



IEC 61158-6-2

Edition 1.0 2007-12

INTERNATIONAL STANDARD

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE

XL

ICS 35.100.70; 25.040.40

ISBN 2-8318-9474-3

CONTENTS

FOREWORD	12
INTRODUCTION	14
1 Scope	15
1.1 General	15
1.2 Specifications	15
1.3 Conformance	15
2 Normative references	16
3 Terms, definitions, symbols, abbreviations and conventions	17
3.1 Terms and definitions from other ISO/IEC standards	17
3.2 Terms and definitions from IEC 61158-5-2	18
3.3 Additional terms and definitions	18
3.4 Abbreviations and symbols	24
3.5 Conventions	24
3.6 Conventions used in state machines	28
4 Abstract syntax	30
4.1 FAL PDU abstract syntax	30
4.2 Data abstract syntax specification	110
4.3 Encapsulation abstract syntax	114
5 Transfer syntax	128
5.1 Compact encoding	128
5.2 Data type reporting	136
6 Structure of FAL protocol state machines	141
7 AP-Context state machine	142
7.1 Overview	142
7.2 Connection object state machine	142
8 FAL service protocol machine (FSPM)	151
8.1 General	151
8.2 Primitive definitions	151
8.3 Parameters of primitives	155
8.4 FSPM state machines	156
9 Application relationship protocol machines (ARPMs)	157
9.1 General	157
9.2 Connection-less ARPM (UCMM)	157
9.3 Connection-oriented ARPMs (transports)	167
10 DLL mapping protocol machine 1 (DMPM 1)	237
10.1 General	237
10.2 Link producer	237
10.3 Link consumer	237
10.4 Primitive definitions	237
10.5 DMPM state machine	239
10.6 Data-link Layer service selection	241
11 DLL mapping protocol machine 2 (DMPM 2)	241
11.1 General	241
11.2 Mapping of UCMM PDUs	241
11.3 Mapping of transport class 0 and class 1 PDUs	246
11.4 Mapping of transport class 2 and class 3 PDU's	248

11.5 Mapping of transport classes 4 to 6	248
11.6 IGMP Usage	248
11.7 Management of an encapsulation session	249
12 DLL mapping protocol machine 3 (DMPM 3)	250
Bibliography	251
Figure 1 – Attribute table format and terms	24
Figure 2 – Service request/response parameter	25
Figure 3 – Example of an STD	29
Figure 4 – Network connection parameters	48
Figure 5 – Time tick	50
Figure 6 – Connection establishment time-out	52
Figure 7 – Transport Class Trigger attribute	76
Figure 8 – CP2/3_initial_comm_characteristics attribute format	79
Figure 9 – Segment type	87
Figure 10 – Port segment	88
Figure 11 – Logical segment encoding	90
Figure 12 – Extended network segment	94
Figure 13 – Encapsulation message	115
Figure 14 – FixedLengthBitString compact encoding bit placement rules	132
Figure 15 – Example compact encoding of a SWORD FixedLengthBitString	133
Figure 16 – Example compact encoding of a WORD FixedLengthBitString	133
Figure 17 – Example compact encoding of a DWORD FixedLengthBitString	133
Figure 18 – Example compact encoding of a LWORD FixedLengthBitString	133
Figure 19 – Example 2 of formal encoding of a structure type specification	138
Figure 20 – Example of abbreviated encoding of a structure type specification	138
Figure 21 – Example 1 of formal encoding of an array type specification	139
Figure 22 – Example 2 of formal encoding of an array type specification	139
Figure 23 – Example 1 of abbreviated encoding of an array type specification	140
Figure 24 – Example 2 of abbreviated encoding of an array type specification	140
Figure 25 – I/O Connection object state transition diagram	142
Figure 26 – Bridged Connection object state transition diagram	146
Figure 27 – Explicit Messaging Connection object state transition diagram	148
Figure 28 – State transition diagram of UCMM client	160
Figure 29 – State transition diagram of high-end UCMM server	161
Figure 30 – State transition diagram of low-end UCMM server	164
Figure 31 – Sequence diagram for a UCMM with one outstanding message	165
Figure 32 – Sequence diagram for a UCMM with multiple outstanding messages	166
Figure 33 – TPDU buffer	167
Figure 34 – Data flow diagram using a client transport class 0 and server transport class 0	170
Figure 35 – Sequence diagram of data transfer using transport class 0	170
Figure 36 – Class 0 client STD	171
Figure 37 – Class 0 server STD	172

Figure 38 – Data flow diagram using client transport class 1 and server transport class 1	173
Figure 39 – Sequence diagram of data transfer using client transport class 1 and server transport class 1	174
Figure 40 – Class 1 client STD	176
Figure 41 – Class 1 server STD	177
Figure 42 – Data flow diagram using client transport class 2 and server transport class 2	179
Figure 43 – Diagram of data transfer using client transport class 2 and server transport class 2 <i>without</i> returned data	180
Figure 44 – Sequence diagram of data transfer using client transport class 2 and server transport class 2 <i>with</i> returned data	181
Figure 45 – Class 2 client STD	182
Figure 46 – Class 2 server STD	184
Figure 47 – Data flow diagram using client transport class 3 and server transport class 3	187
Figure 48 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 <i>without</i> returned data	188
Figure 49 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 <i>with</i> returned data	189
Figure 50 – Class 3 client STD	191
Figure 51 – Class 3 server STD	194
Figure 52 – Data flow diagram using transport classes 4 and 5	196
Figure 53 – Sequence diagram of message exchange using transport classes 4 and 5	197
Figure 54 – Sequence diagram of messages overwriting each other	198
Figure 55 – Sequence diagram of queued message exchange using transport classes 4 and 5	199
Figure 56 – Sequence diagram of retries using transport classes 4 and 5	199
Figure 57 – Sequence diagram of idle traffic using transport classes 4 and 5	200
Figure 58 – Classes 4 and 5 basic structure	201
Figure 59 – Class 6 basic structure	202
Figure 60 – Classes 4 to 6 general STD	203
Figure 61 – Class 4 sender STD	205
Figure 62 – Class 4 receiver STD	208
Figure 63 – Sequence diagram of three fragments using transport class 5	211
Figure 64 – Sequence diagram of fragmentation with retries using transport class 5	212
Figure 65 – Sequence diagram of two fragments using transport class 5	212
Figure 66 – Sequence diagram of aborted message using transport class 5	213
Figure 67 – Class 5 sender STD	214
Figure 68 – Class 5 receiver STD	217
Figure 69 – Data flow diagram for transport class 6	221
Figure 70 – Sequence diagram of message exchange using transport class 6	223
Figure 71 – Sequence diagram of retries using transport class 6	223
Figure 72 – Sequence diagram of idle traffic using transport class 6	224
Figure 73 – Sequence diagram of request overwriting null	225
Figure 74 – Sequence diagram of response overwriting ACK of null	225

Figure 75 – Sequence diagram of three fragments using transport class 6	226
Figure 76 – Sequence diagram of fragmentation with retries using transport class 6	227
Figure 77 – Sequence diagram of two fragments using transport class 6.....	227
Figure 78 – Sequence diagram of aborted fragmented sequence using transport class 6	228
Figure 79 – Class 6 client STD.....	229
Figure 80 – Class 6 server STD	232
Figure 81 – Data flow diagram for a link producer and consumer	237
Figure 82 – State transition diagram for a link producer	240
Figure 83 – State transition diagram for a link consumer.....	240
 Table 1 – Get_Attribute_All response service rules	 26
Table 2 – Example class level object/service specific response data of Get_Attribute_All	26
Table 3 – Example Get_Attribute_All data array method	26
Table 4 – Set_Attribute_All request service rules	27
Table 5 – Example Set_Attribute_All attribute ordering method	27
Table 6 – Example Set_Attribute_All data array method.....	28
Table 7 – State event matrix format	29
Table 8 – Example state event matrix	30
Table 9 – UCMM_PDU header format	33
Table 10 – UCMM command codes.....	33
Table 11 – Transport class 0 header	34
Table 12 – Transport class 1 header	34
Table 13 – Transport class 2 header	34
Table 14 – Transport class 3 header	35
Table 15 – Classes 4 to 6 header format.....	35
Table 16 – Real-time data header – exclusive owner	36
Table 17 – Real-time data header– redundant owner	36
Table 18 – Forward_Open request format	39
Table 19 – Forward_Open_Good response format	39
Table 20 – Forward_Open_Bad response format	40
Table 21 – Large_Forward_Open request format	41
Table 22 – Large_Forward_Open_Good response format	41
Table 23 – Large_Forward_Open_Bad response format.....	42
Table 24 – Forward_Close request format.....	43
Table 25 – Forward_Close_Good response format.....	43
Table 26 – Forward_Close_Bad response format	43
Table 27 – Unconnected_Send request format.....	44
Table 28 – Unconnected_Send_Good response format.....	45
Table 29 – Unconnected_Send_Bad response format	45
Table 30 – Get_Connection_Data request format.....	46
Table 31 – Get_Connection_Data response format	46
Table 32 – Search_Connection_Data request format	47

Table 33 – Get_Object_Owner request format	47
Table 34 – Forward_Open_Good response format	47
Table 35 – Time-out multiplier	50
Table 36 – Time tick units	51
Table 37 – Selection of connection ID	54
Table 38 – Transport class, trigger and Is_Server format	55
Table 39 – MR_Request_Header format	55
Table 40 – MR_Response_Header format	55
Table 41 – Structure of Get_Attribute_All_ResponsePDU body	56
Table 42 – Structure of Set_Attribute_All_RequestPDU body	56
Table 43 – Structure of Get_Attribute_List_RequestPDU body	56
Table 44 – Structure of Get_Attribute_List_ResponsePDU body	56
Table 45 – Structure of Set_Attribute_List_RequestPDU body	57
Table 46 – Structure of Set_Attribute_List_ResponsePDU body	57
Table 47 – Structure of Reset_RequestPDU body	57
Table 48 – Structure of Reset_ResponsePDU body	57
Table 49 – Structure of Start_RequestPDU body	57
Table 50 – Structure of Start_ResponsePDU body	58
Table 51 – Structure of Stop_RequestPDU body	58
Table 52 – Structure of Stop_ResponsePDU body	58
Table 53 – Structure of Create_RequestPDU body	58
Table 54 – Structure of Create_ResponsePDU body	58
Table 55 – Structure of Delete_RequestPDU body	59
Table 56 – Structure of Delete_ResponsePDU body	59
Table 57 – Structure of Get_Attribute_Single_ResponsePDU body	59
Table 58 – Structure of Set_Attribute_Single_RequestPDU body	59
Table 59 – Structure of Set_Attribute_Single_ResponsePDU body	59
Table 60 – Structure of Find_Next_Object_Instance_RequestPDU body	59
Table 61 – Structure of Find_Next_Object_Instance_ResponsePDU body	60
Table 62 – Structure of Apply_Attributes_RequestPDU body	60
Table 63 – Structure of Apply_Attributes_ResponsePDU body	60
Table 64 – Structure of Save_RequestPDU body	60
Table 65 – Structure of Save_ResponsePDU body	60
Table 66 – Structure of Restore_RequestPDU body	61
Table 67 – Structure of Restore_ResponsePDU body	61
Table 68 – Structure of Group_Sync_RequestPDU body	61
Table 69 – Structure of Group_Sync_ResponsePDU body	61
Table 70 – Identity object class attributes	61
Table 71 – Identity object instance attributes	62
Table 72 – Identity object bit definitions for status instance attribute	63
Table 73 – Bits 4 – 7 of status instance attribute	63
Table 74 – Class level object/service specific response data of Get_Attribute_All	64
Table 75 – Instance level object/service specific response data of Get_Attribute_All	64

Table 76 – Modified instance level object/service specific response data of Get_Attribute_All	65
Table 77 – Object-specific parameter for Reset.....	65
Table 78 – Message Router object class attributes	65
Table 79 – Message Router object instance attributes	66
Table 80 – Class level object/service specific response data of Get_Attribute_All	66
Table 81 – Instance level object/service specific response data of Get_Attribute_All.....	66
Table 82 – Assembly object class attributes.....	66
Table 83 – Assembly object instance attributes.....	67
Table 84 – Acknowledge Handler object class attributes	67
Table 85 – Acknowledge Handler object instance attributes	68
Table 86 – Structure of Add_AckData_Path_RequestPDU body.....	68
Table 87 – Structure of Remove_AckData_Path_RequestPDU body	68
Table 88 – Time Sync object instance attributes	69
Table 89 – Structure of Management_Message_RequestPDU body	71
Table 90 – Structure of Management_Message_ResponsePDU body.....	71
Table 91 – Management Message Command values.....	71
Table 92 – Connection Manager object class attributes.....	72
Table 93 – Connection Manager object instance attributes.....	72
Table 94 – Connection object class attributes	73
Table 95 – Connection object instance attributes	74
Table 96 – Values assigned to the state attribute.....	75
Table 97 – Values assigned to the instance_type attribute	75
Table 98 – Possible values within Direction Bit	76
Table 99 – Possible values within Production Trigger Bits.....	76
Table 100 – Possible values within Transport Class Bits.....	77
Table 101 – Transport Class_Trigger attribute	78
Table 102 – Values defined for the CP2/3_produced_connection_id attribute.....	78
Table 103 – Values defined for the CP2/3_consumed_connection_id attribute	79
Table 104 – Values for the Initial Production Characteristics nibble	80
Table 105 – Values for the Initial Consumption Characteristics nibble.....	81
Table 106 – Values for the watchdog_timeout_action.....	84
Table 107 – Structure of Connection_Bind_RequestPDU body.....	85
Table 108 – Object specific status for Connection_Bind service	86
Table 109 – Structure of Producing_Application_Lookup_RequestPDU body	86
Table 110 – Structure of Producing_Application_Lookup_ResponsePDU body.....	86
Table 111 – Producing_Application_Lookup Service status codes.....	86
Table 112 – Possible port segment examples	88
Table 113 – TCP/IP link address examples	89
Table 114 – Electronic key segment format.....	91
Table 115 – Logical segments examples.....	92
Table 116 – Network segments	92
Table 117 – Extended subtype definitions	94

Table 118 – Data segment	94
Table 119 – ANSI_Extended_Symbol segment	95
Table 120 – Addressing categories	95
Table 121 – Class code ID ranges	96
Table 122 – Attribute ID ranges	96
Table 123 – Service code ranges	96
Table 124 – Class codes	97
Table 125 – Reserved class attributes for all object class definitions	97
Table 126 – Common services list	98
Table 127 – Acknowledge Handler object specific services list	99
Table 128 – Time Sync object specific services list	99
Table 129 – Services specific to Connection Manager	99
Table 130 – Services specific to Connection object	99
Table 131 – Device type numbering	100
Table 132 – Connection Manager service request error codes	101
Table 133 – General status codes	106
Table 134 – Identity object status codes	108
Table 135 – Encapsulation header	115
Table 136 – Encapsulation command codes	116
Table 137 – Encapsulation status codes	117
Table 138 – Options flags	117
Table 139 – Nop request encapsulation header	117
Table 140 – RegisterSession request encapsulation header	118
Table 141 – RegisterSession request data portion	118
Table 142 – Options flags	118
Table 143 – RegisterSession reply encapsulation header	118
Table 144 – RegisterSession reply data portion	119
Table 145 – UnRegisterSession request encapsulation header	119
Table 146 – ListServices request encapsulation header	120
Table 147 – ListServices reply encapsulation header	120
Table 148 – ListServices reply data portion	120
Table 149 – Service type codes	121
Table 150 – Communications capability flags	121
Table 151 – ListIdentity request encapsulation header	121
Table 152 – ListIdentity reply encapsulation header	122
Table 153 – ListIdentity reply data portion	122
Table 154 – ListInterfaces request encapsulation header	123
Table 155 – ListInterfaces reply encapsulation header	123
Table 156 – SendRRData request encapsulation header	124
Table 157 – SendRRData request data portion	124
Table 158 – SendRRData reply encapsulation header	124
Table 159 – SendUnitData request encapsulation header	125
Table 160 – SendUnitData request data portion	125

Table 161 – Common packet format.....	125
Table 162 – Address and data item structure	126
Table 163 – Address type ID's	126
Table 164 – Data type ID's.....	126
Table 165 – Null address type.....	126
Table 166 – Connected address type	127
Table 167 – Sequenced address type	127
Table 168 – UCMM data type.....	127
Table 169 – Connected data type.....	127
Table 170 – Sockaddr info items	128
Table 171 – BOOLEAN encoding	129
Table 172 – Example compact encoding of a BOOL value	129
Table 173 – Encoding of SignedInteger values	129
Table 174 – Example compact encoding of a SignedInteger value	129
Table 175 – UnsignedInteger values	130
Table 176 – Example compact encoding of an UnsignedInteger	130
Table 177 – FixedLengthReal values	130
Table 178 – Example compact encoding of a REAL value	130
Table 179 – Example compact encoding of a LREAL value	130
Table 180 – FixedLengthReal values	131
Table 181 – STRING value	131
Table 182 – STRING2 value	131
Table 183 – STRINGN value.....	131
Table 184 – SHORT_STRING value.....	131
Table 185 – Example compact encoding of a STRING value	132
Table 186 – Example compact encoding of STRING2 value	132
Table 187 – SHORT_STRING type	132
Table 188 – Example compact encoding of a single dimensional ARRAY.....	134
Table 189 – Example compact encoding of a multi-dimensional ARRAY	134
Table 190 – Example compact encoding of a STRUCTURE	135
Table 191 – Identification codes and descriptions of elementary data types	136
Table 192 – Example 1 of formal encoding of a structure type specification	137
Table 193 – I/O Connection state event matrix.....	143
Table 194 – Bridged Connection state event matrix	147
Table 195 – Explicit Messaging Connection state event matrix.....	149
Table 196 – Primitives issued by FAL user to FSPM	152
Table 197 – Primitives issued by FAL user to FSPM	153
Table 198 – Primitives issued by FSPM to FAL user	155
Table 199 – Parameters used with primitives exchanged between FAL user and FSPM	156
Table 200 – Primitives issued by FSPM to ARPM	158
Table 201 – Primitives issued by ARPM to FSPM	158
Table 202 – Parameters used with primitives exchanged between FSPM and ARPM	159
Table 203 – UCMM client states	159

Table 204 – State event matrix of UCMM client.....	160
Table 205 – High-end UCMM server states	161
Table 206 – State event matrix of high-end UCMM server	163
Table 207 – Low-end UCMM server states	164
Table 208 – State event matrix of low-end UCMM server	164
Table 209 – Notification	167
Table 210 – Transport classes	168
Table 211 – Primitives issued by FSPM to ARPM	168
Table 212 – Primitives issued by ARPM to FSPM	169
Table 213 – Parameters used with primitives exchanged between FSPM and ARPM	169
Table 214 – Class 0 transport client states.....	171
Table 215 – Class 0 client SEM	171
Table 216 – Class 0 transport server states	172
Table 217 – Class 0 server SEM	172
Table 218 – Class 1 transport client states.....	175
Table 219 – Class 1 client SEM	176
Table 220 – Class 1 transport server states	177
Table 221 – Class 1 server SEM	178
Table 222 – Class 2 transport client states.....	182
Table 223 – Class 2 client SEM	183
Table 224 – Class 2 transport server states	184
Table 225 – Class 2 server SEM	185
Table 226 – Class 3 transport client states.....	190
Table 227 – Class 3 client SEM	191
Table 228 – Class 3 transport server states	193
Table 229 – Class 3 server SEM	195
Table 230 – Write and trigger events in class 4 and 5 transport	197
Table 231 – Common states for transport classes 4 to 6	202
Table 232 – Classes 4 to 6 general SEM	203
Table 233 – Class 4 transport sender states	205
Table 234 – Class 4 sender SEM	206
Table 235 – Class 4 transport receiver states	207
Table 236 – Class 4 receiver SEM	209
Table 237 – Class 5 transport sender states	214
Table 238 – Class 5 sender SEM	215
Table 239 – Class 5 transport receiver states	217
Table 240 – Class 5 receiver SEM	218
Table 241 – Class 6 transport client states.....	229
Table 242 – Class 6 client state event matrix	230
Table 243 – Class 6 transport server states	231
Table 244 – Class 6 server SEM	233
Table 245 – Primitives issued by ARPM to DMPM.....	238
Table 246 – Primitives issued by DMPM to ARPM.....	238

Table 247 – Parameters used with primitives exchanged between ARPM and DMPM	238
Table 248 – Primitives exchanged between data-link layer and DMPM.....	238
Table 249 – Parameters used with primitives exchanged between DMPM and Data-link	239
Table 250 – Link producer states	239
Table 251 – State event matrix of link producer	240
Table 252 – Link consumer states.....	240
Table 253 – State event matrix of link consumer	241
Table 254 – Network Connection ID selection	242
Table 255 – Example multicast assignments.....	246
Table 256 – UDP data format for class 0 and class 1	247

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 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.

NOTE Use of some of the associated protocol types is restricted by their 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 particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission from their respective intellectual-property-right holders.

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 first edition and its companion parts of the IEC 61158-6 subseries cancel and replace IEC 61158-6:2003. This edition of this part constitutes a technical revision. This part and its Type 2 companion parts also cancel and replace IEC/PAS 62413, published in 2005

This edition of IEC 61158-6 includes the following significant changes from the previous edition:

- a) deletion of the former Type 6 fieldbus for lack of market relevance;
- b) addition of new types of fieldbuses;

c) partition of part 6 of the third edition into multiple parts numbered -6-2, -6-3, ...

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/476/FDIS	65C/487/RVD

Full information on the voting for the approval of this 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.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result 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.

NOTE The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

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

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/TR 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 implementors 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

- 1) define the wire-representation of the service primitives defined in IEC 61158-5-2, and
- 2) 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) 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. It is this latter objective which gives rise to the diversity of protocols standardized in IEC 61158-6.

1.3 Conformance

This standard 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 referenced documents are indispensable for the application 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.

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

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

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

IEC 61588:2004¹, *Precision clock synchronization protocol for networked measurement and control systems*

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

IEC 62026-3², *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet*

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

ISO/IEC 8824, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*

ISO/IEC 8825, *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 Multiple-Octet Coded Character Set (UCS)*

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

ISO 11898:1993³, *Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication*

¹ Compliance with future editions of this standard will need checking.

² To be published.

³ A newer edition of this standard has been published, but only the cited edition applies.