Record Set PDR is defined in [DSP0248](#).

222 9 PLDM for FRU Data Transfer

223 This clause defines the data representations and PLDM commands for FRU data transfer.

224 9.1 PLDM Representation of FRU Record Data

225 In the PLDM messages for FRU data transfers, the FRU Record Data representation is as shown in Table
226 7.

Table 7 – PLDM Representation of FRU Record Data

Byte	Type	Field
Variable	–	<p>FRU Record Data (one or more)</p> <p>See Table 2 for the PLDM representation of PLDM FRU Record Data.</p>
Variable	uint8[]	<p>Pad</p> <p>0 to 3 number of pad bytes. The value stored in each pad byte is 0x00.</p> <p>The transmitter can compute the number of pad bytes from the FRU Data by using the following algorithm:</p> <p>Let L be the total number of bytes in the FRU Record Data excluding the pad and the integrity checksum.</p> <p>if (L modulo 4 = 0) then NumPadBytes = 0; else NumPadBytes = 4 – L modulo 4;</p> <p>The receiver can compute the number of pad bytes from the FRU Record Data by using the following algorithm. In the algorithm, the receiver parses FRU Record Data until the remaining bytes are less than 8. When it reaches that stage, the remaining bytes contain the pad bytes and four bytes of data integrity checksum.</p> <p>Let L be the total number of bytes in the FRU Record Data including the pad and the integrity checksum.</p> <pre> RemBytes = L; i = 0; while (RemBytes >= 8) { Process the ith FRU Record Data in the FRU Record Table; RemBytes = RemBytes - 4 – Total length of ith FRU Record Data including the formatted and unformatted areas; i = i+1; } NumPadBytes = RemBytes modulo 4; </pre>
	uint32	<p>FRUDataStructureIntegrityChecksum</p> <p>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.</p> <p>For this specification, the CRC-32 algorithm with the polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (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.</p>

228 **9.2 PLDM Commands for FRU Data Transfer**229 **9.2.1 Overview**

230 Table 8 defines the PLDM command codes defined in the following subclause for the PLDM for FRU data
 231 transfer. The PLDM FRU messages have their own PLDM message type, which is defined in [DSP0245](#).

232 **Table 8 – PLDM for FRU Data Transfer Command Codes**

Command	Code Value	Requirement	Section
GetFRURecordTableMetadata	0x01	Mandatory	See 9.2.2.
GetFRURecordTable	0x02	Mandatory	See 9.2.3.
SetFRURecordTable	0x03	Conditional	See 9.2.4.
GetFRURecordByOption	0x04	Optional	See 9.2.5.

233 The requirements specified in Table 4 are relative to the services provided by the PLDM terminus.

234 **9.2.2 Get FRURecordTableMetadata**

235 The GetFRURecordTableMetadata command, described in Table 9, is used to get the FRU Record Table
 236 metadata information that includes the FRU Record major version, the FRU Record minor version, the
 237 size of the largest FRU Record data, total length of the FRU Record Table, total number of FRU Record
 238 Data structures, and the integrity checksum on the FRU Record Table data.

239 **Table 9 – GetFRURecordTableMetadata 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 }
1	uint8	FRUDATAMajorVersion The major version of the FRU DATA specification with which the FRU Record Table. For an implementation compliant with this specification, the FRUDATAMajorVersion shall be set to 0x01.
2	uint8	FRUDATAMinorVersion The minor version of the FRU DATA specification with which the FRU Record Table. For an implementation compliant with this specification, the FRUDATAMinorVersion shall be set to 0x00.
3:6	uint32	FRURecordTableMaximumSize 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.

Byte	Type	Request Data
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	FRU DATAstructureTableIntegrityChecksum (CRC-32) Integrity checksum shall be computed on the FRU Record Table data as shown in Table 7 excluding pad bytes. See Table 7 for more information about this integrity checksum.

240 **9.2.3 GetFRURecordTable**

241 The GetFRURecordTable command, described in Table 10, is used to get the FRU Record Table data.
 242 This command is defined to allow the FRU Record Table data to be transferred using a sequence of one
 243 or more command/response messages. When more than one command is used to transfer the FRU
 244 Record Table, the response messages contain the non-overlapping contiguous portions of FRU Record
 245 Table as defined in Table 7. By combining the portions of FRU Record Table from the response
 246 messages, the entire FRU Record Table can be reconstructed.

247 **Table 10 – GetFRURecordTable Command Format**

Byte	Type	Request Data
0:3	uint32	DataTransferHandle A handle that is used to identify an FRU Record Table data transfer. This handle is ignored by the responder when the TransferOperationFlag is set to GetFirstPart.
4	enum8	TransferOperationFlag The operation flag that indicates whether this is the start of the transfer Possible values: {GetNextPart=0x00, GetFirstPart=0x01}
Byte	Type	Response Data
0	enum8	CompletionCode Possible values: { PLDM_BASE_CODES, INVALID_DATA_TRANSFER_HANDLE=0x80, INVALID_TRANSFER_OPERATION_FLAG=0x81, FRU_DATA_STRUCTURE_TABLE_UNAVAILABLE=0x85 }
1:4	uint32	NextDataTransferHandle A handle that is used to identify the next portion of the transfer
5	enum8	TransferFlag The transfer flag that indicates what part of the transfer this response represents Possible values: {Start=0x01, Middle=0x02, End=0x04, StartAndEnd = 0x05}
Variable	–	Portion of FRU Record Table This data is a portion of the overall FRU Record Table format shown in Table 3. The portion that is returned is determined by the combination of the DataTransferHandle and TransferOperationFlag fields passed in the request.

248 **9.2.4 SetFRURecordTable**

249 The SetFRURecordTable command, described in Table 11, is used to write the FRU Record Table. This
 250 command is defined to allow the FRU Record Table to be transferred using a sequence of one or more
 251 command/response messages. When more than one command is used to transfer the FRU Record
 252 Table, the request messages contain the non-overlapping contiguous portions of FRU Record Table as
 253 defined in Table 7. By combining the portions of FRU record table from the request messages, the entire
 254 FRU Record Table can be reconstructed.

255 **Table 11 – SetFRURecordTable Command Format**

Byte	Type	Request Data
0:3	uint32	DataTransferHandle A handle that is used to identify FRU Record Table transfer. This handle is ignored by the responder when the TransferFlag is set to Start or StartAndEnd.
4	enum8	TransferFlag The transfer flag that indicates what part of the transfer this request represents Possible values: {Start=0x01, Middle=0x02, End=0x04, StartAndEnd = 0x05}
Variable	–	Portion of FRU Record Table See Table 7 for the format.
Byte	Type	Response Data
0	enum8	CompletionCode Possible values: { PLDM_BASE_CODES, INVALID_DATA_TRANSFER_HANDLE=0x80, INVALID_TRANSFER_FLAG=0x82, INVALID_DATA_INTEGRITY_CHECK=0x84 }
1:4	uint32	NextDataTransferHandle A handle that is used to identify the next portion of the transfer

256 **9.2.5 GetFRURecordByOption**

257 The GetFRURecordByOption command, described in Table 12, is used to get the FRU records by record
 258 handle and length. This command is defined to allow the FRU Record Table to be transferred using a
 259 sequence of one or more command/response messages. When more than one command is used to
 260 transfer the FRU Record Table, the response messages contain the non-overlapping contiguous portions
 261 of FRU Record Data as defined in Table 7. By combining the portions of FRU Record Data from the
 262 response messages, the entire FRU Record Table can be reconstructed.

263

Table 12 – GetFRURecordByOption Command Format

Byte	Type	Request Data
0:3	uint32	DataTransferHandle A handle that is used to identify FRU Record Data transfer. This handle is ignored by the responder when the TransferOperationFlag is set to GetFirstPart.
4:5	uint16	FRUTableHandle A handle that is used to identify FRU DATA records.
6:7	uint16	Record Set Identifier Possible values: {All record sets=0x0000, Specific record set=0x0001 – 0xffff}
8	uint8	Record Type Possible values: {All record types=0x00, Specific record types=0x01 – 0xff}
9	uint8	Field Type Possible values: {All record field types=0x00, Specific field types=0x01 – 0xff} If field type is non-zero, the record type shall also be non-zero.
10	enum8	TransferOperationFlag The operation flag that indicates whether this is the start of the transfer Possible values: {GetNextPart=0x00, GetFirstPart=0x01}
Byte	Type	Response Data
0	enum8	CompletionCode Possible values: { PLDM_BASE_CODES, INVALID_DATA_TRANSFER_HANDLE=0x80, INVALID_TRANSFER_OPERATION_FLAG=0x81, FRU_DATA_STRUCTURE_TABLE_UNAVAILABLE=0x85 }
1:4	uint32	NextDataTransferHandle A handle that is used to identify the next portion of the transfer
5	enum8	TransferFlag The transfer flag that indicates what part of the transfer this response represents Possible values: {Start=0x01, Middle=0x02, End=0x04, StartAndEnd = 0x05}
Variable	–	FRU DATAStructureData See Table 7 for the format.

264 **10 PLDM for FRU Data Transfer Examples**

265 This clause provides examples of PLDM communications using the PLDM commands defined in this
266 specification.

267 **10.1 Multipart Transfers**

268 The commands defined in clause 9 for transferring FRU Record Table data support multipart transfers.
269 The Get* and Set* commands use flags and data transfer handles to perform multipart transfers. For a
270 data transfer for initiating a data transfer (or getting the first part of data) using a Get* command, the
271 TransferOperationFlag shall be set to GetFirstPart in the request of the Get* command.

- 272 • For transferring a part other than the first part of data by using a Get* command, the
273 TransferOperationFlag shall be set to GetNextPart and the DataTransferHandle shall be set to
274 the NextDataTransferHandle that was obtained in the response of the previous Get* command
275 for this data transfer.
 - 276 • The TransferFlag specified in the request of a Set* command or the response of a Get*
277 command has the following meanings:
 - 278 – Start, which is the first part of the data transfer
 - 279 – Middle, which is neither the first nor the last part of the data transfer
 - 280 – End, which is the last part of the data transfer
 - 281 – StartAndEnd, which is the first and the last part of the data transfer
 - 282 • The requester shall consider a data transfer complete and ignore the NextDataTransferHandle
283 when the TransferFlag in the response of a Get* command is set to End or StartAndEnd.
 - 284 • The responder shall consider a data transfer complete when the TransferFlag in the request of
285 a Set* command is set to End or StartAndEnd.
- 286

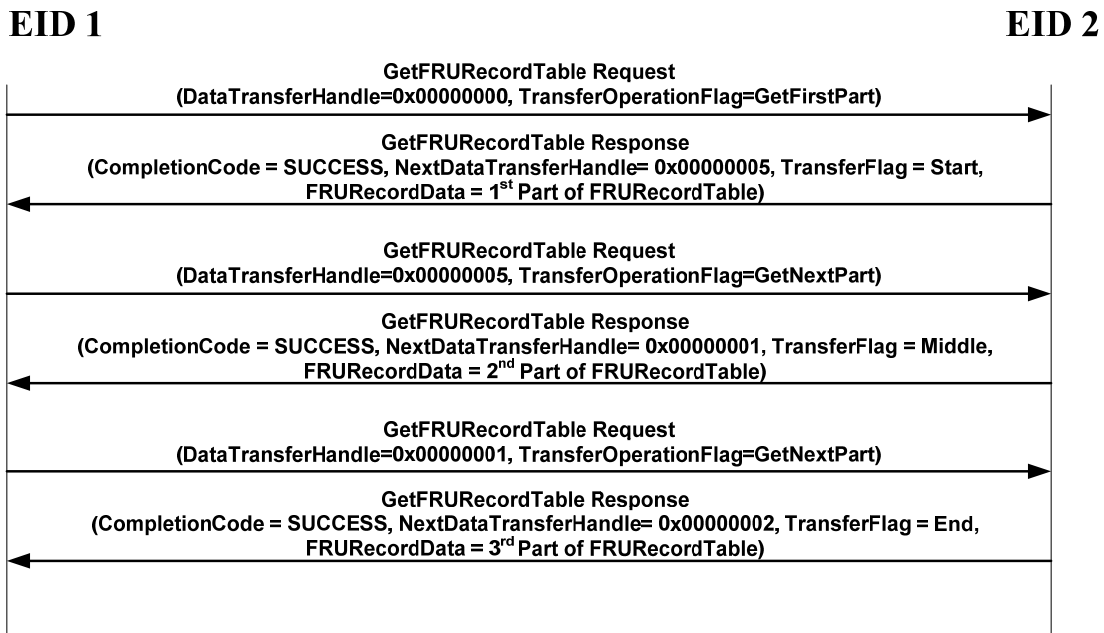


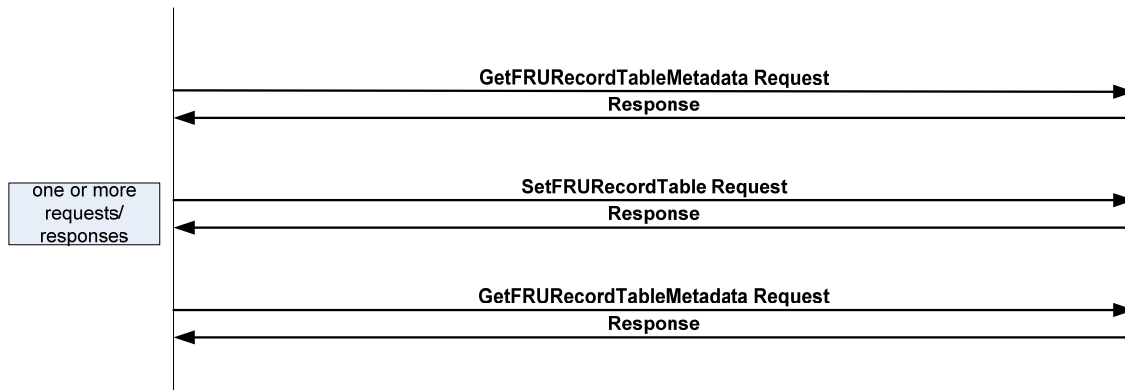
Figure 1 – Multipart FRU Record Table Transfer Using the GetFRURecordTable Command

10.2 FRU Record Table Transfer between Endpoints Example

In this example, the EID 1 sets the FRU Record Table on the EID 2. EID1 first queries the FRU Record Table metadata by using the GetFRURecordTableMetadata command. The response from the EID 2 to this command indicates that the EID 2 does not have the latest FRU Record Table. Upon finding that the EID 2 does not have the latest FRU Record Table, EID 1 transfers the FRU Record Table to EID 2 by using the SetFRURecordTable command. After transferring the latest FRU Record Table, EID 1 reads the FRU Record Table metadata on the EID 2 by using the GetFRURecordTableMetadata command to confirm that the FRU records were correctly set. This example can be used in a push model where EID 2 is maintaining a copy of the FRU Record Table provided by EID 1 and EID 1 pushes to EID 2 a copy of the FRU Record Table by using SetFRURecordTable command. Figure 2 shows the data transfer.

EID 1

EID 2



299

300 **Figure 2 – Example of FRU Record Table Transfer Using the SetFRURecordTable Command**

301 **10.3 GetFRURecordByOption Examples**

302 The following examples show three **GetFRURecordByOption** commands and their respective
 303 responses. Table 13 describes a sample FRU Record Table that has two FRU record set identifiers that
 304 include two FRU fields with part number and serial number FRU field types. The second record set
 305 identifier also contains an OEM FRU Record.

306

Table 13 – 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 = ASCII
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 = ASCII
FRU Field #1 Type	3 = Part Number
FRU Field #1 Length	6

Field	Value
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
Number of FRU fields	2
Encoding Type for FRU fields	1 = ASCII
FRU Field #1 Type	1 = vendorIANA
FRU Field #1 Length	4
FRU Field #1 Value	3704
FRU Field #2 Type	2 = OEM
FRU Field #2 Length	6
FRU Field #2 Value	“Fusion”

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

308 In the **GetFRURecordByOption** command:

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

310 This results in the data shown in Table 14.

311 **Table 14 – 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 = ASCII
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”

312

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

314 In the **GetFRURecordByOption** command:

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

316 This results in the data shown in Table 15.

317 **Table 15 – 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 = ASCII
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 = ASCII
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”

318

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

321 In the **GetFRURecordByOption** command:

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

323 This results in the data shown in Table 16.

324 **Table 16 – 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 = ASCII
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 = ASCII
FRU Field #2 Type	4 = Serial Number
FRU Field #2 Length	7
FRU Field #2 Value	“SN34567”

325

326
327
328
329
330

ANNEX A (informative)

Change Log

Version	Date	Description
1.0.0	2011-10-26	DMTF Standard

331