



IEC 61158-6-2

Edition 4.0 2019-06

# INTERNATIONAL STANDARD

---

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

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

---

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-7007-3

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

## CONTENTS

|  |     |
|--|-----|
| FOREWORD.....  | 14  |
| INTRODUCTION.....  | 16  |
| 1 Scope.....   | 17  |
| 1.1 General.....   | 17  |
| 1.2 Specifications .....   | 17  |
| 1.3 Conformance .....  | 18  |
| 2 Normative references .....   | 18  |
| 3 Terms, definitions, symbols, abbreviated terms and conventions ..... | 20  |
| 3.1 Terms and definitions from other ISO/IEC standards.....            | 20  |
| 3.1.1 Terms and definitions from ISO/IEC 7498-1 .....                  | 20  |
| 3.1.2 Terms and definitions from ISO/IEC 9545 .....                    | 21  |
| 3.1.3 Terms and definitions from ISO/IEC 8824-1 .....                  | 21  |
| 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 .....                                | 29  |
| 3.5 Conventions.....   | 29  |
| 3.5.1 General concept .....  | 29  |
| 3.5.2 Attribute specification .....                                    | 30  |
| 3.5.3 Common services .....  | 30  |
| 3.5.4 State machine conventions .....                                  | 34  |
| 4 Abstract syntax.....   | 35  |
| 4.1 FAL PDU abstract syntax .....                                      | 35  |
| 4.1.1 General .....  | 35  |
| 4.1.2 PDU structure.....   | 36  |
| 4.1.3 UCMM_PDUs .....  | 38  |
| 4.1.4 Transport_Headers.....   | 40  |
| 4.1.5 CM_PDUs.....   | 43  |
| 4.1.6 CM PDU components .....  | 57  |
| 4.1.7 MR headers .....   | 66  |
| 4.1.8 OM_Service_PDU.....  | 67  |
| 4.1.9 Message and connection paths.....                                | 122 |
| 4.1.10 Class, attribute and service codes .....                        | 137 |
| 4.1.11 Error codes.....  | 143 |
| 4.2 Data abstract syntax specification.....                            | 157 |
| 4.2.1 Transport format specification.....                              | 157 |
| 4.2.2 Abstract syntax notation .....                                   | 157 |
| 4.2.3 Control network data specification .....                         | 158 |
| 4.2.4 Data type specification / dictionaries.....                      | 160 |
| 4.3 Encapsulation abstract syntax.....                                 | 162 |
| 4.3.1 Encapsulation protocol .....                                     | 162 |
| 4.3.2 Command descriptions .....                                       | 165 |
| 4.3.3 Common packet format.....  | 177 |
| 5 Transfer syntax.....   | 181 |
| 5.1 Compact encoding .....   | 181 |
| 5.1.1 Encoding rules.....  | 181 |

|        |  |     |
|--------|--|-----|
| 5.1.2  | Encoding constraints .....                                 | 182 |
| 5.1.3  | Examples.....  | 182 |
| 5.2    | Data type reporting .....                                  | 188 |
| 5.2.1  | Object data representation.....                            | 188 |
| 5.2.2  | Elementary data type reporting .....                       | 189 |
| 5.2.3  | Constructed data type reporting.....                       | 190 |
| 6      | Structure of FAL protocol state machines .....             | 196 |
| 7      | AP-Context state machine .....                             | 196 |
| 7.1    | Overview.....  | 196 |
| 7.2    | Connection object state machine .....                      | 196 |
| 7.2.1  | I/O Connection instance behavior .....                     | 196 |
| 7.2.2  | Bridged Connection instance behavior .....                 | 200 |
| 7.2.3  | Explicit Messaging Connection instance behavior .....      | 201 |
| 8      | FAL service protocol machine (FSPM).....                   | 204 |
| 8.1    | General.....   | 204 |
| 8.2    | Primitive definitions.....                                 | 204 |
| 8.3    | Parameters of primitives .....                             | 209 |
| 8.4    | FSPM state machines .....                                  | 209 |
| 9      | Application relationship protocol machines (ARPMs) .....   | 210 |
| 9.1    | General.....   | 210 |
| 9.2    | Connection-less ARPM (UCMM) .....                          | 210 |
| 9.2.1  | General .....  | 210 |
| 9.2.2  | Primitive definitions .....                                | 210 |
| 9.2.3  | Parameters of primitives .....                             | 211 |
| 9.2.4  | UCMM state machines.....                                   | 212 |
| 9.2.5  | Examples of UCMM sequences .....                           | 217 |
| 9.2.6  | Management UCMM .....                                      | 219 |
| 9.3    | Connection-oriented ARPMs (transports) .....               | 220 |
| 9.3.1  | Transport PDU buffer.....                                  | 220 |
| 9.3.2  | Transport classes .....                                    | 220 |
| 9.3.3  | Common primitive definitions.....                          | 221 |
| 9.3.4  | Parameters of common primitives .....                      | 222 |
| 9.3.5  | Transport state machines – class 0.....                    | 222 |
| 9.3.6  | Transport state machines – class 1.....                    | 226 |
| 9.3.7  | Transport state machines – class 2.....                    | 231 |
| 9.3.8  | Transport state machines – class 3.....                    | 239 |
| 9.3.9  | Transport state machines – classes 4, 5, 6 .....           | 249 |
| 9.3.10 | Transport state machines – class 4.....                    | 249 |
| 9.3.11 | Transport state machines – class 5.....                    | 249 |
| 9.3.12 | Transport state machines – class 6.....                    | 249 |
| 10     | DLL mapping protocol machine 1 (DMPM 1).....               | 249 |
| 10.1   | General.....   | 249 |
| 10.2   | Link producer .....  | 250 |
| 10.3   | Link consumer .....  | 250 |
| 10.4   | Primitive definitions.....                                 | 250 |
| 10.4.1 | Primitives exchanged between DMPM and ARPM.....            | 250 |
| 10.4.2 | Parameters of ARPM/DMPM primitives .....                   | 250 |
| 10.4.3 | Primitives exchanged between data-link layer and DMPM..... | 251 |

|        |  |     |
|--------|--|-----|
| 10.4.4 | Parameters of DMPM/Data-link Layer primitives .....        | 251 |
| 10.4.5 | Network connection ID .....                                | 252 |
| 10.5   | DMPM state machine .....                                   | 253 |
| 10.5.1 | DMPM states .....  | 253 |
| 10.5.2 | Functions used by DMPM .....                               | 254 |
| 10.6   | Data-link Layer service selection.....                     | 254 |
| 11     | DLL mapping protocol machine 2 (DMPM 2) .....              | 254 |
| 11.1   | General.....   | 254 |
| 11.2   | Mapping of UCMM PDUs .....                                 | 255 |
| 11.2.1 | General .....  | 255 |
| 11.2.2 | Common requirements for Connection Manager PDU's.....      | 256 |
| 11.2.3 | Forward_open PDU for class 2 and class 3 connections ..... | 258 |
| 11.2.4 | Forward_open for class 0 and class 1 connections .....     | 258 |
| 11.2.5 | Forward_close .....  | 262 |
| 11.3   | Mapping of transport class 0 and class 1 PDUs.....         | 263 |
| 11.3.1 | Class 0 and class 1 PDUs.....                              | 263 |
| 11.3.2 | No dependency on TCP connections .....                     | 263 |
| 11.3.3 | Class 0 and class 1 packet ordering .....                  | 263 |
| 11.3.4 | Screening incoming connected data .....                    | 264 |
| 11.4   | Mapping of transport class 2 and class 3 PDU's.....        | 264 |
| 11.5   | Mapping of transport classes 4 to 6.....                   | 265 |
| 11.6   | IGMP Usage .....   | 265 |
| 11.6.1 | Background (informative).....                              | 265 |
| 11.6.2 | IGMP Membership Report messages .....                      | 266 |
| 11.6.3 | IGMP Leave Group messages .....                            | 266 |
| 11.7   | Quality of Service (QoS) for CP 2/2 messages .....         | 267 |
| 11.7.1 | Overview .....   | 267 |
| 11.7.2 | DSCP format .....  | 267 |
| 11.7.3 | IEEE 802.1D/IEEE 802.Q format.....                         | 268 |
| 11.7.4 | Mapping CPF 2 traffic to DSCP and IEEE 802.1D .....        | 268 |
| 11.7.5 | CP 2/2 usage of DSCP .....                                 | 269 |
| 11.7.6 | CP 2/2 usage of IEEE 802.1D/IEEE 802.1Q.....               | 269 |
| 11.7.7 | User considerations with IEEE 802.1D/IEEE 802.1Q.....      | 269 |
| 11.8   | Management of an encapsulation session.....                | 270 |
| 11.8.1 | Phases of an encapsulation session .....                   | 270 |
| 11.8.2 | Establishing a session .....                               | 270 |
| 11.8.3 | Terminating a session.....                                 | 270 |
| 11.8.4 | Maintaining a session .....                                | 270 |
| 11.8.5 | TCP connection management.....                             | 271 |
| 12     | DLL mapping protocol machine 3 (DMPM 3) .....              | 271 |
|        | Bibliography.....  | 272 |
|        | Figure 1 – Attribute table format and terms .....          | 30  |
|        | Figure 2 – Service request/response parameter .....        | 30  |
|        | Figure 3 – Example of an STD .....                         | 34  |
|        | Figure 4 – Network connection parameters .....             | 58  |
|        | Figure 5 – Priority/Tick_time bit definition .....         | 61  |
|        | Figure 6 – Member ID/EX description (WORD).....            | 75  |

|  |     |
|--|-----|
| Figure 7 – Transport Class Trigger attribute.....  | 110 |
| Figure 8 – CP2/3_initial_comm_characteristics attribute format .....   | 114 |
| Figure 9 – Segment type.....   | 123 |
| Figure 10 – Port segment.....  | 124 |
| Figure 11 – Logical segment encoding.....  | 126 |
| Figure 12 – Extended network segment .....   | 132 |
| Figure 13 – Symbolic segment encoding .....  | 133 |
| Figure 14 – Encapsulation message .....  | 162 |
| Figure 15 – FixedLengthBitString compact encoding bit placement rules .....  | 185 |
| Figure 16 – Example compact encoding of a SWORD FixedLengthBitString.....  | 186 |
| Figure 17 – Example compact encoding of a WORD FixedLengthBitString.....   | 186 |
| Figure 18 – Example compact encoding of a DWORD FixedLengthBitString .....   | 186 |
| Figure 19 – Example compact encoding of a LWORD FixedLengthBitString .....   | 186 |
| Figure 20 – Example 1 of formal encoding of a structure type specification.....                                    | 191 |
| Figure 21 – Example 2 of formal encoding of a structure type specification.....                                    | 191 |
| Figure 22 – Example 3 of formal encoding of a handle structure type specification .....                            | 192 |
| Figure 23 – Example 4 of formal encoding of a handle structure type specification .....                            | 192 |
| Figure 24 – Example 5 of abbreviated encoding of a structure type specification .....                              | 193 |
| Figure 25 – Example 1 of formal encoding of an array type specification.....                                       | 194 |
| Figure 26 – Example 2 of formal encoding of an array type specification.....                                       | 194 |
| Figure 27 – Example 1 of abbreviated encoding of an array type specification .....                                 | 195 |
| Figure 28 – Example 2 of abbreviated encoding of an array type specification .....                                 | 195 |
| Figure 29 – I/O Connection object state transition diagram .....   | 196 |
| Figure 30 – Bridged Connection object state transition diagram .....   | 200 |
| Figure 31 – Explicit Messaging Connection object state transition diagram .....                                    | 202 |
| Figure 32 – State transition diagram of UCMM client9.....  | 212 |
| Figure 33 – State transition diagram of high–end UCMM server.....  | 214 |
| Figure 34 – State transition diagram of low–end UCMM server .....  | 216 |
| Figure 35 – Sequence diagram for a UCMM with one outstanding message.....  | 218 |
| Figure 36 – Sequence diagram for a UCMM with multiple outstanding messages.....                                    | 219 |
| Figure 37 – TPDU buffer .....  | 220 |
| Figure 38 – Data flow diagram using a client transport class 0 and server transport<br>class 0 .....               | 223 |
| Figure 39 – Sequence diagram of data transfer using transport class 0.....   | 223 |
| Figure 40 – Class 0 client STD .....   | 224 |
| Figure 41 – Class 0 server STD .....   | 225 |
| Figure 42 – Data flow diagram using client transport class 1 and server transport<br>class 1 .....                 | 226 |
| Figure 43 – Sequence diagram of data transfer using client transport class 1 and<br>server transport class 1 ..... | 227 |
| Figure 44 – Class 1 client STD .....   | 229 |
| Figure 45 – Class 1 server STD .....   | 230 |
| Figure 46 – Data flow diagram using client transport class 2 and server transport<br>class 2.....                  | 232 |

Figure 47 – Diagram of data transfer using client transport class 2 and server transport class 2 without returned data ..... 233

Figure 48 – Sequence diagram of data transfer using client transport class 2 and server transport class 2 with returned data ..... 234

Figure 49 – Class 2 client STD ..... 235

Figure 50 – Class 2 server STD ..... 237

Figure 51 – Data flow diagram using client transport class 3 and server transport class 3 ..... 240

Figure 52 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 without returned data ..... 241

Figure 53 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 with returned data ..... 242

Figure 54 – Class 3 client STD ..... 244

Figure 55 – Class 3 server STD ..... 247

Figure 56 – Data flow diagram for a link producer and consumer ..... 249

Figure 57 – State transition diagram for a link producer ..... 253

Figure 58 – State transition diagram for a link consumer ..... 254

Figure 59 – DS field in the IP header ..... 268

Figure 60 – IEEE 802.1Q tagged frame ..... 268

  

Table 1 – Get\_Attributes\_All response service rules ..... 31

Table 2 – Example class level object/service specific response data of Get\_Attributes\_All ..... 31

Table 3 – Example Get\_Attributes\_All data array method ..... 32

Table 4 – Set\_Attributes\_All request service rules ..... 33

Table 5 – Example Set\_Attributes\_All attribute ordering method ..... 33

Table 6 – Example Set\_Attributes\_All data array method ..... 33

Table 7 – State event matrix format ..... 35

Table 8 – Example state event matrix ..... 35

Table 9 – UCMM\_PDU header format ..... 39

Table 10 – UCMM command codes ..... 39

Table 11 – Transport class 0 header ..... 40

Table 12 – Transport class 1 header ..... 40

Table 13 – Transport class 2 header ..... 40

Table 14 – Transport class 3 header ..... 40

Table 15 – Real-time data header – exclusive owner ..... 41

Table 16 – Real-time data header– redundant owner ..... 41

Table 17 – Forward\_Open request format ..... 46

Table 18 – Forward\_Open\_Good response format ..... 46

Table 19 – Forward\_Open\_Bad response format ..... 47

Table 20 – Large\_Forward\_Open request format ..... 48

Table 21 – Large\_Forward\_Open\_Good response format ..... 48

Table 22 – Large\_Forward\_Open\_Bad response format ..... 49

Table 23 – Forward\_Close request format ..... 50

Table 24 – Forward\_Close\_Good response format ..... 50

|  |    |
|--|----|
| Table 25 – Forward_Close_Bad response format .....                       | 51 |
| Table 26 – Unconnected_Send request format .....                         | 51 |
| Table 27 – Unconnected_Send_Good response format .....                   | 52 |
| Table 28 – Unconnected_Send_Bad response format .....                    | 53 |
| Table 29 – Unconnected_Send request format (modified) .....              | 54 |
| Table 30 – Unconnected_Send_Good response format (modified) .....        | 54 |
| Table 31 – Unconnected_Send_Bad response format (modified).....          | 55 |
| Table 32 – Get_Connection_Data request format .....                      | 55 |
| Table 33 – Get_Connection_Data response format .....                     | 55 |
| Table 34 – Search_Connection_Data request format .....                   | 56 |
| Table 35 – Get_Connection_Owner request format .....                     | 57 |
| Table 36 – Get_Connection_Owner response format .....                    | 57 |
| Table 37 – Time-out multiplier.....                                      | 60 |
| Table 38 – Tick time units .....   | 61 |
| Table 39 – Encoded application path ordering.....                        | 65 |
| Table 40 – Transport class, trigger and Is_Server format .....           | 66 |
| Table 41 – MR_Request_Header format .....                                | 66 |
| Table 42 – MR_Response_Header format.....                                | 67 |
| Table 43 – Structure of Get_Attributes_All_ResponsePDU body .....        | 67 |
| Table 44 – Structure of Set_Attributes_All_RequestPDU body .....         | 68 |
| Table 45 – Structure of Get_Attribute_List_RequestPDU body .....         | 68 |
| Table 46 – Structure of Get_Attribute_List_ResponsePDU body .....        | 68 |
| Table 47 – Structure of Set_Attribute_List_RequestPDU body .....         | 68 |
| Table 48 – Structure of Set_Attribute_List_ResponsePDU body.....         | 69 |
| Table 49 – Structure of Reset_RequestPDU body .....                      | 69 |
| Table 50 – Structure of Reset_ResponsePDU body .....                     | 69 |
| Table 51 – Structure of Start_RequestPDU body .....                      | 69 |
| Table 52 – Structure of Start_ResponsePDU body .....                     | 69 |
| Table 53 – Structure of Stop_RequestPDU body .....                       | 70 |
| Table 54 – Structure of Stop_ResponsePDU body .....                      | 70 |
| Table 55 – Structure of Create_RequestPDU body .....                     | 70 |
| Table 56 – Structure of Create_ResponsePDU body .....                    | 70 |
| Table 57 – Structure of Delete_RequestPDU body .....                     | 70 |
| Table 58 – Structure of Delete_ResponsePDU body .....                    | 70 |
| Table 59 – Structure of Get_Attribute_Single_ResponsePDU body .....      | 71 |
| Table 60 – Structure of Set_Attribute_Single_RequestPDU body .....       | 71 |
| Table 61 – Structure of Set_Attribute_Single_ResponsePDU body .....      | 71 |
| Table 62 – Structure of Find_Next_Object_Instance_RequestPDU body .....  | 71 |
| Table 63 – Structure of Find_Next_Object_Instance_ResponsePDU body ..... | 72 |
| Table 64 – Structure of Apply_Attributes_RequestPDU body .....           | 72 |
| Table 65 – Structure of Apply_Attributes_ResponsePDU body .....          | 72 |
| Table 66 – Structure of Save_RequestPDU body .....                       | 72 |
| Table 67 – Structure of Save_ResponsePDU body .....                      | 72 |

|  |    |
|--|----|
| Table 68 – Structure of Restore_RequestPDU body .....  | 73 |
| Table 69 – Structure of Restore_ResponsePDU body .....   | 73 |
| Table 70 – Structure of Get_Member_ResponsePDU body .....  | 73 |
| Table 71 – Structure of Set_Member_RequestPDU body .....   | 73 |
| Table 72 – Structure of Set_Member_ResponsePDU body.....   | 73 |
| Table 73 – Structure of Insert_Member_RequestPDU body.....   | 74 |
| Table 74 – Structure of Insert_Member_ResponsePDU body .....   | 74 |
| Table 75 – Structure of Remove_Member_ResponsePDU body .....   | 74 |
| Table 76 – Common structure of _Member_RequestPDU body (basic format).....                                     | 75 |
| Table 77 – Common structure of _Member_ResponsePDU body (basic format) .....                                   | 75 |
| Table 78 – Common structure of _Member_RequestPDU body (extended format).....                                  | 76 |
| Table 79 – Common structure of _Member_ResponsePDU body (extended format) .....                                | 76 |
| Table 80 – Extended Protocol ID.....   | 76 |
| Table 81 – Structure of _Member_RequestPDU body (Multiple Sequential Members) .....                            | 77 |
| Table 82 – Structure of _Member_ResponsePDU body (Multiple Sequential Members).....                            | 77 |
| Table 83 – Structure of _Member_RequestPDU body (International String Selection) .....                         | 78 |
| Table 84 – Structure of _Member_ResponsePDU body (International String Selection).....                         | 78 |
| Table 85 – Structure of Group_Sync_RequestPDU body.....  | 78 |
| Table 86 – Structure of Group_Sync_ResponsePDU body .....  | 78 |
| Table 87 – Structure of Multiple_Service_Packet_RequestPDU body.....   | 79 |
| Table 88 – Structure of Multiple_Service_Packet_ResponsePDU body .....   | 79 |
| Table 89 – Identity object class attributes .....  | 80 |
| Table 90 – Identity object instance attributes .....   | 80 |
| Table 91 – Identity object bit definitions for status instance attribute .....                                 | 81 |
| Table 92 – Default values for extended device status field (bits 4 to 7) of status<br>instance attribute ..... | 82 |
| Table 93 – Identity object bit definitions for protection mode instance attribute .....                        | 82 |
| Table 94 – Class level object/service specific response data of Get_Attributes_All .....                       | 82 |
| Table 95 – Instance level object/service specific response data of Get_Attributes_All.....                     | 83 |
| Table 96 – Object-specific request parameter for Reset .....   | 83 |
| Table 97 – Reset service parameter values .....  | 84 |
| Table 98 – Communication link attributes that shall be preserved .....   | 84 |
| Table 99 – Structure of Flash_LEDs_RequestPDU body .....   | 84 |
| Table 100 – Message Router object class attributes .....   | 85 |
| Table 101 – Message Router object instance attributes .....  | 85 |
| Table 102 – Class level object/service specific response data of Get_Attributes_All .....                      | 85 |
| Table 103 – Instance level object/service specific response data of Get_Attributes_All.....                    | 86 |
| Table 104 – Structure of Symbolic_Translation_RequestPDU body.....   | 86 |
| Table 105 – Structure of Symbolic_Translation_ResponsePDU body .....   | 86 |
| Table 106 – Object specific status for Symbolic_Translation service .....                                      | 86 |
| Table 107 – Assembly object class attributes.....  | 87 |
| Table 108 – Assembly object instance attributes.....   | 87 |
| Table 109 – Assembly Instance ID ranges .....  | 88 |



|   |     |
|---|-----|
| Table 110 – Acknowledge Handler object class attributes .....   | 88  |
| Table 111 – Acknowledge Handler object instance attributes .....  | 89  |
| Table 112 – Structure of Add_AckData_Path_RequestPDU body .....   | 89  |
| Table 113 – Structure of Remove_AckData_Path_RequestPDU body .....  | 89  |
| Table 114 – Time Sync object class attributes .....   | 90  |
| Table 115 – Time Sync object instance attributes .....  | 90  |
| Table 116 – ClockIdentity encoding for different network implementations .....  | 94  |
| Table 117 – ClockClass values .....   | 94  |
| Table 118 – TimeAccuracy values .....   | 94  |
| Table 119 – TimePropertyFlags bit values .....  | 95  |
| Table 120 – TimeSource values .....   | 95  |
| Table 121 – Types of Clock .....  | 96  |
| Table 122 – Network protocol to PortPhysicalAddressInfo mapping .....   | 96  |
| Table 123 – Parameter object class attributes .....   | 97  |
| Table 124 – Parameter Class Descriptor bit values .....   | 97  |
| Table 125 – Parameter object instance attributes .....  | 98  |
| Table 126 – Semantics of Descriptor Instance attribute .....  | 99  |
| Table 127 – Descriptor Scaling bits usage .....   | 99  |
| Table 128 – Minimum and Maximum Value semantics .....   | 100 |
| Table 129 – Scaling Formula attributes .....  | 101 |
| Table 130 – Scaling links .....   | 101 |
| Table 131 – Class level object/service specific response data of Get_Attributes_All .....                               | 102 |
| Table 132 – Instance level object/service specific response data of Get_Attributes_All<br>(Parameter object stub) ..... | 102 |
| Table 133 – Instance level object/service specific response data of Get_Attributes_All<br>(full Parameter object) ..... | 103 |
| Table 134 – Structure of Get_Enum_String_RequestPDU body .....  | 104 |
| Table 135 – Structure of Get_Enum_String_ResponsePDU body .....   | 104 |
| Table 136 – Enumerated strings Type versus Parameter data type .....  | 104 |
| Table 137 – Connection Manager object class attributes .....  | 105 |
| Table 138 – Connection Manager object instance attributes .....   | 105 |
| Table 139 – Class level object/service specific response data of Get_Attributes_All .....                               | 106 |
| Table 140 – Instance level object/service specific response data of Get_Attributes_All .....                            | 106 |
| Table 141 – Instance level object/service specific request data of Set_Attributes_All .....                             | 107 |
| Table 142 – Connection object class attributes .....  | 107 |
| Table 143 – Connection object instance attributes .....   | 108 |
| Table 144 – Values assigned to the state attribute .....  | 109 |
| Table 145 – Values assigned to the instance_type attribute .....  | 110 |
| Table 146 – Possible values within Direction Bit .....  | 111 |
| Table 147 – Possible values within Production Trigger Bits .....  | 111 |
| Table 148 – Possible values within Transport Class Bits .....   | 112 |
| Table 149 – TransportClass_Trigger attribute values summary .....   | 112 |
| Table 150 – Transport Class 0 client behavior summary .....   | 113 |
| Table 151 – Transport Class 1, 2 and 3 client behavior summary .....  | 113 |

|  |     |
|--|-----|
| Table 152 – Values defined for the CP2/3_produced_connection_id attribute .....    | 114 |
| Table 153 – Values defined for the CP2/3_consumed_connection_id attribute .....    | 114 |
| Table 154 – Values for the Initial Production Characteristics nibble .....         | 115 |
| Table 155 – Values for the Initial Consumption Characteristics nibble.....         | 116 |
| Table 156 – Values for the watchdog_timeout_action.....                            | 119 |
| Table 157 – Structure of Connection_Bind_RequestPDU body.....                      | 121 |
| Table 158 – Object specific status for Connection_Bind service .....               | 121 |
| Table 159 – Structure of Producing_Application_Lookup_RequestPDU body .....        | 121 |
| Table 160 – Structure of Producing_Application_Lookup_ResponsePDU body.....        | 122 |
| Table 161 – Producing_Application_Lookup Service status codes.....                 | 122 |
| Table 162 – Possible port segment examples .....                                   | 124 |
| Table 163 – TCP/IP link address examples .....                                     | 125 |
| Table 164 – Extended Logical Type .....  | 126 |
| Table 165 – Electronic key segment format.....                                     | 128 |
| Table 166 – Logical segments examples .....  | 129 |
| Table 167 – Network segments .....   | 130 |
| Table 168 – Extended subtype definitions .....                                     | 132 |
| Table 169 – Symbolic segment examples .....  | 133 |
| Table 170 – Data segment .....   | 134 |
| Table 171 – ANSI_Extended_Symbol segment .....                                     | 134 |
| Table 172 – Addressing categories .....  | 137 |
| Table 173 – Class code ID ranges .....   | 137 |
| Table 174 – Attribute ID ranges .....  | 138 |
| Table 175 – Service code ranges .....  | 138 |
| Table 176 – Class codes.....   | 139 |
| Table 177 – Reserved class attributes for all object class definitions .....       | 140 |
| Table 178 – Common services list .....   | 140 |
| Table 179 – Identity object specific services list.....                            | 141 |
| Table 180 – Message Router object specific services list.....                      | 141 |
| Table 181 – Acknowledge Handler object specific services list.....                 | 141 |
| Table 182 – Parameter object specific services list.....                           | 141 |
| Table 183 – Services specific to Connection Manager .....                          | 142 |
| Table 184 – Services specific to Connection object.....                            | 142 |
| Table 185 – Device type numbering .....  | 143 |
| Table 186 – Connection Manager service request error codes .....                   | 144 |
| Table 187 – General status codes.....  | 153 |
| Table 188 – Extended status code for a general status of "Key Failure in path..... | 155 |
| Table 189 – Identity object status codes .....                                     | 156 |
| Table 190 – Encapsulation header .....   | 163 |
| Table 191 – Encapsulation command codes .....                                      | 163 |
| Table 192 – Encapsulation status codes .....                                       | 165 |
| Table 193 – Nop request encapsulation header .....                                 | 166 |
| Table 194 – RegisterSession request encapsulation header .....                     | 166 |

|   |     |
|---|-----|
| Table 195 – RegisterSession request data portion .....              | 167 |
| Table 196 – RegisterSession reply encapsulation header .....        | 167 |
| Table 197 – RegisterSession reply data portion (successful) .....   | 168 |
| Table 198 – UnRegisterSession request encapsulation header .....    | 168 |
| Table 199 – ListServices request encapsulation header .....         | 169 |
| Table 200 – ListServices reply encapsulation header .....           | 169 |
| Table 201 – ListServices reply data portion (successful) .....      | 170 |
| Table 202 – Communications capability flags .....                   | 170 |
| Table 203 – ListIdentity request encapsulation header .....         | 171 |
| Table 204 – ListIdentity reply encapsulation header .....           | 172 |
| Table 205 – ListIdentity reply data portion (successful) .....      | 172 |
| Table 206 – CPF 2 identity item .....                               | 173 |
| Table 207 – ListInterfaces request encapsulation header .....       | 174 |
| Table 208 – ListInterfaces reply encapsulation header .....         | 174 |
| Table 209 – SendRRData request encapsulation header .....           | 175 |
| Table 210 – SendRRData request data portion .....                   | 175 |
| Table 211 – SendRRData reply encapsulation header .....             | 176 |
| Table 212 – SendUnitData request encapsulation header .....         | 176 |
| Table 213 – SendUnitData request data portion .....                 | 176 |
| Table 214 – Common packet format .....                              | 177 |
| Table 215 – CPF item format .....                                   | 177 |
| Table 216 – Item Type ID numbers .....                              | 177 |
| Table 217 – Null address item .....                                 | 178 |
| Table 218 – Connected address item .....                            | 178 |
| Table 219 – Sequenced address item .....                            | 179 |
| Table 220 – Unconnected data item .....                             | 179 |
| Table 221 – Connected data item .....                               | 179 |
| Table 222 – Sockaddr info items .....                               | 180 |
| Table 223 – Usage of CPF items .....                                | 181 |
| Table 224 – BOOLEAN encoding .....                                  | 182 |
| Table 225 – Example compact encoding of a BOOL value .....          | 182 |
| Table 226 – Encoding of SignedInteger values .....                  | 182 |
| Table 227 – Example compact encoding of a SignedInteger value ..... | 182 |
| Table 228 – UnsignedInteger values .....                            | 183 |
| Table 229 – Example compact encoding of an UnsignedInteger .....    | 183 |
| Table 230 – FixedLengthReal values .....                            | 183 |
| Table 231 – Example compact encoding of a REAL value .....          | 183 |
| Table 232 – Example compact encoding of a LREAL value .....         | 183 |
| Table 233 – FixedLengthReal values .....                            | 184 |
| Table 234 – STRING value .....                                      | 184 |
| Table 235 – STRING2 value .....                                     | 184 |
| Table 236 – STRINGN value .....                                     | 184 |
| Table 237 – SHORT_STRING value .....                                | 185 |

|   |     |
|---|-----|
| Table 238 – Example compact encoding of a STRING value .....                          | 185 |
| Table 239 – Example compact encoding of STRING2 value .....                           | 185 |
| Table 240 – SHORT_STRING type .....   | 185 |
| Table 241 – Example compact encoding of a single dimensional ARRAY .....              | 187 |
| Table 242 – Example compact encoding of a multi-dimensional ARRAY .....               | 187 |
| Table 243 – Example compact encoding of a STRUCTURE .....                             | 188 |
| Table 244 – Identification codes and descriptions of elementary data types .....      | 189 |
| Table 245 – Identification codes and descriptions of constructed data types .....     | 190 |
| Table 246 – Formal structure encoding definition .....                                | 190 |
| Table 247 – Formal structure with handles encoding definition .....                   | 191 |
| Table 248 – Abbreviated structure encoding definition .....                           | 192 |
| Table 249 – Formal array encoding definition .....                                    | 193 |
| Table 250 – Abbreviated array encoding definition .....                               | 195 |
| Table 251 – I/O Connection state event matrix .....                                   | 197 |
| Table 252 – Bridged Connection state event matrix .....                               | 200 |
| Table 253 – Explicit Messaging Connection state event matrix .....                    | 202 |
| Table 254 – Primitives issued by FAL user to FSPM .....                               | 205 |
| Table 255 – Primitives issued by FAL user to FSPM .....                               | 205 |
| Table 256 – Primitives issued by FSPM to FAL user .....                               | 208 |
| Table 257 – Parameters used with primitives exchanged between FAL user and FSPM ..... | 209 |
| Table 258 – Primitives issued by FSPM to ARPM .....                                   | 211 |
| Table 259 – Primitives issued by ARPM to FSPM .....                                   | 211 |
| Table 260 – Parameters used with primitives exchanged between FSPM and ARPM .....     | 211 |
| Table 261 – UCMM client states .....  | 212 |
| Table 262 – State event matrix of UCMM client .....                                   | 213 |
| Table 263 – High-end UCMM server states .....   | 214 |
| Table 264 – State event matrix of high-end UCMM server .....                          | 215 |
| Table 265 – Low-end UCMM server states .....  | 216 |
| Table 266 – State event matrix of low-end UCMM server .....                           | 217 |
| Table 267 – Notification .....  | 220 |
| Table 268 – Transport classes .....   | 221 |
| Table 269 – Primitives issued by FSPM to ARPM .....                                   | 221 |
| Table 270 – Primitives issued by ARPM to FSPM .....                                   | 222 |
| Table 271 – Parameters used with primitives exchanged between FSPM and ARPM .....     | 222 |
| Table 272 – Class 0 transport client states .....                                     | 224 |
| Table 273 – Class 0 client SEM .....  | 224 |
| Table 274 – Class 0 transport server states .....                                     | 225 |
| Table 275 – Class 0 server SEM .....  | 225 |
| Table 276 – Class 1 transport client states .....                                     | 228 |
| Table 277 – Class 1 client SEM .....  | 229 |
| Table 278 – Class 1 transport server states .....                                     | 230 |
| Table 279 – Class 1 server SEM .....  | 231 |
| Table 280 – Class 2 transport client states .....                                     | 235 |

|  |     |
|--|-----|
| Table 281 – Class 2 client SEM .....   | 236 |
| Table 282 – Class 2 transport server states .....                                      | 237 |
| Table 283 – Class 2 server SEM .....   | 238 |
| Table 284 – Class 3 transport client states .....                                      | 243 |
| Table 285 – Class 3 client SEM .....   | 244 |
| Table 286 – Class 3 transport server states .....                                      | 246 |
| Table 287 – Class 3 server SEM .....   | 248 |
| Table 288 – Primitives issued by ARPM to DMPM .....                                    | 250 |
| Table 289 – Primitives issued by DMPM to ARPM .....                                    | 250 |
| Table 290 – Parameters used with primitives exchanged between ARPM and DMPM .....      | 251 |
| Table 291 – Primitives exchanged between data-link layer and DMPM .....                | 251 |
| Table 292 – Parameters used with primitives exchanged between DMPM and Data-link ..... | 251 |
| Table 293 – Selection of connection ID .....   | 252 |
| Table 294 – Link producer states .....   | 253 |
| Table 295 – State event matrix of link producer .....                                  | 253 |
| Table 296 – Link consumer states .....   | 254 |
| Table 297 – State event matrix of link consumer .....                                  | 254 |
| Table 298 – UCMM request .....   | 255 |
| Table 299 – UCMM reply .....   | 256 |
| Table 300 – Network Connection ID selection .....                                      | 257 |
| Table 301 – Sockaddr Info usage .....  | 259 |
| Table 302 – Example multicast assignments .....  | 262 |
| Table 303 – UDP data format for class 0 and class 1 .....                              | 263 |
| Table 304 – Transport class 2 and class 3 connected data .....                         | 265 |
| Table 305 – Default DSCP and IEEE 802.1D mapping .....                                 | 268 |

# INTERNATIONAL ELECTROTECHNICAL COMMISSION

---

## INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

### Part 6-2: Application layer protocol specification – Type 2 elements

#### FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

NOTE Combinations of protocol types are specified in IEC 61784-1 and IEC 61784-2.

International Standard IEC 61158-6-2 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This fourth edition cancels and replaces the third edition published in 2014. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- clarifications of response PDU formats in case of failure in 4.1.5;
- clarifications of connection request priority definition and handling in 4.1.6.5 and 4.1.6.6;
- clarification of connection remaining path in 4.1.6.12;
- extensions of general syntax in 4.1.8.1;
- extensions and clarifications of Identity object PDUs in 4.1.8.2;
- updates of Assembly object PDUs in 4.1.8.4;
- extensions and clarification of Time sync object PDUs in 4.1.8.6;
- updates of Parameter object PDUs in 4.1.8.7;
- extensions of Connection Manager object PDUs in 4.1.8.8;
- clarifications of Connection object PDUs in 4.1.8.9;
- extensions and clarifications of message and connection paths in 4.1.9;
- updates of object and service class codes in 4.1.10 and error codes in 4.1.11;
- updates of data types in 4.2.4, 5.1.3 and 5.2.3;
- updates of the encapsulation abstract syntax in 4.3;
- updates to the DLL mapping protocol machine 2 in Clause 11;
- miscellaneous editorial corrections.

The text of this International Standard is based on the following documents:

| FDIS         | Report on voting |
|--------------|------------------|
| 65C/948/FDIS | 65C/956/RVD      |

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with ISO/IEC Directives, Part 2.

A list of all parts of the IEC 61158 series, published under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

## INTRODUCTION

This document is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC 61158-1.

The application protocol provides the application service by making use of the services available from the data-link or other immediately lower layer. The primary aim of this document is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer application entities (AEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- as a guide for implementers and designers;
- for use in the testing and procurement of equipment;
- as part of an agreement for the admittance of systems into the open systems environment;
- as a refinement to the understanding of time-critical communications within OSI.

This document is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this document together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.



## **INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –**

### **Part 6-2: Application layer protocol specification – Type 2 elements**

## **1 Scope**

### **1.1 General**

The Fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs.”

This part of IEC 61158 provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 2 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This International Standard specifies interactions between remote applications and defines the externally visible behavior provided by the Type 2 fieldbus application layer in terms of

- a) the formal abstract syntax defining the application layer protocol data units conveyed between communicating application entities;
- b) the transfer syntax defining encoding rules that are applied to the application layer protocol data units;
- c) the application context state machine defining the application service behavior visible between communicating application entities;
- d) the application relationship state machines defining the communication behavior visible between communicating application entities.

The purpose of this document is to define the protocol provided to

- a) define the wire-representation of the service primitives defined in IEC 61158-5-2, and
- b) define the externally visible behavior associated with their transfer.

This document specifies the protocol of the Type 2 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI application layer structure (ISO/IEC 9545).

### **1.2 Specifications**

The principal objective of this document is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-2.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols.

### 1.3 Conformance

This document does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems. Conformance is achieved through implementation of this application layer protocol specification.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-1:2019, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-2:2014, *Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition – Type 2 elements*  
IEC 61158-3-2:2014/AMD1:2019

IEC 61158-4-2:2019, *Industrial communication networks – Fieldbus specifications – Part 4-2: Data-link layer protocol specification – Type 2 elements*

IEC 61158-5-2:2019, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements*

IEC 61588:2009, *Precision clock synchronization protocol for networked measurement and control systems*

IEC 61784-3-2, *Industrial communication networks – Profiles – Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2*

IEC 61800-7-202, *Adjustable speed electrical power drive systems – Part 7-202: Generic interface and use of profiles for power drive systems – Profile type 2 specification*

IEC 62026-3:2014, *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC/IEEE 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Standard for Ethernet*

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 8825-1, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 10646, *Information technology – Universal Coded Character Set (UCS)*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ISO 639-2, *Codes for the representation of names of languages – Part 2: Alpha-3 code*

ISO 11898:1993<sup>1</sup>, *Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication*

IEEE Std 802.1D-2004, *IEEE standard for local and metropolitan area networks – Media Access Control (MAC) bridges*, available at <<http://www.ieee.org>> [viewed 2018-09-04]

IEEE Std 802.1Q-2005<sup>2</sup>, *IEEE standard for local and metropolitan area networks – Virtual bridged local area networks*, available at <<http://www.ieee.org>> [viewed 2018-09-04]

IEEE Std 802.3-2015, *IEEE Standard for Ethernet*, available at <<http://www.ieee.org>> [viewed 2018-09-04]

IETF RFC 791, *Internet Protocol*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1035, *Domain Names – Implementation and Specification*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1112, *Host Extensions for IP Multicasting*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1117, *Internet Numbers*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1122, *Requirements for Internet Hosts – Communication Layers*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 1759, *Printer MIB*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2236, *Internet Group Management Protocol, Version 2*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2475, *An Architecture for Differentiated Services*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 2597, *Assured Forwarding PHB Group*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

---

<sup>1</sup> A newer edition of this document has been published, but only the cited edition applies.

<sup>2</sup> A newer edition of this document has been published, but only the cited edition applies.

IETF RFC 2873, *TCP Processing of the IPv4 Precedence Field*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 3140, *Per Hop Behavior Identification Codes*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 3376, *Internet Group Management Protocol, Version 3*, available at <<http://www.ietf.org>> [viewed 2018-09-04]

IETF RFC 4594, *Configuration Guidelines for DiffServ Service Classes*, available at <<http://www.ietf.org>> [viewed 2018-09-04]