

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

EESTI STANDARDI EESSÕNA

NATIONAL FOREWORD

See Eesti standard EVS-EN 61158-6-2:2014 sisaldab Euroopa standardi EN 61158-6-2:2014 inglisekeelset teksti.	This Estonian standard EVS-EN 61158-6-2:2014 consists of the English text of the European standard EN 61158-6-2:2014.
Standard on jõustunud sellekohase teate avaldamisega EVS Teatajas.	This standard has been endorsed with a notification published in the official bulletin of the Estonian Centre for Standardisation.
Euroopa standardimisorganisatsioonid on teinud Euroopa standardi rahvuslikele liikmetele kättesaadavaks 24.10.2014.	Date of Availability of the European standard is 24.10.2014.
Standard on kättesaadav Eesti Standardikeskusest.	The standard is available from the Estonian Centre for Standardisation.

Tagasisidet standardi sisu kohta on võimalik edastada, kasutades EVS-i veebilehel asuvat tagasiside vormi või saates e-kirja meiliaadressile standardiosakond@evs.ee.

ICS 33

Standardite reprodutseerimise ja levitamise õigus kuulub Eesti Standardikeskusele

Andmete paljundamine, taastekitamine, kopeerimine, salvestamine elektroonsesse süsteemi või edastamine ükskõik millises vormis või millisel teel ilma Eesti Standardikeskuse kirjaliku loata on keelatud.

Kui Teil on küsimusi standardite autorikaitse kohta, võtke palun ühendust Eesti Standardikeskusega:
Aru 10, 10317 Tallinn, Eesti; www.evs.ee; telefon 605 5050; e-post info@evs.ee

The right to reproduce and distribute standards belongs to the Estonian Centre for Standardisation

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, without a written permission from the Estonian Centre for Standardisation.

If you have any questions about copyright, please contact Estonian Centre for Standardisation:
Aru 10, 10317 Tallinn, Estonia; www.evs.ee; phone 605 5050; e-mail info@evs.ee

English Version

Industrial communication networks - Fieldbus specifications -
Part 6-2: Application layer protocol specification - Type 2
elements
(IEC 61158-6-2:2014)

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 6-2: Spécification du protocole de la
couche application - Éléments de type 2
(CEI 61158-6-2:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 6-2:
Protokollspezifikation des Application Layer
(Anwendungsschicht) - Typ 2-Elemente
(IEC 61158-6-2:2014)

This European Standard was approved by CENELEC on 2014-09-23. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 65C/764/FDIS, future edition 3 of IEC 61158-6-2, prepared by SC 65C "Industrial networks" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-6-2:2014.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2015-06-23
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2017-09-23

This document supersedes EN 61158-6-2:2012.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61158-6-2:2014 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 61131-3	NOTE	Harmonized as EN 61131-3.
IEC 61784-1:2014	NOTE	Harmonized as EN 61784-1:2014 (not modified).
IEC 61784-2:2014	NOTE	Harmonized as EN 61784-2:2014 (not modified).

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61158-1	2014	Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series	EN 61158-1	2014
IEC 61158-3-2	2014	Industrial communication networks - Fieldbus specifications - Part 3-2: Data-link layer service definition - Type 2 elements	EN 61158-3-2	2014
IEC 61158-4-2	2014	Industrial communication networks - Fieldbus specifications - Part 4-2: Data-link layer protocol specification - Type 2 elements	EN 61158-4-2	¹⁾
IEC 61158-5-2	2014	Industrial communication networks - Fieldbus specifications - Part 5-2: Application layer service definition - Type 2 elements	EN 61158-5-2	2014
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	EN 61784-3-2	-
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	EN 61800-7-202	-

1) To be published.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 62026-3	2008	Low-voltage switchgear and controlgear - Controller-device interfaces (CDIs) - Part 3: DeviceNet	EN 62026-3	2009
ISO/IEC 7498-1	-	Information technology - Open Systems Interconnection - Basic reference model: The basic model	-	-
ISO/IEC 8802-3	-	Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications	-	-
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	Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication	-	-
IEEE 802.1D	2004	IEEE Standard for local and metropolitan area networks - Media Access Control (MAC) Bridges	-	-
IEEE 802.1Q	2005	IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks	-	-
IEEE 802.3	2008	IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications	-	-

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IETF RFC 791	-	Internet Protocol	-	-
IETF RFC 1035	-	Domain Names - Implementation and Specification	-	-
IETF RFC 1112	-	Host Extensions for IP Multicasting	-	-
IETF RFC 1117	-	Internet numbers	-	-
IETF RFC 1122	-	Requirements for Internet Hosts - Communication Layers	-	-
IETF RFC 1759	-	Printer MIB	-	-
IETF RFC 2236	-	Internet Group Management Protocol, Version 2	-	-
IETF RFC 2474	-	Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers	-	-
IETF RFC 2475	-	An Architecture for Differentiated Services	-	-
IETF RFC 2597	-	Assured Forwarding PHB Group	-	-
IETF RFC 2873	-	TCP Processing of the IPv4 Precedence Field	-	-
IETF RFC 3140	-	Per Hop Behavior Identification Codes	-	-
IETF RFC 3246	-	An Expedited Forwarding PHB (Per-Hop Behavior)	-	-
IETF RFC 3376	-	Internet Group Management Protocol, Version 3	-	-
IETF RFC 4594	-	Configuration Guidelines for DiffServ Service Classes	-	-

CONTENTS

FOREWORD.....	12
INTRODUCTION.....	14
1 Scope.....	15
1.1 General.....	15
1.2 Specifications.....	15
1.3 Conformance.....	16
2 Normative references	16
3 Terms, definitions, symbols, abbreviations and conventions	18
3.1 Terms and definitions from other ISO/IEC standards	18
3.2 Terms and definitions from IEC 61158-5-2.....	19
3.3 Additional terms and definitions.....	19
3.4 Abbreviations and symbols.....	26
3.5 Conventions	27
4 Abstract syntax.....	32
4.1 FAL PDU abstract syntax	32
4.2 Data abstract syntax specification	149
4.3 Encapsulation abstract syntax	154
5 Transfer syntax	171
5.1 Compact encoding.....	171
5.2 Data type reporting.....	179
6 Structure of FAL protocol state machines	186
7 AP-Context state machine	187
7.1 Overview	187
7.2 Connection object state machine	187
8 FAL service protocol machine (FSPM).....	195
8.1 General.....	195
8.2 Primitive definitions	195
8.3 Parameters of primitives.....	199
8.4 FSPM state machines.....	200
9 Application relationship protocol machines (ARPMs)	200
9.1 General.....	200
9.2 Connection-less ARPM (UCMM).....	201
9.3 Connection-oriented ARPMs (transports).....	210
10 DLL mapping protocol machine 1 (DMPM 1).....	238
10.1 General.....	238
10.2 Link producer	238
10.3 Link consumer.....	239
10.4 Primitive definitions	239
10.5 DMPM state machine	241
10.6 Data-link Layer service selection	243
11 DLL mapping protocol machine 2 (DMPM 2).....	243
11.1 General.....	243
11.2 Mapping of UCMM PDUs.....	243
11.3 Mapping of transport class 0 and class 1 PDUs	251
11.4 Mapping of transport class 2 and class 3 PDU's	252

11.5 Mapping of transport classes 4 to 6	253
11.6 IGMP Usage.....	253
11.7 Quality of Service (QoS) for CP 2/2 messages	254
11.8 Management of an encapsulation session	258
12 DLL mapping protocol machine 3 (DMPM 3)	258
Bibliography.....	259
Figure 1 – Attribute table format and terms	27
Figure 2 – Service request/response parameter	28
Figure 3 – Example of an STD	31
Figure 4 – Network connection parameters	54
Figure 5 – Time tick	57
Figure 6 – Connection establishment time-out	59
Figure 7 – Member ID/EX description (WORD)	71
Figure 8 – Transport Class Trigger attribute.....	103
Figure 9 – CP2/3_initial_comm_characteristics attribute format	107
Figure 10 – Segment type.....	116
Figure 11 – Port segment.....	117
Figure 12 – Logical segment encoding.....	119
Figure 13 – Extended network segment	124
Figure 14 – Symbolic segment encoding.....	125
Figure 15 – Encapsulation message	154
Figure 16 – FixedLengthBitString compact encoding bit placement rules	176
Figure 17 – Example compact encoding of a SWORD FixedLengthBitString.....	176
Figure 18 – Example compact encoding of a WORD FixedLengthBitString	176
Figure 19 – Example compact encoding of a DWORD FixedLengthBitString	176
Figure 20 – Example compact encoding of a LWORD FixedLengthBitString	176
Figure 21 – Example 1 of formal encoding of a structure type specification.....	181
Figure 22 – Example 2 of formal encoding of a structure type specification.....	182
Figure 23 – Example 3 of formal encoding of a handle structure type specification	182
Figure 24 – Example 4 of formal encoding of a handle structure type specification	183
Figure 25 – Example 5 of abbreviated encoding of a structure type specification	183
Figure 26 – Example 1 of formal encoding of an array type specification	184
Figure 27 – Example 2 of formal encoding of an array type specification	185
Figure 28 – Example 1 of abbreviated encoding of an array type specification	186
Figure 29 – Example 2 of abbreviated encoding of an array type specification	186
Figure 30 – I/O Connection object state transition diagram	187
Figure 31 – Bridged Connection object state transition diagram	191
Figure 32 – Explicit Messaging Connection object state transition diagram	192
Figure 33 – State transition diagram of UCMM client9.....	203
Figure 34 – State transition diagram of high-end UCMM server.....	205
Figure 35 – State transition diagram of low-end UCMM server	207
Figure 36 – Sequence diagram for a UCMM with one outstanding message.....	208
Figure 37 – Sequence diagram for a UCMM with multiple outstanding messages.....	209

Figure 38 – TPDU buffer	210
Figure 39 – Data flow diagram using a client transport class 0 and server transport class 0	213
Figure 40 – Sequence diagram of data transfer using transport class 0.....	213
Figure 41 – Class 0 client STD	214
Figure 42 – Class 0 server STD	215
Figure 43 – Data flow diagram using client transport class 1 and server transport class 1	216
Figure 44 – Sequence diagram of data transfer using client transport class 1 and server transport class 1	217
Figure 45 – Class 1 client STD	219
Figure 46 – Class 1 server STD	220
Figure 47 – Data flow diagram using client transport class 2 and server transport class 2	222
Figure 48 – Diagram of data transfer using client transport class 2 and server transport class 2 without returned data	223
Figure 49 – Sequence diagram of data transfer using client transport class 2 and server transport class 2 with returned data	224
Figure 50 – Class 2 client STD	225
Figure 51 – Class 2 server STD	227
Figure 52 – Data flow diagram using client transport class 3 and server transport class 3	230
Figure 53 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 without returned data.....	231
Figure 54 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 with returned data	232
Figure 55 – Class 3 client STD	234
Figure 56 – Class 3 server STD	236
Figure 57 – Data flow diagram for a link producer and consumer	238
Figure 58 – State transition diagram for a link producer	242
Figure 59 – State transition diagram for a link consumer.....	242
Figure 60 – DS field in the IP header	255
Figure 61 – IEEE 802.1Q tagged frame.....	256
Table 1 – Get_Attribute_All response service rules	28
Table 2 – Example class level object/service specific response data of Get_Attribute_All	29
Table 3 – Example Get_Attribute_All data array method	29
Table 4 – Set_Attribute_All request service rules	30
Table 5 – Example Set_Attribute_All attribute ordering method	30
Table 6 – Example Set_Attribute_All data array method.....	30
Table 7 – State event matrix format	32
Table 8 – Example state event matrix	32
Table 9 – UCMM_PDU header format	36
Table 10 – UCMM command codes.....	36
Table 11 – Transport class 0 header	37

Table 12 – Transport class 1 header	37
Table 13 – Transport class 2 header	37
Table 14 – Transport class 3 header	37
Table 15 – Real-time data header – exclusive owner	38
Table 16 – Real-time data header– redundant owner	38
Table 17 – Forward_Open request format	42
Table 18 – Forward_Open_Good response format	43
Table 19 – Forward_Open_Bad response format	44
Table 20 – Large_Forward_Open request format	44
Table 21 – Large_Forward_Open_Good response format	45
Table 22 – Large_Forward_Open_Bad response format.....	46
Table 23 – Forward_Close request format	46
Table 24 – Forward_Close_Good response format.....	47
Table 25 – Forward_Close_Bad response format	47
Table 26 – Unconnected_Send request format.....	48
Table 27 – Unconnected_Send_Good response format.....	49
Table 28 – Unconnected_Send_Bad response format	49
Table 29 – Unconnected_Send request format (modified)	50
Table 30 – Unconnected_Send_Good response format (modified)	51
Table 31 – Unconnected_Send_Bad response format (modified).....	51
Table 32 – Get_Connection_Data request format.....	52
Table 33 – Get_Connection_Data response format	52
Table 34 – Search_Connection_Data request format	53
Table 35 – Get_Connection_Owner request format	53
Table 36 – Get_Connection_Owner response format	53
Table 37 – Time-out multiplier.....	57
Table 38 – Time tick units	57
Table 39 – Encoded application path ordering	62
Table 40 – Transport class, trigger and Is_Server format	63
Table 41 – MR_Request_Header format	63
Table 42 – MR_Response_Header format.....	64
Table 43 – Structure of Get_Attribute_All_ResponsePDU body	64
Table 44 – Structure of Set_Attribute_All_RequestPDU body	65
Table 45 – Structure of Get_Attribute_List_RequestPDU body	65
Table 46 – Structure of Get_Attribute_List_ResponsePDU body	65
Table 47 – Structure of Set_Attribute_List_RequestPDU body	65
Table 48 – Structure of Set_Attribute_List_ResponsePDU body.....	65
Table 49 – Structure of Reset_RequestPDU body	66
Table 50 – Structure of Reset_ResponsePDU body	66
Table 51 – Structure of Start_RequestPDU body	66
Table 52 – Structure of Start_ResponsePDU body	66
Table 53 – Structure of Stop_RequestPDU body	66
Table 54 – Structure of Stop_ResponsePDU body	67

Table 55 – Structure of Create_RequestPDU body	67
Table 56 – Structure of Create_ResponsePDU body	67
Table 57 – Structure of Delete_RequestPDU body	67
Table 58 – Structure of Delete_ResponsePDU body	67
Table 59 – Structure of Get_Attribute_Single_ResponsePDU body	68
Table 60 – Structure of Set_Attribute_Single_RequestPDU body	68
Table 61 – Structure of Set_Attribute_Single_ResponsePDU body	68
Table 62 – Structure of Find_Next_Object_Instance_RequestPDU body	68
Table 63 – Structure of Find_Next_Object_Instance_ResponsePDU body	68
Table 64 – Structure of Apply_Attributes_RequestPDU body	69
Table 65 – Structure of Apply_Attributes_ResponsePDU body	69
Table 66 – Structure of Save_RequestPDU body	69
Table 67 – Structure of Save_ResponsePDU body	69
Table 68 – Structure of Restore_RequestPDU body	69
Table 69 – Structure of Restore_ResponsePDU body	70
Table 70 – Structure of Get_Member_ResponsePDU body	70
Table 71 – Structure of Set_Member_RequestPDU body	70
Table 72 – Structure of Set_Member_ResponsePDU body	70
Table 73 – Structure of Insert_Member_RequestPDU body	70
Table 74 – Structure of Insert_Member_ResponsePDU body	71
Table 75 – Structure of Remove_Member_ResponsePDU body	71
Table 76 – Common structure of _Member_RequestPDU body (basic format)	72
Table 77 – Common structure of _Member_ResponsePDU body (basic format)	72
Table 78 – Common structure of _Member_RequestPDU body (extended format)	72
Table 79 – Common structure of _Member_ResponsePDU body (extended format)	73
Table 80 – Extended Protocol ID	73
Table 81 – Structure of _Member_RequestPDU body (Multiple Sequential Members)	73
Table 82 – Structure of _Member_ResponsePDU body (Multiple Sequential Members)	74
Table 83 – Structure of _Member_RequestPDU body (International String Selection)	74
Table 84 – Structure of _Member_ResponsePDU body (International String Selection)	74
Table 85 – Structure of Group_Sync_RequestPDU body	75
Table 86 – Structure of Group_Sync_ResponsePDU body	75
Table 87 – Identity object class attributes	75
Table 88 – Identity object instance attributes	75
Table 89 – Identity object bit definitions for status instance attribute	77
Table 90 – Default values for extended device status field (bits 4 to 7) of status instance attribute	77
Table 91 – Class level object/service specific response data of Get_Attribute_All	77
Table 92 – Instance level object/service specific response data of Get_Attribute_All	78
Table 93 – Object-specific parameter for Reset	78
Table 94 – Reset service parameter values	78
Table 95 – Message Router object class attributes	79
Table 96 – Message Router object instance attributes	79

Table 97 – Class level object/service specific response data of Get_Attribute_All	79
Table 98 – Instance level object/service specific response data of Get_Attribute_All	80
Table 99 – Structure of Symbolic_Translation_RequestPDU body	80
Table 100 – Structure of Symbolic_Translation_ResponsePDU body	80
Table 101 – Object specific status for Symbolic_Translation service	80
Table 102 – Assembly object class attributes	81
Table 103 – Assembly object instance attributes	81
Table 104 – Assembly Instance ID ranges	81
Table 105 – Acknowledge Handler object class attributes	82
Table 106 – Acknowledge Handler object instance attributes	83
Table 107 – Structure of Add_AckData_Path_RequestPDU body	83
Table 108 – Structure of Remove_AckData_Path_RequestPDU body	83
Table 109 – Time Sync object class attributes	84
Table 110 – Time Sync object instance attributes	84
Table 111 – ClockIdentity encoding for different network implementations	87
Table 112 – ClockClass values	88
Table 113 – TimeAccuracy values	88
Table 114 – TimePropertyFlags bit values	89
Table 115 – TimeSource values	89
Table 116 – Types of Clock	89
Table 117 – Network protocol to PortPhysicalAddressInfo mapping	89
Table 118 – Parameter object class attributes	90
Table 119 – Parameter Class Descriptor bit values	90
Table 120 – Parameter object instance attributes	91
Table 121 – Semantics of Descriptor Instance attribute	92
Table 122 – Minimum and Maximum Value semantics	92
Table 123 – Scaling Formula attributes	93
Table 124 – Scaling links	94
Table 125 – Class level object/service specific response data of Get_Attribute_All	95
Table 126 – Instance level object/service specific response data of Get_Attribute_All (Parameter object stub)	95
Table 127 – Instance level object/service specific response data of Get_Attribute_All (full Parameter object)	96
Table 128 – Structure of Get_Enum_String_RequestPDU body	97
Table 129 – Structure of Get_Enum_String_ResponsePDU body	97
Table 130 – Enumerated strings Type versus Parameter data type	97
Table 131 – Connection Manager object class attributes	98
Table 132 – Connection Manager object instance attributes	98
Table 133 – Class level object/service specific response data of Get_Attribute_All	99
Table 134 – Instance level object/service specific response data of Get_Attribute_All	99
Table 135 – Instance level object/service specific request data of Set_Attribute_All	100
Table 136 – Connection object class attributes	101
Table 137 – Connection object instance attributes	101
Table 138 – Values assigned to the state attribute	102

Table 139 – Values assigned to the instance_type attribute	103
Table 140 – Possible values within Direction Bit	104
Table 141 – Possible values within Production Trigger Bits	104
Table 142 – Possible values within Transport Class Bits	105
Table 143 – TransportClass_Trigger attribute values summary	105
Table 144 – Transport Class 0 client behavior summary	106
Table 145 – Transport Class 1, 2 and 3 client behavior summary	106
Table 146 – Values defined for the CP2/3_produced_connection_id attribute	106
Table 147 – Values defined for the CP2/3_consumed_connection_id attribute	107
Table 148 – Values for the Initial Production Characteristics nibble	108
Table 149 – Values for the Initial Consumption Characteristics nibble	109
Table 150 – Values for the watchdog_timeout_action	112
Table 151 – Structure of Connection_Bind_RequestPDU body	114
Table 152 – Object specific status for Connection_Bind service	114
Table 153 – Structure of Producing_Application_Lookup_RequestPDU body	114
Table 154 – Structure of Producing_Application_Lookup_ResponsePDU body	114
Table 155 – Producing_Application_Lookup Service status codes	115
Table 156 – Possible port segment examples	117
Table 157 – TCP/IP link address examples	118
Table 158 – Extended Logical Type	119
Table 159 – Electronic key segment format	121
Table 160 – Logical segments examples	122
Table 161 – Network segments	122
Table 162 – Extended subtype definitions	124
Table 163 – Symbolic segment examples	125
Table 164 – Data segment	126
Table 165 – ANSI_Extended_Symbol segment	126
Table 166 – Addressing categories	129
Table 167 – Class code ID ranges	129
Table 168 – Attribute ID ranges	130
Table 169 – Service code ranges	130
Table 170 – Class codes	131
Table 171 – Reserved class attributes for all object class definitions	131
Table 172 – Common services list	132
Table 173 – Message Router object specific services list	133
Table 174 – Acknowledge Handler object specific services list	133
Table 175 – Parameter object specific services list	133
Table 176 – Services specific to Connection Manager	133
Table 177 – Services specific to Connection object	134
Table 178 – Device type numbering	134
Table 179 – Connection Manager service request error codes	136
Table 180 – General status codes	144
Table 181 – Extended status code for a general status of "Key Failure in path"	146

Table 182 – Identity object status codes	147
Table 183 – Encapsulation header	155
Table 184 – Encapsulation command codes	155
Table 185 – Encapsulation status codes	157
Table 186 – Nop request encapsulation header	158
Table 187 – RegisterSession request encapsulation header	158
Table 188 – RegisterSession request data portion	158
Table 189 – RegisterSession reply encapsulation header	159
Table 190 – RegisterSession reply data portion	159
Table 191 – UnRegisterSession request encapsulation header	160
Table 192 – ListServices request encapsulation header	160
Table 193 – ListServices reply encapsulation header	161
Table 194 – ListServices reply data portion	161
Table 195 – Communications capability flags	162
Table 196 – ListIdentity request encapsulation header	162
Table 197 – ListIdentity reply encapsulation header	163
Table 198 – ListIdentity reply data portion (successful)	163
Table 199 – CPF 2 identity item	164
Table 200 – ListInterfaces request encapsulation header	164
Table 201 – ListInterfaces reply encapsulation header	165
Table 202 – SendRRData request encapsulation header	165
Table 203 – SendRRData request data portion	166
Table 204 – SendRRData reply encapsulation header	166
Table 205 – SendUnitData request encapsulation header	167
Table 206 – SendUnitData request data portion	167
Table 207 – Common packet format	167
Table 208 – CPF item format	168
Table 209 – Item Type ID numbers	168
Table 210 – Null address item	169
Table 211 – Connected address item	169
Table 212 – Sequenced address item	169
Table 213 – Unconnected data item	169
Table 214 – Connected data item	170
Table 215 – Sockaddr info items	170
Table 216 – Usage of CPF items	171
Table 217 – BOOLEAN encoding	172
Table 218 – Example compact encoding of a BOOL value	172
Table 219 – Encoding of SignedInteger values	173
Table 220 – Example compact encoding of a SignedInteger value	173
Table 221 – UnsignedInteger values	173
Table 222 – Example compact encoding of an UnsignedInteger	173
Table 223 – FixedLengthReal values	173
Table 224 – Example compact encoding of a REAL value	174

Table 225 – Example compact encoding of a LREAL value	174
Table 226 – FixedLengthReal values	174
Table 227 – STRING value	174
Table 228 – STRING2 value	175
Table 229 – STRINGN value	175
Table 230 – SHORT_STRING value	175
Table 231 – Example compact encoding of a STRING value	175
Table 232 – Example compact encoding of STRING2 value	175
Table 233 – SHORT_STRING type	175
Table 234 – Example compact encoding of a single dimensional ARRAY	177
Table 235 – Example compact encoding of a multi-dimensional ARRAY	178
Table 236 – Example compact encoding of a STRUCTURE	178
Table 237 – Identification codes and descriptions of elementary data types	179
Table 238 – Identification codes and descriptions of constructed data types	180
Table 239 – Formal structure encoding definition	181
Table 240 – Formal structure with handles encoding definition	182
Table 241 – Abbreviated structure encoding definition	183
Table 242 – Formal array encoding definition	184
Table 243 – Abbreviated array encoding definition	185
Table 244 – I/O Connection state event matrix	188
Table 245 – Bridged Connection state event matrix	191
Table 246 – Explicit Messaging Connection state event matrix	193
Table 247 – Primitives issued by FAL user to FSPM	196
Table 248 – Primitives issued by FAL user to FSPM	196
Table 249 – Primitives issued by FSPM to FAL user	198
Table 250 – Parameters used with primitives exchanged between FAL user and FSPM	200
Table 251 – Primitives issued by FSPM to ARPM	202
Table 252 – Primitives issued by ARPM to FSPM	202
Table 253 – Parameters used with primitives exchanged between FSPM and ARPM	202
Table 254 – UCMM client states	203
Table 255 – State event matrix of UCMM client	204
Table 256 – High-end UCMM server states	205
Table 257 – State event matrix of high-end UCMM server	206
Table 258 – Low-end UCMM server states	207
Table 259 – State event matrix of low-end UCMM server	207
Table 260 – Notification	210
Table 261 – Transport classes	211
Table 262 – Primitives issued by FSPM to ARPM	211
Table 263 – Primitives issued by ARPM to FSPM	212
Table 264 – Parameters used with primitives exchanged between FSPM and ARPM	212
Table 265 – Class 0 transport client states	214
Table 266 – Class 0 client SEM	214
Table 267 – Class 0 transport server states	215

Table 268 – Class 0 server SEM	215
Table 269 – Class 1 transport client states	218
Table 270 – Class 1 client SEM	219
Table 271 – Class 1 transport server states	220
Table 272 – Class 1 server SEM	221
Table 273 – Class 2 transport client states	225
Table 274 – Class 2 client SEM	226
Table 275 – Class 2 transport server states	227
Table 276 – Class 2 server SEM	228
Table 277 – Class 3 transport client states	233
Table 278 – Class 3 client SEM	234
Table 279 – Class 3 transport server states	235
Table 280 – Class 3 server SEM	237
Table 281 – Primitives issued by ARPM to DMPM	239
Table 282 – Primitives issued by DMPM to ARPM	239
Table 283 – Parameters used with primitives exchanged between ARPM and DMPM	239
Table 284 – Primitives exchanged between data-link layer and DMPM	240
Table 285 – Parameters used with primitives exchanged between DMPM and Data-link	240
Table 286 – Selection of connection ID	241
Table 287 – Link producer states	241
Table 288 – State event matrix of link producer	242
Table 289 – Link consumer states	242
Table 290 – State event matrix of link consumer	243
Table 291 – UCMM request	243
Table 292 – UCMM reply	244
Table 293 – Network Connection ID selection	246
Table 294 – Sockaddr Info usage	247
Table 295 – Example multicast assignments	250
Table 296 – UDP data format for class 0 and class 1	251
Table 297 – Transport class 2 and class 3 connected data	252
Table 298 – Default DSCP and IEEE 802.1D mapping	256

INTRODUCTION

This part of IEC 61158 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 standard 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 standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard 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 standard 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 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 standard 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 standard 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 standard 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.