

# INTERNATIONAL STANDARD

**Industrial communication networks – Fieldbus specifications –  
Part 6-2: Application layer protocol specification – Type 2 elements**

IEC 61158-6-2:2023

<https://standards.iteh.ai/catalog/standards/sist/e854ec23-e379-42d4-9cfa-b8840b623c8e/iec-61158-6-2-2023>



**THIS PUBLICATION IS COPYRIGHT PROTECTED**  
**Copyright © 2023 IEC, Geneva, Switzerland**

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Secretariat  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland

Tel.: +41 22 919 02 11  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)

**About the IEC**

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

**About IEC publications**

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

**IEC publications search - [webstore.iec.ch/advsearchform](http://webstore.iec.ch/advsearchform)**

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, ...). It also gives information on projects, replaced and withdrawn publications.

**IEC Products & Services Portal - [products.iec.ch](http://products.iec.ch)**

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

**IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)**

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

**Electropedia - [www.electropedia.org](http://www.electropedia.org)**

The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

**IEC Customer Service Centre - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)**

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: [sales@iec.ch](mailto:sales@iec.ch).

[IEC 61158-6-2:2023](https://standards.iteh.ai/catalog/standards/sist/e854cc25-e379-42d4-9cfa-b8840b623c8e/iec-61158-6-2-2023)

---

<https://standards.iteh.ai/catalog/standards/sist/e854cc25-e379-42d4-9cfa-b8840b623c8e/iec-61158-6-2-2023>



IEC 61158-6-2

Edition 5.0 2023-03

# INTERNATIONAL STANDARD

---

**Industrial communication networks – Fieldbus specifications –  
Part 6-2: Application layer protocol specification – Type 2 elements**

[IEC 61158-6-2:2023](https://standards.iteh.ai/catalog/standards/sist/e854ec23-e379-42d4-9cfa-b8840b623c8e/iec-61158-6-2-2023)

<https://standards.iteh.ai/catalog/standards/sist/e854ec23-e379-42d4-9cfa-b8840b623c8e/iec-61158-6-2-2023>

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

---

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-6631-1

**Warning! Make sure that you obtained this publication from an authorized distributor.**

## CONTENTS

|  |     |
|--|-----|
| FOREWORD.....  | 14  |
| INTRODUCTION.....  | 17  |
| 1 Scope.....   | 18  |
| 1.1 General.....   | 18  |
| 1.2 Specifications .....   | 18  |
| 1.3 Conformance .....  | 19  |
| 2 Normative references .....   | 19  |
| 3 Terms, definitions, symbols, abbreviated terms and conventions ..... | 21  |
| 3.1 Terms and definitions from other ISO/IEC standards.....            | 21  |
| 3.1.1 Terms and definitions from ISO/IEC 7498-1 .....                  | 21  |
| 3.1.2 Terms and definitions from ISO/IEC 9545 .....                    | 21  |
| 3.1.3 Terms and definitions from ISO/IEC 8824-1 .....                  | 22  |
| 3.1.4 Terms and definitions from ISO/IEC 8825-1 .....                  | 22  |
| 3.2 Terms and definitions from IEC 61158-5-2 .....                     | 22  |
| 3.3 Additional terms and definitions .....                             | 22  |
| 3.4 Abbreviated terms and symbols .....                                | 30  |
| 3.5 Conventions.....   | 30  |
| 3.5.1 General concept .....  | 30  |
| 3.5.2 Attribute specification .....                                    | 31  |
| 3.5.3 Common services .....  | 31  |
| 3.5.4 State machine conventions .....                                  | 34  |
| 4 Abstract syntax.....   | 36  |
| 4.1 FAL PDU abstract syntax .....                                      | 36  |
| 4.1.1 General .....  | 36  |
| 4.1.2 PDU structure.....   | 36  |
| 4.1.3 UCMM_PDUs .....  | 39  |
| 4.1.4 Transport_Headers.....   | 40  |
| 4.1.5 CM_PDUs.....   | 43  |
| 4.1.6 CM PDU components .....  | 56  |
| 4.1.7 MR headers .....   | 65  |
| 4.1.8 OM_Service_PDU.....  | 66  |
| 4.1.9 Message and connection paths.....                                | 130 |
| 4.1.10 Class, attribute and service codes .....                        | 146 |
| 4.1.11 Error codes.....  | 154 |
| 4.2 Data abstract syntax specification.....                            | 169 |
| 4.2.1 Transport format specification.....                              | 169 |
| 4.2.2 Abstract syntax notation .....                                   | 170 |
| 4.2.3 Control network data specification .....                         | 170 |
| 4.2.4 Data type specification / dictionaries.....                      | 172 |
| 4.3 Encapsulation abstract syntax.....                                 | 174 |
| 4.3.1 Encapsulation protocol .....                                     | 174 |
| 4.3.2 Encapsulation messages .....                                     | 175 |
| 4.3.3 Command descriptions .....                                       | 178 |
| 4.3.4 Common packet format.....  | 190 |
| 5 Transfer syntax.....   | 194 |
| 5.1 Compact encoding .....   | 194 |

|        |  |     |
|--------|--|-----|
| 5.1.1  | Encoding rules.....  | 194 |
| 5.1.2  | Encoding constraints .....                                 | 195 |
| 5.1.3  | Examples.....  | 195 |
| 5.2    | Data type reporting .....                                  | 202 |
| 5.2.1  | Object data representation.....                            | 202 |
| 5.2.2  | Elementary data type reporting .....                       | 202 |
| 5.2.3  | Constructed data type reporting.....                       | 203 |
| 6      | Structure of FAL protocol state machines .....             | 209 |
| 7      | AP-Context state machine .....                             | 210 |
| 7.1    | Overview.....  | 210 |
| 7.2    | Connection object state machine .....                      | 210 |
| 7.2.1  | I/O Connection instance behavior .....                     | 210 |
| 7.2.2  | Bridged Connection instance behavior .....                 | 215 |
| 7.2.3  | Explicit Messaging Connection instance behavior .....      | 217 |
| 8      | FAL service protocol machine (FSPM).....                   | 219 |
| 8.1    | General.....   | 219 |
| 8.2    | Primitive definitions.....                                 | 219 |
| 8.3    | Parameters of primitives .....                             | 224 |
| 8.4    | FSPM state machines .....                                  | 225 |
| 9      | Application relationship protocol machines (ARPMs) .....   | 225 |
| 9.1    | General.....   | 225 |
| 9.2    | Connection-less ARPM (UCMM) .....                          | 226 |
| 9.2.1  | General .....  | 226 |
| 9.2.2  | Primitive definitions .....                                | 226 |
| 9.2.3  | Parameters of primitives .....                             | 227 |
| 9.2.4  | UCMM state machines .....                                  | 228 |
| 9.2.5  | Examples of UCMM sequences .....                           | 233 |
| 9.2.6  | Management UCMM .....                                      | 235 |
| 9.3    | Connection-oriented ARPMs (transports) .....               | 236 |
| 9.3.1  | Transport PDU buffer.....                                  | 236 |
| 9.3.2  | Transport classes .....                                    | 237 |
| 9.3.3  | Common primitive definitions.....                          | 237 |
| 9.3.4  | Parameters of common primitives .....                      | 238 |
| 9.3.5  | Transport state machines – class 0.....                    | 238 |
| 9.3.6  | Transport state machines – class 1.....                    | 242 |
| 9.3.7  | Transport state machines – class 2.....                    | 247 |
| 9.3.8  | Transport state machines – class 3.....                    | 255 |
| 10     | DLL mapping protocol machine 1 (DMPM 1).....               | 265 |
| 10.1   | General.....   | 265 |
| 10.2   | Link producer .....  | 265 |
| 10.3   | Link consumer .....  | 266 |
| 10.4   | Primitive definitions.....                                 | 266 |
| 10.4.1 | Primitives exchanged between DMPM and ARPM.....            | 266 |
| 10.4.2 | Parameters of ARPM/DMPM primitives .....                   | 266 |
| 10.4.3 | Primitives exchanged between data-link layer and DMPM..... | 266 |
| 10.4.4 | Parameters of DMPM/Data-link Layer primitives .....        | 267 |
| 10.4.5 | Network connection ID.....                                 | 268 |
| 10.5   | DMPM state machine .....                                   | 268 |

|        |  |     |
|--------|--|-----|
| 10.5.1 | DMPM states .....  | 268 |
| 10.5.2 | Functions used by DMPM .....                                     | 270 |
| 10.6   | Data-link Layer service selection.....                           | 270 |
| 11     | DLL mapping protocol machine 2 (DMPM 2).....                     | 270 |
| 11.1   | General.....   | 270 |
| 11.2   | Mapping of UCMM PDUs .....                                       | 270 |
| 11.2.1 | General .....  | 270 |
| 11.2.2 | Common requirements for Connection Manager PDU's.....            | 272 |
| 11.2.3 | Forward_Open PDU for class 2 and class 3 connections .....       | 274 |
| 11.2.4 | Forward_Open for class 0 and class 1 connections.....            | 274 |
| 11.2.5 | Forward_close.....   | 279 |
| 11.3   | Mapping of transport class 0 and class 1 PDUs.....               | 279 |
| 11.3.1 | Class 0 and class 1 PDUs.....                                    | 279 |
| 11.3.2 | No dependency on TCP connections .....                           | 279 |
| 11.3.3 | Class 0 and class 1 packet ordering .....                        | 280 |
| 11.3.4 | Screening incoming connected data .....                          | 280 |
| 11.4   | Mapping of transport class 2 and class 3 PDU's.....              | 280 |
| 11.5   | IGMP Usage .....   | 281 |
| 11.5.1 | Background (informative).....                                    | 281 |
| 11.5.2 | IGMP Membership Report messages .....                            | 282 |
| 11.5.3 | IGMP Leave Group messages .....                                  | 282 |
| 11.6   | Quality of Service (QoS) for Type 2 Ethernet messages.....       | 282 |
| 11.6.1 | Overview .....   | 282 |
| 11.6.2 | DSCP format .....  | 283 |
| 11.6.3 | IEEE Std 802.1Q-2018 format.....                                 | 284 |
| 11.6.4 | Mapping Type 2 traffic to DSCP and IEEE Std 802.1Q-2018.....     | 284 |
| 11.6.5 | Usage of DSCP for Type 2 Ethernet .....                          | 285 |
| 11.6.6 | Usage of IEEE Std 802.1Q-2018 for Type 2 Ethernet.....           | 285 |
| 11.6.7 | User considerations with IEEE Std 802.1Q-2018 .....              | 286 |
| 11.7   | Encapsulation using TCP .....                                    | 286 |
| 11.7.1 | General .....  | 286 |
| 11.7.2 | Management of a TCP encapsulation session.....                   | 286 |
| 11.7.3 | TCP connection management.....                                   | 287 |
| 11.8   | Encapsulation using UDP.....                                     | 288 |
| 12     | DLL mapping protocol machine 3 (DMPM 3).....                     | 288 |
|        | Bibliography.....  | 289 |
|        | Figure 1 – Attribute table format and terms .....                | 31  |
|        | Figure 2 – Service request/response parameter.....               | 31  |
|        | Figure 3 – Example of an STD .....                               | 34  |
|        | Figure 4 – Network connection parameters .....                   | 56  |
|        | Figure 5 – Priority/Tick_time bit definition .....               | 59  |
|        | Figure 6 – Member ID/EX description (WORD).....                  | 74  |
|        | Figure 7 – Transport Class Trigger attribute.....                | 118 |
|        | Figure 8 – DN_initial_comm_characteristics attribute format..... | 122 |
|        | Figure 9 – Segment type.....                                     | 131 |
|        | Figure 10 – Port segment.....                                    | 132 |

|   |     |
|---|-----|
| Figure 11 – Logical segment encoding .....  | 134 |
| Figure 12 – Extended network segment .....  | 141 |
| Figure 13 – Symbolic segment encoding .....   | 142 |
| Figure 14 – Encapsulation message .....   | 175 |
| Figure 15 – FixedLengthBitString compact encoding bit placement rules .....   | 199 |
| Figure 16 – Example compact encoding of a SWORD FixedLengthBitString.....   | 199 |
| Figure 17 – Example compact encoding of a WORD FixedLengthBitString.....  | 199 |
| Figure 18 – Example compact encoding of a DWORD FixedLengthBitString .....  | 200 |
| Figure 19 – Example compact encoding of a LWORD FixedLengthBitString .....  | 200 |
| Figure 20 – Example 1 of formal encoding of a structure type specification.....   | 205 |
| Figure 21 – Example 2 of formal encoding of a structure type specification.....   | 205 |
| Figure 22 – Example 3 of formal encoding of a handle structure type specification .....   | 206 |
| Figure 23 – Example 4 of formal encoding of a handle structure type specification .....   | 206 |
| Figure 24 – Example 5 of abbreviated encoding of a structure type specification .....   | 207 |
| Figure 25 – Example 1 of formal encoding of an array type specification.....  | 207 |
| Figure 26 – Example 2 of formal encoding of an array type specification.....  | 208 |
| Figure 27 – Example 1 of abbreviated encoding of an array type specification .....  | 209 |
| Figure 28 – Example 2 of abbreviated encoding of an array type specification .....  | 209 |
| Figure 29 – I/O Connection object state transition diagram .....  | 210 |
| Figure 30 – Bridged Connection object state transition diagram .....  | 215 |
| Figure 31 – Explicit Messaging Connection object state transition diagram .....   | 217 |
| Figure 32 – State transition diagram of UCMM client.....  | 229 |
| Figure 33 – State transition diagram of high–end UCMM server.....   | 231 |
| Figure 34 – State transition diagram of low–end UCMM server .....   | 233 |
| Figure 35 – Sequence diagram for a UCMM with one outstanding message.....   | 234 |
| Figure 36 – Sequence diagram for a UCMM with multiple outstanding messages.....   | 235 |
| Figure 37 – TPDU buffer .....   | 236 |
| Figure 38 – Data flow diagram using a client transport class 0 and server transport<br>class 0 .....                                  | 239 |
| Figure 39 – Sequence diagram of data transfer using transport class 0.....  | 239 |
| Figure 40 – Class 0 client STD .....  | 240 |
| Figure 41 – Class 0 server STD .....  | 241 |
| Figure 42 – Data flow diagram using client transport class 1 and server transport<br>class 1 .....                                    | 242 |
| Figure 43 – Sequence diagram of data transfer using client transport class 1 and<br>server transport class 1 .....                    | 243 |
| Figure 44 – Class 1 client STD .....  | 245 |
| Figure 45 – Class 1 server STD .....  | 246 |
| Figure 46 – Data flow diagram using client transport class 2 and server transport<br>class 2 .....                                    | 248 |
| Figure 47 – Diagram of data transfer using client transport class 2 and server transport<br>class 2 without returned data .....       | 249 |
| Figure 48 – Sequence diagram of data transfer using client transport class 2 and<br>server transport class 2 with returned data ..... | 250 |
| Figure 49 – Class 2 client STD .....  | 251 |

|   |     |
|---|-----|
| Figure 50 – Class 2 server STD .....  | 253 |
| Figure 51 – Data flow diagram using client transport class 3 and server transport class 3 .....                                       | 256 |
| Figure 52 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 without returned data ..... | 257 |
| Figure 53 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 with returned data .....    | 258 |
| Figure 54 – Class 3 client STD .....  | 260 |
| Figure 55 – Class 3 server STD .....  | 263 |
| Figure 56 – Data flow diagram for a link producer and consumer .....  | 265 |
| Figure 57 – State transition diagram for a link producer .....  | 269 |
| Figure 58 – State transition diagram for a link consumer .....  | 270 |
| Figure 59 – DS field in the IP header .....   | 284 |
| Figure 60 – IEEE Std 802.1Q-2018 tagged frame .....   | 284 |
| <br>  |     |
| Table 1 – Get_Attributes_All response service rules .....   | 32  |
| Table 2 – Example class level object/service specific response data of Get_Attributes_All .....                                       | 32  |
| Table 3 – Set_Attributes_All request service rules .....  | 33  |
| Table 4 – Example request data of Set_Attributes_All .....  | 33  |
| Table 5 – State event matrix format .....   | 35  |
| Table 6 – Example state event matrix .....  | 35  |
| Table 7 – UCMM_PDU header format .....  | 39  |
| Table 8 – UCMM command codes .....  | 39  |
| Table 9 – Transport class 0 header .....  | 40  |
| Table 10 – Transport class 1 header .....   | 40  |
| Table 11 – Transport class 2 header .....   | 40  |
| Table 12 – Transport class 3 header .....   | 41  |
| Table 13 – Real-time data header – exclusive owner .....  | 41  |
| Table 14 – Real-time data header– redundant owner .....   | 41  |
| Table 15 – Forward_Open request format .....  | 46  |
| Table 16 – Forward_Open_Good response format .....  | 47  |
| Table 17 – Forward_Open_Bad response format .....   | 47  |
| Table 18 – Large_Forward_Open request format .....  | 48  |
| Table 19 – Large_Forward_Open_Good response format .....  | 49  |
| Table 20 – Large_Forward_Open_Bad response format .....   | 49  |
| Table 21 – Forward_Close request format .....   | 50  |
| Table 22 – Forward_Close_Good response format .....   | 50  |
| Table 23 – Forward_Close_Bad response format .....  | 51  |
| Table 24 – Unconnected_Send request format .....  | 52  |
| Table 25 – Unconnected_Send_Good response format .....  | 52  |
| Table 26 – Unconnected_Send_Bad response format .....   | 53  |
| Table 27 – Get_Connection_Data request format .....   | 54  |
| Table 28 – Get_Connection_Data response format .....  | 54  |



|  |    |
|--|----|
| Table 29 – Search_Connection_Data request format .....                     | 55 |
| Table 30 – Get_Connection_Owner request format .....                       | 55 |
| Table 31 – Get_Connection_Owner response format .....                      | 56 |
| Table 32 – Time-out multiplier.....  | 59 |
| Table 33 – Tick time units .....   | 60 |
| Table 34 – Encoded application path ordering.....                          | 64 |
| Table 35 – Transport class, trigger and Is_Server format .....             | 65 |
| Table 36 – MR_Request_Header format .....                                  | 65 |
| Table 37 – MR_Response_Header format.....                                  | 66 |
| Table 38 – Structure of Get_Attributes_All_ResponsePDU body .....          | 66 |
| Table 39 – Structure of Set_Attributes_All_RequestPDU body .....           | 67 |
| Table 40 – Structure of Get_Attribute_List_RequestPDU body .....           | 67 |
| Table 41 – Structure of Get_Attribute_List_ResponsePDU body .....          | 67 |
| Table 42 – Structure of Set_Attribute_List_RequestPDU body .....           | 67 |
| Table 43 – Structure of Set_Attribute_List_ResponsePDU body.....           | 68 |
| Table 44 – Structure of Reset_RequestPDU body .....                        | 68 |
| Table 45 – Structure of Reset_ResponsePDU body .....                       | 68 |
| Table 46 – Structure of Start_RequestPDU body .....                        | 68 |
| Table 47 – Structure of Start_ResponsePDU body.....                        | 68 |
| Table 48 – Structure of Stop_RequestPDU body.....                          | 69 |
| Table 49 – Structure of Stop_ResponsePDU body .....                        | 69 |
| Table 50 – Structure of Create_RequestPDU body .....                       | 69 |
| Table 51 – Structure of Create_ResponsePDU body .....                      | 69 |
| Table 52 – Structure of Delete_RequestPDU body .....                       | 69 |
| Table 53 – Structure of Delete_ResponsePDU body .....                      | 70 |
| Table 54 – Structure of Get_Attribute_Single_ResponsePDU body .....        | 70 |
| Table 55 – Structure of Set_Attribute_Single_RequestPDU body .....         | 70 |
| Table 56 – Structure of Set_Attribute_Single_ResponsePDU body .....        | 70 |
| Table 57 – Structure of Find_Next_Object_Instance_RequestPDU body .....    | 71 |
| Table 58 – Structure of Find_Next_Object_Instance_ResponsePDU body .....   | 71 |
| Table 59 – Structure of Apply_Attributes_RequestPDU body .....             | 71 |
| Table 60 – Structure of Apply_Attributes_ResponsePDU body .....            | 71 |
| Table 61 – Structure of Save_RequestPDU body .....                         | 71 |
| Table 62 – Structure of Save_ResponsePDU body .....                        | 72 |
| Table 63 – Structure of Restore_RequestPDU body.....                       | 72 |
| Table 64 – Structure of Restore_ResponsePDU body .....                     | 72 |
| Table 65 – Structure of Get_Member_ResponsePDU body .....                  | 72 |
| Table 66 – Structure of Set_Member_RequestPDU body .....                   | 72 |
| Table 67 – Structure of Set_Member_ResponsePDU body.....                   | 73 |
| Table 68 – Structure of Insert_Member_RequestPDU body.....                 | 73 |
| Table 69 – Structure of Insert_Member_ResponsePDU body .....               | 73 |
| Table 70 – Structure of Remove_Member_ResponsePDU body .....               | 73 |
| Table 71 – Common structure of _Member_RequestPDU body (basic format)..... | 74 |

|  |    |
|--|----|
| Table 72 – Common structure of _Member_ResponsePDU body (basic format) .....                                   | 75 |
| Table 73 – Common structure of _Member_RequestPDU body (extended format).....                                  | 75 |
| Table 74 – Common structure of _Member_ResponsePDU body (extended format) .....                                | 75 |
| Table 75 – Extended Protocol ID.....   | 76 |
| Table 76 – Structure of _Member_RequestPDU body (Multiple Sequential Members) .....                            | 76 |
| Table 77 – Structure of _Member_ResponsePDU body (Multiple Sequential Members).....                            | 76 |
| Table 78 – Structure of _Member_RequestPDU body (International String Selection) .....                         | 77 |
| Table 79 – Structure of _Member_ResponsePDU body (International String Selection).....                         | 77 |
| Table 80 – Structure of Group_Sync_RequestPDU body.....  | 77 |
| Table 81 – Structure of Group_Sync_ResponsePDU body .....  | 78 |
| Table 82 – Structure of Multiple_Service_Packet_RequestPDU body.....   | 78 |
| Table 83 – Structure of Multiple_Service_Packet_ResponsePDU body .....   | 78 |
| Table 84 – Structure of Get_Connection_Point_Member_List_ResponsePDU body.....                                 | 79 |
| Table 85 – Identity object class attributes .....  | 80 |
| Table 86 – Identity object instance attributes .....   | 80 |
| Table 87 – Identity object Vendor ID ranges .....  | 83 |
| Table 88 – Identity object bit definitions for status instance attribute.....                                  | 83 |
| Table 89 – Default values for extended device status field (bits 4 to 7) of status<br>instance attribute ..... | 83 |
| Table 90 – Identity object bit definitions for protection mode instance attribute .....                        | 84 |
| Table 91 – Identity object bit definitions for features supported attribute .....                              | 84 |
| Table 92 – Class level object/service specific response data of Get_Attributes_All .....                       | 84 |
| Table 93 – Instance level object/service specific response data of Get_Attributes_All.....                     | 85 |
| Table 94 – Object-specific request parameter for Reset .....   | 86 |
| Table 95 – Reset service parameter values .....  | 86 |
| Table 96 – Communication link attributes that shall be preserved .....   | 86 |
| Table 97 – Structure of Flash_LEDs_RequestPDU body .....   | 87 |
| Table 98 – Message Router object class attributes .....  | 87 |
| Table 99 – Message Router object instance attributes .....   | 87 |
| Table 100 – Class level object/service specific response data of Get_Attributes_All .....                      | 88 |
| Table 101 – Instance level object/service specific response data of Get_Attributes_All.....                    | 88 |
| Table 102 – Structure of Symbolic_Translation_RequestPDU body.....   | 88 |
| Table 103 – Structure of Symbolic_Translation_ResponsePDU body .....   | 88 |
| Table 104 – Object specific status for Symbolic_Translation service .....                                      | 89 |
| Table 105 – Structure of Send_Receive_Fragment_RequestPDU body – Phase 1 .....                                 | 89 |
| Table 106 – Structure of Send_Receive_Fragment_RequestPDU body – Phase 2 .....                                 | 89 |
| Table 107 – Structure of Send_Receive_Fragment_ResponsePDU body – Phase 2.....                                 | 90 |
| Table 108 – Request/Response Fragmentation Flags .....   | 90 |
| Table 109 – Fragmentation Flags Usage.....   | 90 |
| Table 110 – Object specific status for Send_Receive_Fragment service .....                                     | 91 |
| Table 111 – Assembly object class attributes.....  | 92 |
| Table 112 – Assembly object instance attributes.....   | 92 |
| Table 113 – Assembly Instance ID ranges .....  | 93 |

|   |     |
|---|-----|
| Table 114 – Standard Network Diagnostic assembly content and ordering .....   | 94  |
| Table 115 – Object-specific request parameter for Create.....   | 94  |
| Table 116 – Object-specific response parameter for Create .....   | 95  |
| Table 117 – Acknowledge Handler object class attributes .....   | 95  |
| Table 118 – Acknowledge Handler object instance attributes .....  | 95  |
| Table 119 – Structure of Add_AckData_Path_RequestPDU body .....   | 96  |
| Table 120 – Structure of Remove_AckData_Path_RequestPDU body .....  | 96  |
| Table 121 – Time Sync object class attributes .....   | 96  |
| Table 122 – Time Sync object instance attributes .....  | 97  |
| Table 123 – ClockIdentity encoding for different network implementations .....  | 101 |
| Table 124 – ClockClass values .....   | 101 |
| Table 125 – TimeAccuracy values.....  | 102 |
| Table 126 – TimePropertyFlags bit values .....  | 102 |
| Table 127 – TimeSource values .....   | 103 |
| Table 128 – Types of Clock .....  | 103 |
| Table 129 – Network protocol to PortPhysicalAddressInfo mapping .....   | 103 |
| Table 130 – Time Sync connection point 1, Standard Network Diagnostics .....  | 104 |
| Table 131 – Class level object/service specific response data of Get_Attributes_All .....                               | 104 |
| Table 132 – Parameter object class attributes.....  | 105 |
| Table 133 – Parameter Class Descriptor bit values .....   | 105 |
| Table 134 – Parameter object instance attributes.....   | 106 |
| Table 135 – Semantics of Descriptor Instance attribute.....   | 107 |
| Table 136 – Descriptor Scaling bits usage .....   | 107 |
| Table 137 – Minimum and Maximum Value semantics.....  | 108 |
| Table 138 – Scaling Formula attributes .....  | 109 |
| Table 139 – Scaling links .....   | 109 |
| Table 140 – Class level object/service specific response data of Get_Attributes_All .....                               | 110 |
| Table 141 – Instance level object/service specific response data of Get_Attributes_All<br>(Parameter object stub) ..... | 110 |
| Table 142 – Instance level object/service specific response data of Get_Attributes_All<br>(full Parameter object) ..... | 111 |
| Table 143 – Structure of Get_Enum_String_RequestPDU body.....   | 112 |
| Table 144 – Structure of Get_Enum_String_ResponsePDU body .....   | 112 |
| Table 145 – Enumerated strings Type versus Parameter data type .....  | 112 |
| Table 146 – Connection Manager object class attributes.....   | 113 |
| Table 147 – Connection Manager object instance attributes.....  | 113 |
| Table 148 – Connection Manager connection point 1, Standard Network Diagnostics .....                                   | 114 |
| Table 149 – Class level object/service specific response data of Get_Attributes_All .....                               | 114 |
| Table 150 – Instance level object/service specific response data of Get_Attributes_All.....                             | 115 |
| Table 151 – Instance level object/service specific request data of Set_Attributes_All.....                              | 115 |
| Table 152 – Connection object class attributes .....  | 116 |
| Table 153 – Connection object instance attributes .....   | 116 |
| Table 154 – Values assigned to the state attribute .....  | 117 |
| Table 155 – Values assigned to the instance_type attribute .....  | 118 |

|  |     |
|--|-----|
| Table 156 – Possible values within Direction Bit .....                       | 119 |
| Table 157 – Possible values within Production Trigger Bits.....              | 119 |
| Table 158 – Possible values within Transport Class Bits.....                 | 120 |
| Table 159 – TransportClass_Trigger attribute values summary .....            | 120 |
| Table 160 – Transport Class 0 client behavior summary .....                  | 121 |
| Table 161 – Transport Class 1, 2 and 3 client behavior summary.....          | 121 |
| Table 162 – Values defined for the DN_produced_connection_id attribute ..... | 121 |
| Table 163 – Values defined for the DN_consumed_connection_id attribute.....  | 122 |
| Table 164 – Values for the Initial Production Characteristics nibble .....   | 123 |
| Table 165 – Values for the Initial Consumption Characteristics nibble.....   | 124 |
| Table 166 – Values for the watchdog_timeout_action.....                      | 127 |
| Table 167 – Object-specific response parameters for Apply_Attributes .....   | 129 |
| Table 168 – Object-specific response parameter for Set_Attribute_Single..... | 129 |
| Table 169 – Structure of Connection_Bind_RequestPDU body.....                | 129 |
| Table 170 – Object specific status for Connection_Bind service .....         | 129 |
| Table 171 – Structure of Producing_Application_Lookup_RequestPDU body .....  | 130 |
| Table 172 – Structure of Producing_Application_Lookup_ResponsePDU body.....  | 130 |
| Table 173 – Producing_Application_Lookup Service status codes.....           | 130 |
| Table 174 – Possible port segment examples .....                             | 133 |
| Table 175 – TCP/IP link address examples.....                                | 134 |
| Table 176 – Extended Logical Type .....                                      | 135 |
| Table 177 – Electronic key segment format.....                               | 136 |
| Table 178 – Key Format Table (key type 4).....                               | 137 |
| Table 179 – Serial Number Key Format Table (key type 5) .....                | 137 |
| Table 180 – Logical segments examples.....                                   | 138 |
| Table 181 – Network segments .....   | 139 |
| Table 182 – Extended network segment subtype definitions.....                | 141 |
| Table 183 – Symbolic segment examples .....                                  | 143 |
| Table 184 – Data segment.....  | 143 |
| Table 185 – ANSI_Extended_Symbol segment .....                               | 144 |
| Table 186 – Addressing categories .....                                      | 146 |
| Table 187 – Class code ID ranges .....                                       | 147 |
| Table 188 – Class Attribute ID ranges.....                                   | 147 |
| Table 189 – Instance Attribute ID ranges .....                               | 147 |
| Table 190 – Connection Point ranges .....                                    | 148 |
| Table 191 – Service code ranges.....   | 148 |
| Table 192 – Class codes.....   | 149 |
| Table 193 – Reserved class attributes for all object class definitions ..... | 150 |
| Table 194 – Common services list .....                                       | 151 |
| Table 195 – Identity object specific services list.....                      | 151 |
| Table 196 – Message Router object specific services list.....                | 152 |
| Table 197 – Acknowledge Handler object specific services list.....           | 152 |
| Table 198 – Parameter object specific services list.....                     | 152 |

|  |     |
|--|-----|
| Table 199 – Services specific to Connection Manager .....                          | 152 |
| Table 200 – Services specific to Connection object.....                            | 153 |
| Table 201 – Device type numbering .....  | 153 |
| Table 202 – Implementation profile numbering.....                                  | 154 |
| Table 203 – Connection Manager service request error codes .....                   | 155 |
| Table 204 – General status codes.....  | 165 |
| Table 205 – Extended status code for a general status of "Key Failure in path..... | 167 |
| Table 206 – Identity object status codes .....                                     | 168 |
| Table 207 – TCP port numbers .....   | 174 |
| Table 208 – UDP port numbers.....  | 175 |
| Table 209 – Encapsulation header .....   | 175 |
| Table 210 – Encapsulation command codes .....                                      | 176 |
| Table 211 – Encapsulation status codes .....                                       | 177 |
| Table 212 – Nop request encapsulation header .....                                 | 178 |
| Table 213 – RegisterSession request encapsulation header .....                     | 179 |
| Table 214 – RegisterSession request data portion .....                             | 179 |
| Table 215 – RegisterSession reply encapsulation header .....                       | 180 |
| Table 216 – RegisterSession reply data portion (successful) .....                  | 180 |
| Table 217 – UnRegisterSession request encapsulation header .....                   | 181 |
| Table 218 – ListServices request encapsulation header .....                        | 182 |
| Table 219 – ListServices reply encapsulation header.....                           | 182 |
| Table 220 – ListServices reply data portion (successful).....                      | 182 |
| Table 221 – Communications capability flags.....                                   | 183 |
| Table 222 – ListIdentity request encapsulation header.....                         | 184 |
| Table 223 – ListIdentity reply encapsulation header.....                           | 185 |
| Table 224 – ListIdentity reply data portion (successful).....                      | 185 |
| Table 225 – Type 2 identity item .....   | 186 |
| Table 226 – Type 2 Ethernet Capability item.....                                   | 186 |
| Table 227 – ListInterfaces request encapsulation header.....                       | 187 |
| Table 228 – ListInterfaces reply encapsulation header.....                         | 187 |
| Table 229 – SendRRData request encapsulation header .....                          | 188 |
| Table 230 – SendRRData request data portion .....                                  | 188 |
| Table 231 – SendRRData reply encapsulation header .....                            | 189 |
| Table 232 – SendUnitData request encapsulation header .....                        | 189 |
| Table 233 – SendUnitData request data portion.....                                 | 190 |
| Table 234 – Common packet format.....  | 190 |
| Table 235 – CPF item format .....  | 190 |
| Table 236 – Item Type ID numbers .....   | 191 |
| Table 237 – Null address item.....   | 191 |
| Table 238 – Connected address item.....  | 192 |
| Table 239 – Sequenced address item .....   | 192 |
| Table 240 – Unconnected data item.....   | 192 |
| Table 241 – Connected data item .....  | 193 |

|   |     |
|---|-----|
| Table 242 – Sockaddr info items .....   | 193 |
| Table 243 – Usage of CPF items .....  | 194 |
| Table 244 – BOOLEAN encoding .....  | 195 |
| Table 245 – Example compact encoding of a BOOL value .....                            | 195 |
| Table 246 – Encoding of SignedInteger values .....                                    | 196 |
| Table 247 – Example compact encoding of a SignedInteger value .....                   | 196 |
| Table 248 – UnsignedInteger values .....  | 196 |
| Table 249 – Example compact encoding of an UnsignedInteger .....                      | 196 |
| Table 250 – FixedLengthReal values .....  | 196 |
| Table 251 – Example compact encoding of a REAL value .....                            | 197 |
| Table 252 – Example compact encoding of a LREAL value .....                           | 197 |
| Table 253 – FixedLengthReal values .....  | 197 |
| Table 254 – STRING value .....  | 198 |
| Table 255 – STRING2 value .....   | 198 |
| Table 256 – STRINGN value .....   | 198 |
| Table 257 – SHORT_STRING value .....  | 198 |
| Table 258 – Example compact encoding of a STRING value .....                          | 198 |
| Table 259 – Example compact encoding of STRING2 value .....                           | 199 |
| Table 260 – SHORT_STRING type .....   | 199 |
| Table 261 – Example compact encoding of a single dimensional ARRAY .....              | 200 |
| Table 262 – Example compact encoding of a multi-dimensional ARRAY .....               | 201 |
| Table 263 – Example compact encoding of a STRUCTURE .....                             | 201 |
| Table 264 – Identification codes and descriptions of elementary data types .....      | 203 |
| Table 265 – Identification codes and descriptions of constructed data types .....     | 204 |
| Table 266 – Formal structure encoding definition .....                                | 204 |
| Table 267 – Formal structure with handles encoding definition .....                   | 205 |
| Table 268 – Abbreviated structure encoding definition .....                           | 206 |
| Table 269 – Formal array encoding definition .....                                    | 207 |
| Table 270 – Abbreviated array encoding definition .....                               | 208 |
| Table 271 – I/O Connection state event matrix .....                                   | 211 |
| Table 272 – Bridged Connection state event matrix .....                               | 216 |
| Table 273 – Explicit Messaging Connection state event matrix .....                    | 217 |
| Table 274 – Primitives issued by FAL user to FSPM .....                               | 220 |
| Table 275 – Primitives issued by FAL user to FSPM .....                               | 221 |
| Table 276 – Primitives issued by FSPM to FAL user .....                               | 223 |
| Table 277 – Parameters used with primitives exchanged between FAL user and FSPM ..... | 225 |
| Table 278 – Primitives issued by FSPM to ARPM .....                                   | 227 |
| Table 279 – Primitives issued by ARPM to FSPM .....                                   | 227 |
| Table 280 – Parameters used with primitives exchanged between FSPM and ARPM .....     | 228 |
| Table 281 – UCMM client states .....  | 228 |
| Table 282 – State event matrix of UCMM client .....                                   | 229 |
| Table 283 – High-end UCMM server states .....   | 230 |
| Table 284 – State event matrix of high-end UCMM server .....                          | 231 |

|  |     |
|--|-----|
| Table 285 – Low-end UCMM server states.....  | 232 |
| Table 286 – State event matrix of low–end UCMM server .....                            | 233 |
| Table 287 – Notification .....   | 236 |
| Table 288 – Transport classes .....  | 237 |
| Table 289 – Primitives issued by FSPM to ARPM .....                                    | 237 |
| Table 290 – Primitives issued by ARPM to FSPM .....                                    | 238 |
| Table 291 – Parameters used with primitives exchanged between FSPM and ARPM .....      | 238 |
| Table 292 – Class 0 transport client states .....                                      | 240 |
| Table 293 – Class 0 client SEM .....   | 240 |
| Table 294 – Class 0 transport server states .....                                      | 241 |
| Table 295 – Class 0 server SEM .....   | 241 |
| Table 296 – Class 1 transport client states .....                                      | 244 |
| Table 297 – Class 1 client SEM .....   | 245 |
| Table 298 – Class 1 transport server states .....                                      | 246 |
| Table 299 – Class 1 server SEM .....   | 247 |
| Table 300 – Class 2 transport client states .....                                      | 251 |
| Table 301 – Class 2 client SEM .....   | 252 |
| Table 302 – Class 2 transport server states .....                                      | 253 |
| Table 303 – Class 2 server SEM .....   | 254 |
| Table 304 – Class 3 transport client states .....                                      | 259 |
| Table 305 – Class 3 client SEM .....   | 260 |
| Table 306 – Class 3 transport server states .....                                      | 262 |
| Table 307 – Class 3 server SEM .....   | 264 |
| Table 308 – Primitives issued by ARPM to DMPM .....                                    | 266 |
| Table 309 – Primitives issued by DMPM to ARPM .....                                    | 266 |
| Table 310 – Parameters used with primitives exchanged between ARPM and DMPM .....      | 266 |
| Table 311 – Primitives exchanged between data-link layer and DMPM .....                | 267 |
| Table 312 – Parameters used with primitives exchanged between DMPM and Data-link ..... | 267 |
| Table 313 – Selection of connection ID .....   | 268 |
| Table 314 – Link producer states .....   | 268 |
| Table 315 – State event matrix of link producer .....                                  | 269 |
| Table 316 – Link consumer states .....   | 269 |
| Table 317 – State event matrix of link consumer .....                                  | 270 |
| Table 318 – UCMM request .....   | 271 |
| Table 319 – UCMM reply .....   | 272 |
| Table 320 – Network Connection ID selection .....                                      | 273 |
| Table 321 – Sockaddr Info usage .....  | 275 |
| Table 322 – Example multicast assignments .....  | 278 |
| Table 323 – UDP data format for class 0 and class 1 .....                              | 279 |
| Table 324 – Transport class 2 and class 3 connected data .....                         | 281 |
| Table 325 – Default DSCP and IEEE Std 802.1Q-2018 mapping .....                        | 285 |