

INTERNATIONAL STANDARD

IEC
61158-6

Third edition
2003-05

**Digital data communications
for measurement and control –
Fieldbus for use in industrial
control systems –**

**Part 6:
Application layer protocol specification**



Reference number
IEC 61158-6:2003(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**
- **Catalogue of IEC publications**

The on-line catalogue on the IEC web site (http://www.iec.ch/searchpub/cur_fut.htm) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.
- **IEC Just Published**

This summary of recently issued publications (http://www.iec.ch/online_news/justpub/jp_entry.htm) is also available by email. Please contact the Customer Service Centre (see below) for further information.
- **Customer Service Centre**

If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: custserv@iec.ch
Tel: +41 22 919 02 11
Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC
61158-6

Third edition
2003-05

Digital data communications for measurement and control – Fieldbus for use in industrial control systems –

Part 6: Application layer protocol specification

© IEC 2003 — Copyright - all rights reserved

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE **XW**

For price, see current catalogue

CONTENTS

FOREWORD.....	32
0 Introduction.....	34
1 Scope	36
2 Normative references.....	37
3 Terms and definitions.....	38
3.1 Summary.....	38
3.2 Terms and definitions from other ISO/IEC standards	38
3.3 Terms and definitions from IEC 61158-5.....	39
3.4 Other terms and definitions.....	39
3.5 Abbreviations and symbols.....	49
3.6 Conventions	53
3.7 Conventions used in state machines.....	61
4 Type 1	68
4.1 FAL syntax description	68
4.2 Transfer syntaxes.....	107
4.3 FAL protocol state machines structure.....	132
4.4 AP-Context state machine	134
4.5 FAL Service Protocol Machine (FSPM).....	151
4.6 Application Relationship Protocol Machines (ARPMs).....	157
4.7 DLL Mapping Protocol Machine (DMPM)	298
4.8 Protocol options	312
5 Type 2	316
5.1 Abstract syntax.....	316
5.2 Transfer syntax	390
5.3 Structure of FAL protocol state machines	402
5.4 Context state machine	402
5.5 FAL Service Protocol Machine (FSPM).....	402
5.6 Application Relationship Protocol Machines (ARPMs).....	408
5.7 DLL Mapping Protocol Machine (DMPM)	492
5.8 Alternate DLL mapping protocol machine.....	496
6 Type 3	502
6.1 FAL syntax description	502
6.2 Transfer syntax	506
6.3 FAL protocol state machines	553
6.4 AP-Context state machine	567
6.5 FAL Service Protocol Machines (FSPMs)	567
6.6 Application Relationship Protocol Machines (ARPMs).....	650
6.7 DLL Mapping Protocol Machines (DMPMs).....	791
6.8 Parameters for a DP-slave.....	827
7 Type 4	828
7.1 FAL syntax description	828
7.2 Transfer syntaxes.....	830
7.3 FAL Protocol State Machines.....	837
7.4 AP-Context State Machine.....	838
7.5 FAL Service Protocol Machine (FSPM).....	838

7.6	Application Relationship Protocol Machine (ARPM)	843
7.7	DLL Mapping Protocol Machine (DMPM)	846
7.8	Protocol options	849
8	Type 5	850
8.1	Overview	850
8.2	FAL Syntax Description	850
8.3	Transfer syntax	850
8.4	FAL Protocol State Machine Structure	899
8.5	SMK State Machine	899
8.6	VCR State Machine	914
8.7	FAL Service Protocol Machine (FSPM)	915
8.8	Application Relationship Protocol Machines (ARPMs)	916
8.9	DLL Mapping Protocol Machine (DMPM)	930
9	Type 6	936
9.1	Reference	936
9.2	Usage of Parameters	936
9.3	AP Context State Tables	936
9.4	Application Relationship protocol machines (ARPMs)	936
9.5	DLL Mapping protocol machine (DMPM)	952
10	Type 7	962
10.1	Abstract syntax of data type	962
10.2	Transfer Syntaxes	967
10.3	Structure of Protocol Machines	1027
10.4	AP-Context state machine	1028
10.5	Sub-MMS FAL Service Protocol Machine (FSPM)	1028
10.6	DLL Mapping Protocol Machine (DMPM) and Association Relationship Protocol Machine (ARPM)	1033
10.7	Protocol options	1081
11	Type 8	1101
11.1	FAL Syntax Description	1101
11.2	Transfer Syntax	1110
11.3	Protocol Machine Overview	1118
11.4	AP-Context State Machine	1119
11.5	FAL Service Protocol Machine (FSPM)	1119
11.6	Application Relationship Protocol Machines (ARPMs)	1119
11.7	DLL Mapping Protocol Machine	1128
12	Type 9	1139
12.1	FAL syntax description	1139
12.2	Transfer Syntax	1158
12.3	FAL Protocol State Machines Structure	1167
12.4	VCR PM State Machine	1170
12.5	AREP State Machines	1182
13	Type 10	1230
13.1	FAL Syntax Description	1230
13.2	Transfer Syntax	1247
13.3	FAL Protocol State Machines	1247

13.4 AP Context State Machine	1248
13.5 FAL Service Protocol Machines (FSPM)	1248
13.6 Application Relationship Protocol Machine (ARPM)	1303
13.7 DLL Mapping Protocol Machine (DMPM)	1306
13.8 Protocol Options	1309
Figure 1 – Relationship of IEC 61158-6 to other fieldbus layers and to users of the fieldbus Application service.....	34
Figure 2 – Attribute table format and terms	53
Figure 3 – Service request/response parameter	54
Figure 4 – Common structure of specific fields.....	58
Figure 5 – Example of an STD	63
Figure 6 – Example of an evaluation net	67
Figure 7 – APDU overview	108
Figure 8 – Identification information (format 1)	108
Figure 9 – Identification information (format 2).....	109
Figure 10 – Identification information (format 3).....	109
Figure 11 – Identification information (format 4).....	109
Figure 12 – Coding of the data type BinaryDate	111
Figure 13 – Encoding of Time of Day value.....	111
Figure 14 – Encoding of Time Difference value	112
Figure 15 – Encoding Time value	112
Figure 16 – Encoding of data of data type DL—Time-offset.....	113
Figure 17 – Structure of an object definition.....	114
Figure 18 – APDU overview	117
Figure 19 – Type field	117
Figure 20 – Identifier octet (context-specific)	118
Figure 21 – Identifier octet (FAL-specific)	118
Figure 22 – Length octet (one-octet format)	119
Figure 23 – Length octet (three-octet format).....	119
Figure 24 – APDU overview	127
Figure 25 – Relationships among protocol machines and adjacent layers	133
Figure 26 – AP to AP-Context initiation state machine	135
Figure 27 – State transition diagram of FSPM.....	153
Figure 28 – State transition diagram of the QUU ARPM	160
Figure 29 – State transition diagram of QUB-CO ARPM.....	169
Figure 30 – State transition diagram of the QUB-CL ARPM.....	181
Figure 31 – State transition diagram of QUB-seg ARPM	191
Figure 32 – State transition diagram of QUB-FC ARPM	211
Figure 33 – State transition diagram of BUB ARPM.....	238
Figure 34 – State transition diagram of BNB ARPM (basic state machine)	254
Figure 35 – State transition diagram of BNB ARPM (confirmed service sending and receiving – client)	255
Figure 36 – State transition diagram of BNB ARPM (confirmed service receiving and responding – server).....	255
Figure 37 – State transition diagram of BNB ARPM (unconfirmed service sending- client) ..	255
Figure 38 – State transition diagram of the BNU ARPM	279

Figure 39 – State transition diagram of the BNU-MP ARPM	291
Figure 40 – State transition diagram of DMPM	304
Figure 41 – Primitives exchanged between protocol machines	313
Figure 42 – Primitives exchanged between protocol machines	314
Figure 43 – Primitives exchanged between protocol machines	315
Figure 44 – Network connection parameters	333
Figure 45 – Time tick	335
Figure 46 – Connection establishment time-out	337
Figure 47 – Segment type	352
Figure 48 – Port segment	353
Figure 49 – Encapsulation message	377
Figure 50 – FixedLengthBitString compact encoding bit placement rules	394
Figure 51 – Example compact encoding of a OCTET FixedLengthBitString	394
Figure 52 – Example compact encoding of a WORD FixedLengthBitString	395
Figure 53 – Example compact encoding of a DWORD FixedLengthBitString	395
Figure 54 – Example compact encoding of a LWORD FixedLengthBitString	395
Figure 55 – Example 2 of formal encoding of a structure type specification	399
Figure 56 – Example of abbreviated encoding of a structure type specification	399
Figure 57 – Example 1 of formal encoding of an array type specification	400
Figure 58 – Example 2 of formal encoding of an array type specification	401
Figure 59 – Example 1 of abbreviated encoding of an array type specification	401
Figure 60 – Example 2 of abbreviated encoding of an array type specification	402
Figure 61 – State transition diagram of UCMM client	411
Figure 62 – State transition diagram of high-end UCMM server	413
Figure 63 – State transition diagram of low-end UCMM server	415
Figure 64 – Sequence diagram for a UCMM with one outstanding message	416
Figure 65 – Sequence diagram for a UCMM with multiple outstanding messages	417
Figure 66 – TPDU buffer	418
Figure 67 – Data flow diagram using a client transport class 0 and server transport class 0	420
Figure 68 – Sequence diagram of data transfer using transport class 0	421
Figure 69 – Class 0 client STD	422
Figure 70 – Class 0 server STD	423
Figure 71 – Data flow diagram using client transport class 1 and server transport class 1	424
Figure 72 – Sequence diagram of data transfer using client transport class 1 and server transport class 1	425
Figure 73 – Class 1 client STD	427
Figure 74 – Class 1 server STD	428
Figure 75 – Data flow diagram using client transport class 2 and server transport class 2	430
Figure 76 – Diagram of data transfer using client transport class 2 and server transport class 2 <i>without</i> returned data	431
Figure 77 – Sequence diagram of data transfer using client transport class 2 and server transport class 2 <i>with</i> returned data	432
Figure 78 – Class 2 client STD	434
Figure 79 – Class 2 server STD	436
Figure 80 – Data flow diagram using client transport class 3 and server transport class 3	439
Figure 81 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 <i>without</i> returned data	440

Figure 82 – Sequence diagram of data transfer using client transport class 3 and server transport class 3 <i>with</i> returned data	441
Figure 83 – Class 3 client STD	443
Figure 84 – Class 3 server STD	445
Figure 85 – Data flow diagram using transport classes 4 and 5	447
Figure 86 – Sequence diagram of message exchange using transport classes 4 and 5	448
Figure 87 – Sequence diagram of messages overwriting each other	449
Figure 88 – Sequence diagram of queued message exchange using transport classes 4 and 5	450
Figure 89 – Sequence diagram of retries using transport classes 4 and 5	451
Figure 90 – Sequence diagram of idle traffic using transport classes 4 and 5	452
Figure 91 – Classes 4 and 5 basic structure	453
Figure 92 – Class 6 basic structure	454
Figure 93 – Classes 4 to 6 general STD	455
Figure 94 – Class 4 sender STD	457
Figure 95 – Class 4 receiver STD	460
Figure 96 – Sequence diagram of three fragments using transport class 5	463
Figure 97 – Sequence diagram of fragmentation with retries using transport class 5	464
Figure 98 – Sequence diagram of two fragments using transport class 5	465
Figure 99 – Sequence diagram of aborted message using transport class 5	465
Figure 100 – Class 5 sender STD	467
Figure 101 – Class 5 receiver STD	470
Figure 102 – Data flow diagram for transport class 6	474
Figure 103 – Sequence diagram of message exchange using transport class 6	476
Figure 104 – Sequence diagram of retries using transport class 6	476
Figure 105 – Sequence diagram of idle traffic using transport class 6	477
Figure 106 – Sequence diagram of request overwriting null	478
Figure 107 – Sequence diagram of response overwriting ACK of null	479
Figure 108 – Sequence diagram of three fragments using transport class 6	480
Figure 109 – Sequence diagram of fragmentation with retries using transport class 6	481
Figure 110 – Sequence diagram of two fragments using transport class 6	482
Figure 111 – Sequence diagram of aborted fragmented sequence using transport class 6 ..	483
Figure 112 – Class 6 client STD	484
Figure 113 – Class 6 server STD	487
Figure 114 – Data flow diagram for a link producer and consumer	492
Figure 115 – State transition diagram for a link producer	495
Figure 116 – State transition diagram for a link consumer	496
Figure 117 – Coding of the data type BinaryDate	508
Figure 118 – Encoding of Time Of Day value	508
Figure 119 – Encoding of Time Difference value	508
Figure 120 – Encoding of Network Time value	509
Figure 121 – Encoding of Network Time Difference value	509
Figure 122 – Example Modul_Status_Array	514
Figure 123 – Example of Ext_Diag_Data in case of DPV1 diagnosis format with alarm and status PDU	549
Figure 124 – Example of Ext_Diag_Data in case of the basic diagnosis format	550
Figure 125 – Example of a special identifier format	550

Figure 126 – Example of a special identifier format with data types.....	551
Figure 127 – Example of a special identifier format with data types.....	551
Figure 128 – Example of a empty slot with data types.....	551
Figure 129 – Example for multi-variable device with AI and DO function blocks	552
Figure 130 – Identifiers (ID)	552
Figure 131 – Identifier list	552
Figure 132 – Structure of the Data_Unit for the request- and response-DLPDU	553
Figure 133 – Structuring of the protocol machines and adjacent layers in a DP-slave	556
Figure 134 – Structuring of the protocol machines and adjacent layers in a DP-master (class 1)	557
Figure 135 – Structuring of the protocol machines and adjacent layers in a DP-master (class 2)	558
Figure 136 – Sequence of the communication between DP-master and DP-slave	560
Figure 137 – Sequence of communication between DP-master (class 2) and DP-master (class 1)	562
Figure 138 – Sequence of acyclic communication between DP-master (class 1) and DP-slave	563
Figure 139 – Example for connection establishment on MS2.....	565
Figure 140 – Idle at master-side on MS2.....	566
Figure 141 – Idle at slave-side on MS2	567
Figure 142 – Example for connection establishment on MS2(server-side)	696
Figure 143 – Structure of RM entries in the RM_Registry.....	697
Figure 144 – APDU Header structure	830
Figure 145 – Instruction subfield of ControlStatus	830
Figure 146 – Errorcode subfield of ControlStatus	831
Figure 147 – Remaining subfields of ControlStatus	831
Figure 148 – DataFieldFormat encoding	832
Figure 149 – Structure of request APDU Body	832
Figure 150 – Structure of response APDU Body.....	832
Figure 151 – Variable Identifier	832
Figure 152 – Code subfield of Variable Identifier.....	833
Figure 153 – Summary of FAL Architecture.....	837
Figure 154 – FSPM proxy object state machine	839
Figure 155 – FSPM real object state machine	843
Figure 156 – ARPM State Machine	844
Figure 157 – DLPM State Machine.....	847
Figure 158 – State Transition Diagram for SMK	901
Figure 159 – State Transition Diagram of Client / Server ARPM.....	919
Figure 160 – State Transition diagram of the publisher / subscriber ARPM	926
Figure 161 – State transition diagram of DMPM	932
Figure 162 – State transition diagram of QUB-PC ARPM	940
Figure 163 – State transition diagram of the BNU-PC ARPM.....	950
Figure 164 – State transition diagram of DMPM	955
Figure 165 – Encoding of a CompactValue	967
Figure 166 – Organisation of the bits and octets within a PDU	968
Figure 167 – Encoding of a Bitstring	972
Figure 168 – Encoding of a Floating point.....	972

Figure 169 – Encoding of a structure	974
Figure 170 – Encoding of a boolean array.....	975
Figure 171 – Representation of a MCS PDU	981
Figure 172 – Relationships among Protocol Machines and Adjacent Layers.....	1027
Figure 173 – A_Readloc service evaluation net.....	1033
Figure 174 – A_Writeloc service evaluation net.....	1034
Figure 175 – A_Update service evaluation net	1035
Figure 176 – A_Readfar service evaluation net	1037
Figure 177 – A_writefar service evaluation net.....	1039
Figure 178 – A_Sent service evaluation net	1040
Figure 179 – A_Received service evaluation net.....	1040
Figure 180 – Association establishment: Requester element state machine	1047
Figure 181 – Association establishment: Responder element state machine	1048
Figure 182 – Association termination: Requester element state machine	1050
Figure 183 – Association termination: Responder element state machine	1052
Figure 184 – Association revocation: Requester element state machine	1053
Figure 185 – Association revocation: Acceptor element state machine.....	1054
Figure 186 – Interactions between state machine in an associated mode data transfer	1056
Figure 187 – Transfer service – Requester element state machine	1060
Figure 188 – Transfer service: Acceptor element state machine.....	1061
Figure 189 – Unacknowledged transfer: Requester element state machine	1062
Figure 190 – Unacknowledged transfer: Acceptor element state machine	1062
Figure 191 – Acknowledged transfer: Requester element state machine	1064
Figure 192 – Acknowledged transfer: Acceptor element state machine	1065
Figure 193 – Numbering mechanism state machine	1066
Figure 194 – Retry mechanism state machine.....	1068
Figure 195 – Anticipation mechanism state machine	1071
Figure 196 – Segmentation mechanism state machine.....	1073
Figure 197 – Reassembly mechanism state machine	1075
Figure 198 – Interaction of state machine in a non associated data transfer.....	1077
Figure 199 – Unacknowledged transfer: Requester element state machine	1078
Figure 200 – Unacknowledged transfer: Acceptor element state machine	1078
Figure 201 – Acknowledged transfer: Requester element state machine	1080
Figure 202 – Acknowledged transfer: Acceptor element state machine	1081
Figure 203 – APDU overview	1111
Figure 204 – APDU Header.....	1111
Figure 205 – PDU withType Extension	1111
Figure 206 – PDU with Address Extension	1111
Figure 207 – PDU with Type and Length Extension.....	1111
Figure 208 – Example of an Establish-Request PDU.....	1112
Figure 209 – Encoding of a PRIVATE tagged value	1112
Figure 210 – Encoding of a context specific tagged value	1113
Figure 211 – Identification information fields.....	1113
Figure 212 – ID-info for Tag 0 .. 14 , Length entry 0 .. 6.....	1113
Figure 214 – ID-info for Tag 15 .. 255 , Length entry 0 .. 6.....	1113
Figure 215 – ID-info for Tag 0 .. 14 , Length entry 7 .. 255.....	1113

Figure 216 – ID-info for Tag 15 .. 255 , Length entry 7 .. 255	1114
Figure 217 – Encoding of Boolean value TRUE	1114
Figure 218 – Encoding of Boolean value FALSE	1114
Figure 219 – Encoding of Strings	1114
Figure 220 – Encoding of BinaryDate Value	1115
Figure 221 – Encoding of BinaryDate2000 Value	1115
Figure 222 – Encoding of Time of Day value	1116
Figure 223 – Encoding of Time Difference Value	1116
Figure 224 – Encoding of Time Value	1117
Figure 225 – Example for an Object Definition	1118
Figure 226 – Primitives Exchanged between Protocol Machines	1119
Figure 227 – State Transition Diagram of QUB-TM AREP	1126
Figure 228 – State transition diagram of Type-8 DMPM	1131
Figure 229 – Structure of an object description in the OD	1154
Figure 230 – Structure of the OD object description	1154
Figure 231 – Structure of a domain entry in the S-OD	1154
Figure 232 – Object description of program invocation in the DP-OD	1154
Figure 233 – Object description of simple variable in the S-OD	1154
Figure 234 – Object description of array in the S-OD	1154
Figure 235 – Object description of record in the S-OD	1155
Figure 236 – Object description of variable list in the DV-OD	1155
Figure 237 – Object description of data type in the ST-OD	1155
Figure 238 – Object description of data type structure description in the ST-OD	1155
Figure 239 – Object description of event in the S-OD	1155
Figure 240 – Insertion of identification information in the FMS PDU	1158
Figure 241 – Coding of the ID info with and without extension	1160
Figure 242 – Coding with identification	1160
Figure 243 – Coding without identification	1160
Figure 244 – Representation of the value true	1160
Figure 245 – Representation of the value false	1160
Figure 246 – Coding of data of data type Integer16	1161
Figure 247 – Coding of data of data type Unsigned16	1161
Figure 248 – Coding of data of data type Floating Point	1162
Figure 249 – Coding of data of data type Visible String	1162
Figure 250 – Coding of data of data type Octet String	1162
Figure 251 – Coding of data of data type Date	1163
Figure 252 – Coding of data of data type Time Of Day	1164
Figure 253 – Coding of data of data type Time Difference	1164
Figure 254 – Coding of data of data type Bit String	1165
Figure 255 – Coding of data of data type Time Value	1165
Figure 256 – Coding of data of user data definitions with identifier	1165
Figure 257 – Coding of data of user data definitions without identifier	1166
Figure 258 – Coding of ID info for a SEQUENCE	1166
Figure 259 – Relationships among protocol machines and adjacent layers	1168
Figure 260 – VCR state machine	1172
Figure 261 – State transition diagram of FSPM	

Figure 262 – State transition diagram of the QUU ARPM	1196
Figure 263 – State transition diagram of QUB ARPM	1199
Figure 264 – State transition diagram of the BNU ARPM.....	1208
Figure 265 – State transition diagram of DMPM	1213
Figure 266 - Error Message structure.....	1230
Figure 267 – Coding scheme of ITEMQUALITYDEF	1233
Figure 268 – Relationship among Protocol Machines	1248
Figure 269 – State Transition Diagram of FSPM	1258
Figure 270 – State Transition Diagram of ARPM	1304
Figure 271 – State Transition Diagram of DMPM	1308
Table 1– Get_Attribute_All response service rules	55
Table 2 – Example class level object/service specific reply data of Get_Attribute_All.....	55
Table 3 – Example Get_Attribute_All data array method	55
Table 4 – Set_Attribute_All request service rules	56
Table 5 – Example Set_Attribute_All attribute ordering method	56
Table 6 – Example Set_Attribute_All data array method.....	57
Table 7 – Conventions used for state machines	61
Table 8 – State event matrix format	63
Table 9 – Example state event matrix	64
Table 10 – State machine description elements	64
Table 11 – Description of state machine elements	65
Table 12 – Conventions used in state machines.....	65
Table 13 – Reason codes	106
Table 14 – FAL header	107
Table 15 – Identification octet classes	128
Table 16 – Unit of measurement of contents length octets	129
Table 17 – Primitives issued by FAL-user to AP-Context.....	134
Table 18 – Primitives issued by AP-Context to FAL-user.....	134
Table 19 – AP-Context state machine sender transactions.....	136
Table 20 – AP-Context state machine receiver transactions.....	140
Table 21 – Function ResetArep.....	147
Table 22 – Function ApContextTest	148
Table 23 – Function ServicesSupportedTest.....	148
Table 24 – Function ApExplicitConnection	148
Table 25 – Function ImmediateAcknowledge	148
Table 26 – Function ConfirmedServiceCheck.....	148
Table 27 – Function UnconfirmedServiceCheck	148
Table 28 – Function ArServiceCheck	148
Table 29 – Function ArFspmService	149
Table 30 – Function ArAcceeSupported	149
Table 31 – Function MaxFalPduLengthTest	149
Table 32 – Function NegotiateOutstandingServices	149
Table 33 – Function RequestedServicesSupportedTest	149
Table 34 – Function IndicatedServicesSupportedTest.....	150
Table 35 – Function InvokeldExistent	150

Table 36 – Function SameService.....	150
Table 37 – Primitives issued by AP-Context to FSPM	151
Table 38 – Primitives issued by FSPM to AP-Context	152
Table 39 – FSPM state table – Sender transactions.....	153
Table 40 – FSPM state table – receiver transactions.....	155
Table 41 – Function SelectArep	156
Table 42– Primitives issued by FSPM to ARPM	157
Table 43 – Primitives issued by ARPM to FSPM	157
Table 44 – Parameters used with primitives exchanged between FSPM and ARPM	157
Table 45 – QUU ARPM states.....	160
Table 46 – QUU ARPM state table – sender transactions.....	161
Table 47 – QUU ARPM state table – receiver transactions.....	162
Table 48 – Function GetArepId ().....	163
Table 49 – Function BuildFAL-PDU.....	163
Table 50 – Function FAL_Pdu_Type	163
Table 51 – Primitives issued by FSPM to ARPM	163
Table 52 – Primitives issued by ARPM to FSPM	164
Table 53 – Parameters used with primitives exchanged between FSPM and ARPM	164
Table 54 – QUB-CO ARPM states.....	169
Table 55 – QUB-CO ARPM state table – sender transactions	169
Table 56 – QUB-CO ARPM state table – receiver transactions.....	171
Table 57 – Function GetArepId ().....	177
Table 58 – Function BuildFAL-PDU.....	177
Table 59 – Function FAL_Pdu_Type	177
Table 60 – Function AbortIdentifier	177
Table 61 – Function AbortReason	177
Table 62 – Function AbortDetail.....	178
Table 63 – Primitives issued by FSPM to ARPM	178
Table 64 – Primitives issued by ARPM to FSPM	178
Table 65 – Parameters used with primitives exchanged between FSPM and ARPM	179
Table 66 – QUB-CL ARPM states	181
Table 67 – QUB-CL ARPM state table – sender transactions	182
Table 68 – QUB-CL ARPM state table – receiver transactions	183
Table 69 – Function GetArepId ().....	184
Table 70 – Function BuildFAL-PDU.....	185
Table 71 – Function FAL-Pdu_Type	185
Table 72 – Primitives issued by FSPM to ARPM	185
Table 73 – Primitives issued by ARPM to FSPM	186
Table 74 – Parameters used with primitives exchanged between FSPM and ARPM	186
Table 75 – QUB-seg ARPM states	191
Table 76 – QUB-seg ARPM state table – sender transactions	192
Table 77 – ARPM state table – receiver transactions	195
Table 78 – Function GetArepId	203
Table 79 – Function BuildFAL-PDU.....	203
Table 80 – Function FAL_Pdu_Type	204
Table 81 – Function AbortIdentifier	204

Table 82 – Function AbortReason	204
Table 83 – Function AbortDetail	204
Table 84 – Function BuildFAL-segment	204
Table 85 – Function MoreFollows	204
Table 86 – Function ResetIntermediatePDU	205
Table 87 – Function AddSegment	205
Table 88 – Function GetIntermediatePDU	205
Table 89 – Primitives issued by FSPM to ARPM	205
Table 90 – Primitives issued by ARPM to FSPM	206
Table 91 – Parameters used with primitives exchanged between FSPM and ARPM	206
Table 92 – QUB-FC ARPM states	210
Table 93 – QUB-FC ARPM state table – sender transactions	211
Table 94 – QUB-FC ARPM state table – receiver transactions	217
Table 95 – Function GetArepld ()	229
Table 96 – Function BuildFAL-PDU	229
Table 97 – Function FAL_Pdu_Type	229
Table 98 – Function AREPContextCheck()	230
Table 99 – Function AbortIdentifier	230
Table 100 – Function AbortReason	230
Table 101 – Function AbortDetail	230
Table 102 – Function StartTimer	230
Table 103 – Function StopTimer	231
Table 104 – Function ResetCounters	231
Table 105 – Function IncrementCounter	231
Table 106 – Function DecrementCounter	231
Table 107 – Function GetCounterValue	231
Table 108 – Function Xon	231
Table 109 – Function SetXonOff	231
Table 110 – Primitives issued by FSPM to ARPM	232
Table 111 – Primitives issued by ARPM to FSPM	233
Table 112 – Parameters used with primitives exchanged between FSPM and ARPM	233
Table 113 – BUB ARPM states	237
Table 114 – BUB ARPM state table – sender transactions	239
Table 115 – BUB state table – receiver transactions	241
Table 116 – Function GetArepld ()	248
Table 117 – Function BuildFAL-PDU	248
Table 118 – Function FAL_Pdu_Type	248
Table 119 – Function AbortIdentifier	248
Table 120 – Function AbortReason	248
Table 121 – Function AbortDetail	248
Table 122 – Primitives issued by FSPM to ARPM	249
Table 123 – Primitives issued by ARPM to FSPM	249
Table 124 – Parameters used with primitives exchanged between FSPM and ARPM	250
Table 125 – BNB ARPM states	254
Table 126 – BNB ARPM state table – sender transactions (basic state machine)	256
Table 127 – BNB state table – receiver transactions (basic state machine)	256

Table 128 – BNB ARPM state table – sender transactions (confirmed service sending and receiving – client).....	265
Table 129 – BNB state table – receiver transactions (confirmed service sending and receiving – client).....	267
Table 130 – BNB ARPM state table – sender transactions (confirmed service receiving and responding – server)	269
Table 131 – BNB state table – receiver transactions (confirmed service receiving and responding – server)	270
Table 132 – BNB ARPM state table – sender transactions (unconfirmed service sending – client)	271
Table 133 – BNB state table – receiver transactions (unconfirmed service sending – client)	271
Table 134 – Function GetArepld ()	271
Table 135 – Function dmpm_duplicate_dlsdu ()	272
Table 136 – Function BuildFAL-PDU ()	272
Table 137 – Function FAL_Pdu_Type ()	272
Table 138 – Function AbortIdentifier ()	272
Table 139 – Function AbortReason ()	272
Table 140 – Function AbortDetail ()	272
Table 141 – Function StartTimer ()	273
Table 142 – Function StopTimer ()	273
Table 143 – Function StopService ()	273
Table 144 – Function RestoreService ()	273
Table 145 – Function Flag ()	273
Table 146 – Function SetFlag ()	273
Table 147 – Function StoreIndex ()	274
Table 148 – Function TestIndex ()	274
Table 149 – Primitives issued by FSPM to ARPM	274
Table 150 – Primitives issued by ARPM to FSPM	275
Table 151 – Parameters used with primitives exchanged between FSPM and ARPM	275
Table 152 – BNU ARPM states	279
Table 153 – BNU ARPM state table – sender transactions	280
Table 154 – BNU ARPM state table – receiver transactions	281
Table 155 – Function GetArepld ()	284
Table 156 – Function BuildFAL-PDU	284
Table 157 – Function FAL_Pdu_Type	284
Table 158 – Function AbortIdentifier	284
Table 159 – Function AbortReason	284
Table 160 – Function AbortDetail	285
Table 161 – Primitives issued by FSPM to ARPM	285
Table 162 – Primitives issued by ARPM to FSPM	286
Table 163 – Parameters used with primitives exchanged between FSPM and ARPM	286
Table 164 – BNU-MP ARPM states	290
Table 165 – BNU-MP ARPM state table – sender transactions	291
Table 166 – BNU-MP ARPM state table – receiver transactions	293
Table 167 – Function GetArepld ()	297
Table 168 – Function BuildFAL-PDU	297

Table 169 – Function FAL_Pdu_Type	297
Table 170 – Function AbortIdentifier	297
Table 171 – Function AbortReason	298
Table 172 – Function AbortDetail	298
Table 173 – Function dedicatedArep ()	298
Table 174 – Primitives issued by ARPM to DMPM	299
Table 175 – Primitives issued by DMPM to ARPM	300
Table 176 – Parameters used with primitives exchanged between ARPM and DMPM	301
Table 177 – Primitives exchanged between Data Link Layer and DMPM	302
Table 178 – DMPM state descriptions	304
Table 179 – DMPM state table – sender transactions	304
Table 180 – DMPM state table – receiver transactions	307
Table 181 – Function PickArep	311
Table 182 – Function FindAREP	311
Table 183 – Function LocateArep	311
Table 184 – UCMM_PDU header format	318
Table 185 – UCMM command codes	318
Table 186 – Transport class 0 header	319
Table 187 – Transport class 1 header	319
Table 188 – Transport class 2 header	320
Table 189 – Transport class 3 header	320
Table 190 – Classes 4 to 6 header format	320
Table 191 – Real-time data header – exclusive owner	321
Table 192 – Real-time data header– redundant owner	322
Table 193 – Forward_Open request format	324
Table 194 – Forward_Open_Good response format	324
Table 195 – Forward_Open_Bad response format	325
Table 196 – Ex_Forward_Open request format	326
Table 197 – Ex_Forward_Open_Good response format	326
Table 198 – Ex_Forward_Open_Bad response format	327
Table 199 – Forward_Close request format	327
Table 200 – Forward_Close_Good response format	328
Table 201 – Forward_Close_Bad response format	328
Table 202 – Unconnected_Send request format	329
Table 203 – Unconnected_Send_Good response format	329
Table 204 – Unconnected_Send_Bad response format	330
Table 205 – Get_Connection_Data request format	330
Table 206 – Get_Connection_Data response format	331
Table 207 – Search_Connection_Data request format	331
Table 208 – Get_Object_Owner request format	332
Table 209 – Forward_Open_Good response format	332
Table 210 – Time-out multiplier	335
Table 211 – Time tick units	336
Table 212 – Selection of connection ID	339
Table 213 – Transport class, trigger and Is_Server format	340
Table 214 – MR_Request_Header format	

Table 215 – MR_Response_Header format.....	340
Table 216 – Structure of Get_Attribute_All_ResponsePDU body.....	341
Table 217 – Structure of Set_Attribute_All_RequestPDU body.....	341
Table 218 – Structure of Get_Attribute_List_RequestPDU body.....	341
Table 219 – Structure of Get_Attribute_List_ResponsePDU body.....	341
Table 220 – Structure of Set_Attribute_List_RequestPDU body.....	342
Table 221 – Structure of Set_Attribute_List_ResponsePDU body.....	342
Table 222 – Structure of Reset_RequestPDU body.....	342
Table 223 – Structure of Reset_ResponsePDU body.....	342
Table 224 – Structure of Start_RequestPDU body.....	342
Table 225 – Structure of Start_ResponsePDU body.....	342
Table 226 – Structure of Stop_RequestPDU body.....	343
Table 227 – Structure of Stop_ResponsePDU body.....	343
Table 228 – Structure of Create_RequestPDU body.....	343
Table 229 – Structure of Create_ResponsePDU body.....	343
Table 230 – Structure of Delete_RequestPDU body.....	343
Table 231 – Structure of Delete_ResponsePDU body.....	343
Table 232 – Structure of Get_Attribute_Single_ResponsePDU body.....	344
Table 233 – Structure of Set_Attribute_Single_RequestPDU body.....	344
Table 234 – Structure of Set_Attribute_Single_ResponsePDU body.....	344
Table 235 – Structure of Find_Next_Object_Instance_RequestPDU body.....	344
Table 236 – Structure of Find_Next_Object_Instance_ResponsePDU body.....	344
Table 237 – Structure of Apply_Attributes_RequestPDU body.....	345
Table 238 – Structure of Apply_Attributes_ResponsePDU body.....	345
Table 239 – Structure of Save_RequestPDU body.....	345
Table 240 – Structure of Save_ResponsePDU body.....	345
Table 241 – Structure of Restore_RequestPDU body.....	345
Table 242 – Structure of Restore_ResponsePDU body.....	345
Table 243 – Identity object class attributes.....	346
Table 244 – Identity object instance attributes.....	346
Table 245 – Identity object bit definitions for status instance attribute.....	346
Table 246 – Bits 4 – 7 of status instance attribute.....	347
Table 247 – Class level object/service specific response data of Get_Attribute_All.....	347
Table 248 – Instance level object/service specific response data of Get_Attribute_All.....	348
Table 249 – Modified instance level object/service specific response data of Get_Attribute_All.....	348
Table 250 – Object-specific parameter for Reset.....	348
Table 251 – Message router object class attributes.....	349
Table 252 – Message router object instance attributes.....	349
Table 253 – Class level object/service specific response data of Get_Attribute_All.....	349
Table 254 – Instance level object/service specific response data of Get_Attribute_All.....	349
Table 255 – Assembly object class attributes.....	350
Table 256 – Assembly object instance attributes.....	350
Table 257 – Connection manager object class attributes.....	350
Table 258 – Connection manager object instance attributes.....	351
Table 259 – Possible port segment examples.....	353

Table 260 – TCP/IP link address examples	354
Table 261 – Logical segments	355
Table 262 – Electronic key segment format.....	356
Table 263 – Network segments.....	357
Table 264 – Data segment	358
Table 265 – ANSI_Extended_Symbol segment	358
Table 266 – Addressing categories	359
Table 267 – Class code ID ranges	359
Table 268 – Attribute ID ranges	359
Table 269 – Service code ranges	360
Table 270 – objects	360
Table 271 – Reserved class attributes for all object class definitions	361
Table 272 – Common services list	361
Table 273 – Services specific to Connection Manager	362
Table 274 – Device type numbering	362
Table 275 – Connection manager service request error codes	363
Table 276 – General status codes.....	371
Table 277 – Extended status codes	373
Table 278 – Encapsulation header	378
Table 279 – Encapsulation command codes	378
Table 280 – Encapsulation status codes	379
Table 281 – Options flags	379
Table 282 – Nop request encapsulation header	380
Table 283 – RegisterSession request encapsulation header	380
Table 284 – RegisterSession request data portion	380
Table 285 – Options flags	380
Table 286 – RegisterSession reply encapsulation header	381
Table 287 – RegisterSession reply data portion	381
Table 288 – UnRegisterSession request encapsulation header	382
Table 289 – ListServices request encapsulation header.....	382
Table 290 – ListServices reply encapsulation header.....	382
Table 291 – ListServices reply data portion.....	383
Table 292 – Service type codes	383
Table 293 – Communications capability flags.....	383
Table 294 – ListIdentity request encapsulation header.....	384
Table 295 – ListIdentity reply encapsulation header.....	384
Table 296 – ListIdentity reply data portion.....	385
Table 297 – ListInterfaces request encapsulation header.....	385
Table 298 – ListInterfaces reply encapsulation header.....	385
Table 299 – SendRRData request encapsulation header	386
Table 300 – SendRRData request data portion	386
Table 301 – SendRRData reply encapsulation header	387
Table 302 – SendUnitData request encapsulation header	387
Table 303 – SendUnitData request data portion.....	387
Table 304 – Common packet format.....	388
Table 305 – Address and data item structure	388

Table 306 – Address type ID's	388
Table 307 – Data type ID's.....	388
Table 308 – Null address type.....	388
Table 309 – Connected address type	389
Table 310 – Sequenced address type	389
Table 311 – UCMM data type.....	389
Table 312 – Connected data type	389
Table 313 – Sockaddr info items	390
Table 314 – BOOLEAN encoding	391
Table 315 – Example compact encoding of a BOOL value	391
Table 316 – Encoding of SignedInteger values	391
Table 317 – Example compact encoding of a SignedInteger value	391
Table 318 – UnsignedInteger values	392
Table 319 – Example compact encoding of an UnsignedInteger	392
Table 320 – FixedLengthReal values	392
Table 321 – Example compact encoding of a REAL value	392
Table 322 – Example compact encoding of a LREAL value	392
Table 323 – FixedLengthReal values	393
Table 324 – STRING value	393
Table 325 – STRING2 value	393
Table 326 – STRINGN value.....	393
Table 327 – SHORT_STRING value	393
Table 328 – Example compact encoding of a STRING value	394
Table 329 – Example compact encoding of STRING2 value	394
Table 330 – SHORT_STRING type	394
Table 331 – Example compact encoding of a single dimensional ARRAY.....	395
Table 332 – Example compact encoding of a multi-dimensional ARRAY	396
Table 333 – Example compact encoding of a STRUCTURE	397
Table 334 – Identification codes and descriptions of elementary data types	398
Table 335 – Example 1 of formal encoding of a structure type specification	399
Table 336 – Primitives issued by FAL user to FSPM	403
Table 337 – Primitives issued by FAL user to FSPM	405
Table 338 – Primitives issued by FSPM to FAL user	407
Table 339 – Parameters used with primitives exchanged between FAL user and FSPM	408
Table 340 – Primitives issued by FSPM to ARPM	409
Table 341 – Primitives issued by ARPM to FSPM	410
Table 342 – Parameters used with primitives exchanged between FSPM and ARPM	410
Table 343 – UCMM client states	411
Table 344 – State event matrix of UCMM client.....	412
Table 345 – High-end UCMM server states.....	413
Table 346 – State event matrix of high-end UCMM server.....	414
Table 347 – Low-end UCMM server states.....	415
Table 348 – State event matrix of low-end UCMM server	415
Table 349 – Notification	418
Table 350 – Transport classes	419
Table 351 – Primitives issued by FSPM to ARPM	

Table 352 – Primitives issued by ARPM to FSPM	419
Table 353 – Parameters used with primitives exchanged between FSPM and ARPM	420
Table 354 – Class 0 transport client states	421
Table 355 – Class 0 client SEM	422
Table 356 – Class 0 transport server states	423
Table 357 – Class 0 server SEM	423
Table 358 – Class 1 transport client states	426
Table 359 – Class 1 client SEM	427
Table 360 – Class 1 transport server states	428
Table 361 – Class 1 server SEM	429
Table 362 – Class 2 transport client states	433
Table 363 – Class 2 client SEM	434
Table 364 – Class 2 transport server states	435
Table 365 – Class 2 server SEM	437
Table 366 – Class 3 transport client states	442
Table 367 – Class 3 client SEM	443
Table 368 – Class 3 transport server states	444
Table 369 – Class 3 server SEM	446
Table 370 – Write and trigger events in class 4 and 5 transport	448
Table 371 – Common states for transport classes 4 to 6	454
Table 372 – Classes 4 to 6 general SEM	455
Table 373 – Class 4 transport sender states	457
Table 374 – Class 4 sender SEM	458
Table 375 – Class 4 transport receiver states	459
Table 376 – Class 4 receiver SEM	461
Table 377 – Class 5 transport sender states	466
Table 378 – Class 5 sender SEM	468
Table 379 – Class 5 transport receiver states	470
Table 380 – Class 5 receiver SEM	471
Table 381 – Class 6 transport client states	484
Table 382 – Class 6 client state event matrix	485
Table 383 – Class 6 transport server states	486
Table 384 – Class 6 server SEM	489
Table 385 – Primitives issued by ARPM to DMPM	493
Table 386 – Primitives issued by DMPM to ARPM	493
Table 387 – Parameters used with primitives exchanged between ARPM and DMPM	493
Table 388 – Primitives exchanged between Data Link Layer and DMPM	494
Table 389 – Parameters used with primitives exchanged between DMPM and Data Link	494
Table 390 – Link producer states	495
Table 391 – State event matrix of link producer	495
Table 392 – Link consumer states	495
Table 393 – State event matrix of link consumer	496
Table 394 – UDP data format for class 0 and class 1	499
Table 395 – APDU syntax	502
Table 396 – Substitutions	504
Table 397 – Alarm_Type range	

Table 398 – Status_Type value range	512
Table 399 – Alarm_Specifier	513
Table 400 – Range of Modul_Status_Entry (1-4)	514
Table 401 – Error type	516
Table 402 – Specification of the bits Lock_Req and Unlock_Req	519
Table 403 – Range of Length_of_Manufacturer_Specific_Data if used in Chk_Cfg-REQ-PDU	525
Table 404 – Range of Length_of_Manufacturer_Specific_Data if used in Get_Cfg-RES-PDU	526
Table 405 – Values (codes) for data types	527
Table 406 – Specification of the bits for Un-/Sync and Un-/Freeze	528
Table 407 – Coding of the Function_Code/ Function_Num	530
Table 408 – Coding of the Error_Code / Function_Num	531
Table 409 – Values of Error_Decode	531
Table 410 – Coding of Error_Code_1 at DPV1	532
Table 411 – Values of MDiag_Identifier	533
Table 412 – Values for Area_Code_UpDownload	535
Table 413 – Values for Area_CodeActBrct	535
Table 414 – Values for Area_CodeAct	536
Table 415 – Values for Data_rate	537
Table 416 – Values for Slave_Type	539
Table 417 – Values for Alarm_Mode	540
Table 418 – Values for Subnet	545
Table 419 – Values of reason code if instance is DLL	545
Table 420 – Values of reason code if instance is MS2	546
Table 421 – Values of Extended_Function_Num	547
Table 422 – Values of FI_State	548
Table 423 – Assignment of state machines	555
Table 424 – Primitives issued by AP-Context to FSPMS	568
Table 425 – Primitives issued by FSPMS to AP-Context	570
Table 426 – FSPMS state table	576
Table 427 – Functions used by the FSPMS	598
Table 428 – Primitives issued by AP-Context to FSPMM1	599
Table 429 – Primitives issued by FSPMM1 to AP-Context	602
Table 430 – FSPMM1 state table	608
Table 431 – Functions used by the FSPMM1	633
Table 432 – Primitives issued by AP-Context to FSPMM2	633
Table 433 – Primitives issued by FSPMM2 to AP-Context	635
Table 434 – FSPMM2 state table	638
Table 435 – Functions used by the FSPMM2	649
Table 436 – Primitives issued by FSPMS to MSCY1S	650
Table 437 – Primitives issued by MSCY1S to FSPMS	651
Table 438 – Rules for DPV1_Status_1, DPV1_Status_2 and DPV1_Status_3 check	652
Table 439 – MSCY1S state table	657
Table 440 – Functions used by the MSCY1S	677
Table 441 – Primitives issued by FSPMS to MSAC1S	679

Table 442 – Primitives issued by MSAC1S to FSPMS	680
Table 443 – Primitives issued by MSCY1S to MSAC1S	680
Table 444 – Primitives issued by MSAC1S to MSCY1S	680
Table 445 – Parameter used with primitives exchanged between MSAC1S and MSCY1S ...	680
Table 446 – MSAC1S state table	682
Table 447 – Functions used by the MSAC1S	691
Table 448 – Primitives issued by FSPMS to SSCY1S	692
Table 449 – Primitives issued by SSCY1S to FSPMS	692
Table 450 – SSCY1S state table	693
Table 451 – Functions used by the SSCY1S	694
Table 452 – Primitives issued by FSPMS to MSRM2S	695
Table 453 – Primitives issued by MSRM2S to FSPMS	695
Table 454 – MSRM2S state table	698
Table 455 – Primitives issued by FSPMS to MSAC2S	700
Table 456 – Primitives issued by MSAC2S to FSPMS	701
Table 457 – Primitives issued by MSRM2S to MSAC2S	701
Table 458 – Primitives issued by MSAC2S to MSRM2S	702
Table 459 – Parameter used with primitives exchanged with MSAC2S	702
Table 460 – MSAC2S state table	704
Table 461 – Primitives issued by MSCS1S to FSPMS	716
Table 462 – MSCS1S state table	717
Table 463 – Primitives issued by FSPMM1 to MSCY1M	718
Table 464 – Primitives issued by MSCY1M to FSPMM1	718
Table 465 – Parameters used with primitives exchanged between FSPMM1 and MSCY1M	719
Table 466 – MSCY1M state table	722
Table 467 – Primitives issued by FSPMM1 to MSAL1M	736
Table 468 – Primitives issued by MSAL1M to FSPMM1	736
Table 469 – Primitives issued by MSCY1M to MSAL1M	736
Table 470 – Primitives issued by MSAL1M to MSCY1M	736
Table 471 – Parameter used with primitives exchanged between MSAL1M and MSCY1M	737
Table 472 – Possible values in the Alarm_State_Table	738
Table 473 – MSAL1M state table	740
Table 474 – Primitives issued by FSPMM1 to MSAC1M	744
Table 475 – Primitives issued by MSAC1M to FSPMM1	745
Table 476 – Primitives issued by MSAL1M to MSAC1M	745
Table 477 – Primitives issued by MSAC1M to MSAL1M	745
Table 478 – Parameter used with primitives exchanged between MSAL1M and MSCY1M	746
Table 479 – MSAC1M state table	751
Table 480 – Primitives issued by FSPMM1 to MMAC1	757
Table 481 – Primitives issued by MMAC1 to FSPMM1	757
Table 482 – MMAC1 state table	759
Table 483 – Primitives issued by FSPMM1 to MSCS1M	763
Table 484 – Primitives issued by MSCS1M to FSPMM1	764
Table 485 – MSCS1M state table	765

Table 486 – Primitives issued by FSPMM2 to MSAC2M	768
Table 487 – Primitives issued by MSAC2M to FSPMM2	769
Table 488 – Parameters used with primitives exchanged with MSAC2M	769
Table 489 – MSAC2M state table	774
Table 490 – Primitives issued by FSPMM2 to MMAC2	784
Table 491 – Primitives issued by MMAC2 to FSPMM2	785
Table 492 – Parameters used with primitives exchanged with MMAC2	785
Table 493 – MMAC2 state table	787
Table 494 – Primitives issued by FSPMS to DMPMS	791
Table 495 – Primitives issued by DMPMS to FSPMS	791
Table 496 – Primitives issued by MSCY1S to DMPMS	792
Table 497 – Primitives issued by DMPMS to MSCY1S	792
Table 498 – Primitives issued by DMPMS to SSCY1S	793
Table 499 – Primitives issued by MSAC1S, MSRM2S, MSAC2S to DMPMS	793
Table 500 – Primitives issued by DMPMS to MSAC1S, MSRM2S, MSAC2S	793
Table 501 – Primitives issued by DMPMS to MSCS1S	793
Table 502 – Primitives issued by DMPMS to DL	794
Table 503 – Primitives issued by DL to DMPMS	794
Table 504 – Parameters used with primitives exchanged with DMPMS	795
Table 505 – DMPMS state table	796
Table 506 – Functions used by the DMPMS	802
Table 507 – Primitives issued by FSPMM1 to DMPMM1	803
Table 508 – Primitives issued by DMPMM1 to FSPMM1	803
Table 509 – Primitives issued by MSCY1M to DMPMM1	804
Table 510 – Primitives issued by DMPMM1 to MSCY1M	804
Table 511 – Primitives issued by MSAL1M, MSAC1M to DMPMM1	805
Table 512 – Primitives issued by DMPMM1 to MSAL1M, MSAC1M	805
Table 513 – Primitives issued by MMAC1 to DMPMM1	805
Table 514 – Primitives issued by DMPMM1 to MMAC1	805
Table 515 – Primitives issued by MSCS1M to DMPMM1	806
Table 516 – Primitives issued by DMPMM1 to MSCS1M	806
Table 517 – Primitives issued by DMPMM1 to DL	807
Table 518 – Primitives issued by DL to DMPMM1	808
Table 519 – Parameters used with primitives exchanged with DMPMM1	809
Table 520 – Possible values of status	809
Table 521 – DMPMM1 state table	811
Table 522 – Functions used by the DMPMM1	819
Table 523 – Primitives issued by FSPMM2 to DMPMM2	820
Table 524 – Primitives issued by DMPMM2 to FSPMM2	820
Table 525 – Primitives issued by MSAC2M to DMPMM2	821
Table 526 – Primitives issued by DMPMM2 to MSAC2M	821
Table 527 – Primitives issued by MMAC2 to DMPMM2	821
Table 528 – Primitives issued by DMPMM2 to MMAC2	821
Table 529 – Primitives issued by DMPMM2 to DL	822
Table 530 – Primitives issued by DL to DMPMM2	822
Table 531 – Parameters used with primitives exchanged with DMPMM2	822

Table 532 – DMPMM2 state table	823
Table 533 – Functions used by DMPMM2	827
Table 534 – Bus parameter/reaction times for a DP-slave	827
Table 535 – APDU Header	828
Table 536 – APDU Body	829
Table 537 – Transfer syntax for Array	835
Table 538 – Transfer Syntax for Structure	835
Table 539 – Common Variable Object Attributes	835
Table 540 – Variable Type Identifiers	836
Table 541 – FIFO Variable Object Attributes	836
Table 542 – Error Codes	837
Table 543 – Primitives exchanged between FAL-User and FSPM	838
Table 544 – REQUEST.req FSPM constraints	840
Table 545 – REQUEST.req FSPM actions	840
Table 546 – RESPONSE.cnf FSPM constraints	841
Table 547 – RESPONSE.cnf FSPM actions	842
Table 548 – AR Send.ind proxy FSPM constraints	842
Table 549 – AR Send.ind proxy FSPM actions	842
Table 550 – AR Send.ind real FSPM constraints	843
Table 551 – AR Send.ind real FSPM Actions	843
Table 552 – Primitives issued by FSPM to ARPM	844
Table 553 – Primitives issued by ARPM to FSPM	844
Table 554 – Primitives issued by ARPM to ARPM	844
Table 555 – AR Send.req ARPM Constraints	844
Table 556 – AR Send.req ARPM Actions	845
Table 557 – AR Acknowledge.req ARPM Constraints	845
Table 558 – AR Acknowledge.req ARPM Actions	845
Table 559 – AR Send.ind ARPM Constraints	845
Table 560 – AR Send.req ARPM Actions	846
Table 561 – Primitives issued by ARPM to DLPM	846
Table 562 – Primitives issued by DLPM to ARPM	847
Table 563 – Primitives issued by DLPM to Data Link Layer	847
Table 564 – Primitives issued by Data Link Layer to DLPM	847
Table 565 – AR Send.req DLPM Constraints	848
Table 566 – AR Send.req DLPM Actions	848
Table 567 – AR Acknowledge.req DLPM Constraints	848
Table 568 – AR Acknowledge.req DLPM Actions	848
Table 569 – DL-UNITDATA.ind DLPM Constraints	849
Table 570 – DL-UNITDATA.ind DLPM Actions	849
Table 571 – Data Types	851
Table 572 – Data Types	851
Table 573 – Type 5 APDU Header Format	852
Table 574 – FDA Address Use	853
Table 575 – FDA Address header field APDUs sent by a client VCR endpoint	854
Table 576 – FDA Address header field APDUs sent by a server VCR endpoint	854
Table 577 – FDA Address header field APDUs sent by a publisher VCR endpoint	854

Table 578 – FDA Address header field APDUs sent by a report source VCR endpoint	855
Table 579 – APDU Trailer Fields	855
Table 580 – Request APDU Parameters	857
Table 581 – SMK FDA Address Values	859
Table 582 – SMK FDA Address Values	860
Table 583 – Request APDU Parameters	861
Table 584 – SMK FDA Address Values For SM Identify	862
Table 585 – SMK FDA Address Values For SMK Set Assignment Info Request APDUs	862
Table 586 – SMK Clear Address Request APDU Parameters	863
Table 587 – SMK FDA Address Values For SMK Set Assignment Info Request APDUs	863
Table 588 – SMK Set Assignment Info Request APDU Parameters	864
Table 589 – SMK Set Assignment Info Response APDU Parameters	865
Table 590 – SMK FDA Address Values For SMK Device Clear Assignment Info APDUs	865
Table 591 – SMK Clear Assignment Info Request APDU Parameters	866
Table 592 – SMK FDA Address Values For SMK Device Annunciation Request APDUs	866
Table 593 – SMK Device Annunciation Request APDU Parameters	867
Table 594 – Initiate Request APDU Parameters	869
Table 595 – Initiate Response APDU Parameters	869
Table 596 – Abort Request APDU Parameters	869
Table 597 – Get Response APDU Parameters	870
Table 598 – Identify Response APDU Parameters	870
Table 599 – Get OD Request APDU Parameters	870
Table 600 – Get OD Response APDU Parameters	871
Table 601 – Initiate Put OD Request APDU Parameters	871
Table 602 – Put OD Request APDU Parameters	871
Table 603 – Generic Initiate Download Sequence Request APDU Parameters	872
Table 604 – Generic Download Segment Request APDU Parameters	873
Table 605 – Generic Terminate Download Sequence Request APDU Parameters	873
Table 606 – Response APDU Parameters	873
Table 607 – Initiate Download Sequence Request APDU Parameters	874
Table 608 – Download Segment Request APDU Parameters	874
Table 609 – Download Segment Response APDU Parameters	874
Table 610 – Terminate Download Sequence Request APDU Parameters	875
Table 611 – Initiate Upload Sequence Request APDU Parameters	875
Table 612 – Upload Segment Request APDU Parameters	876
Table 613 – Upload Segment Response APDU Parameters	876
Table 614 – Terminate Upload Sequence Request APDU Parameters	876
Table 615 – Request Domain Download Request APDU Parameters	877
Table 616 – Request Domain Upload Request APDU Parameters	877
Table 617 – Create Program Invocation Request APDU Parameters	878
Table 618 – Create Program Invocation Response APDU Parameters	878
Table 619 – Delete Program Invocation Request APDU Parameters	878
Table 620 – Start Request APDU Parameters	879
Table 621 – Stop Request APDU Parameters	879
Table 622 – Resume Request APDU Parameters	879
Table 623 – Reset Request APDU Parameters	879

Table 624 – Kill Request APDU Parameters.....	880
Table 625 – Read Request APDU Parameters	880
Table 626 – Read Response APDU Parameters.....	880
Table 627 – Read with Subindex Request APDU Parameters	881
Table 628 – Read with Subindex Response APDU Parameters	881
Table 629 – Write Request APDU Parameters	881
Table 630 – Write with Subindex Request APDU Parameters.....	882
Table 631 – Define Variable List Request APDU Parameters	882
Table 632 – Define Variable List Response APDU Parameters	882
Table 633 – Delete Variable List Request APDU Parameters	883
Table 634 – Information Report Request APDU Parameters	883
Table 635 – Information Report with Subindex Request APDU Parameters	883
Table 636 – Information Report On Change Request APDU Parameters	884
Table 637 – Information Report On Change with Subindex Request APDU Parameters	884
Table 638 – Event Notification Request APDU Parameters	884
Table 639 – Alter Event Condition Monitoring Request APDU Parameters	884
Table 640 – Acknowledge Event Notification Request APDU Parameters.....	885
Table 641 – LAN Redundancy Diagnostic Message Request APDU Parameters	886
Table 642 – LAN Redundancy Get Information Response APDU Parameters.....	887
Table 643 – LAN Redundancy Get Statistics Request APDU Parameters.....	889
Table 644 – Object Description Header.....	891
Table 645 – Null Object	891
Table 646 – Structure of the List of Object Descriptions.....	891
Table 647 – Structure of a Load Region in the S-OD.....	892
Table 648 – Structure of a Function Invocation in the DP-OD.....	893
Table 649 – Structure of an Event in the S-OD	893
Table 650 – Structure of a Data Type in the ST-OD	894
Table 651 – Structure of a Data Type Structure Description in the ST-OD.....	894
Table 652 – Structure of a Simple Variable in the S-OD	895
Table 653 – Structure of an Array in the S-OD	895
Table 654 – Structure of a Record in the S-OD	896
Table 655 – Structure of a Variable List in the DV-OD.....	896
Table 656 – Common Error Parameters	897
Table 657 – PI Error Parameters.....	897
Table 658 – OD Error Parameters.....	897
Table 659 – Error Class and Error Code Values.....	898
Table 660 – SMKPM Service Primitives	900
Table 661 – SMKPM states.....	901
Table 662 – SMKPM state table – Initialization	902
Table 663 – SMKPM state table – Receive Transitions	902
Table 664 – SMKPM state table – Internal Events.....	907
Table 665 – HseRepeatTimerExpires ()	908
Table 666 – RcvNewNetworkAddress (interface, address)	908
Table 667 – RcvMsg ().....	908
Table 668 – SntpSyncLost ().....	908
Table 669 – AddressToClear (sm_svc)	

Table 670 – AssignmentInfo_Set ()	909
Table 671 – ConfigurationSessionActive ()	909
Table 672 – DeviceRedundancyState ()	909
Table 673 – DevId_Match (sm_svc)	909
Table 674 – DuplicateQueryIdMatch (sm_svc)	909
Table 675 – DuplicatePdTagDetected ()	910
Table 676 – FdaAddressType (sm_svc)	910
Table 677 – IsValid (sm_svc)	910
Table 678 – NetworkAddressChange (interface, address)	910
Table 679 – NumberOfAssignedAddresses ()	910
Table 680 – OperationalRestore ()	911
Table 681 – PdTag_Match (sm_svc)	911
Table 682 – PdTagDeviceIndex_Check (sm_svc)	911
Table 683 – Query_Match (sm_svc)	911
Table 684 – QueryType (sm_svc)	911
Table 685 – SmCacheEntry (sm_svc)	911
Table 686 – Clear_Address (interface_to_clear)	912
Table 687 – Clear_DuplicatePdTagFlag ()	912
Table 688 – Get_AddlCode ()	912
Table 689 – New_Address (interface, address)	912
Table 690 – Restart_HseRepeatTimer ()	913
Table 691 – Restore_Defaults ()	913
Table 692 – Send_SM_CommonErrorRsp (sm_service_type, svc_spec_params)	913
Table 693 – Send_SM_ReqRspMessage (sm_svc)	913
Table 694 – Set_Assignment_Data (sm_svc)	913
Table 695 – Set_DuplicatePdTagFlag ()	913
Table 696 – SvcType (sm_svc)	914
Table 697 – Additional Code Used by Error Class and Code	914
Table 698 – Additional Code Parameter Ids	914
Table 699 – Primitives issued by FSPM to ARPM	916
Table 700 – Primitives issued by ARPM to FSPM	916
Table 701 – Parameters used with primitives exchanged between FSPM and ARPM	917
Table 702 – Client / Server ARPM states	919
Table 703 – Client / Server ARPM state table – Sender transitions	920
Table 704 – Client / Server ARPM state table – Receiver transitions	921
Table 705 – Primitives issued by FSPM to ARPM	923
Table 706 – Primitives issued by ARPM to FSPM	924
Table 707 – Parameters used with primitives exchanged between FSPM and ARPM	924
Table 708 – Publisher / Subscriber ARPM States	925
Table 709 – MulticastARPM state table – sender transitions	926
Table 710 – MulticastARPM state table – receiver transitions	927
Table 711 – BuildFAL-ErrPDU()	927
Table 712 – BuildFAL-ReqRspPDU()	928
Table 713 – GetArepId()	928
Table 714 – ConfigurationArCheckOK()	928
Table 715 – FAL_Pdu_BufferSize()	928

Table 716 – FAL_Pdu_Confirmed()	928
Table 717 – FAL_Pdu_DuplicateMsg ()	928
Table 718 – FAL_Pdu_GetVcrlId()	929
Table 719 – FAL_Pdu_InactivityCloseTime()	929
Table 720 – FAL_Pdu_TransmitDelayTime()	929
Table 721 – FAL_Pdu_SvcType()	929
Table 722 – FAL_Pdu_RemoteAddress()	929
Table 723 – FAL_Pdu_TrailerFields()	929
Table 724 – FAL_Pdu_ServiceSpecificParameters()	930
Table 725 – FAL_Pdu_Valid()	930
Table 726 – MaxOutstandingReached()	930
Table 727 – StartInactivityCloseTimer()	930
Table 728 – Primitives issued by ARPM to DMPM	931
Table 729 – Primitives issued by DMPM to ARPM	931
Table 730 – Parameters used with primitives exchanged between ARPM and DMPM	931
Table 731 – Primitives exchanged between the Socket model and DMPM	931
Table 732 – Parameters of DMPM/Socket model primitives	932
Table 733 – DMPM state descriptions	932
Table 734 – DMPM state table – sender transitions	933
Table 735 – DMPM state table – receiver transitions	934
Table 736 – ConnectionOriented	934
Table 737 – GetBufferedData	934
Table 738 – GetConnectionId	934
Table 739 – LoadBuffer	934
Table 740 – RemainingBufferSizeCheck	935
Table 741 – StartTransmitDelayTimer	935
Table 742 – Type 6 specific transition	936
Table 743 – Primitives issued by FSPM to ARPM	937
Table 744 – Primitives issued by ARPM to FSPM	937
Table 745 – Parameters used with primitives exchanged between FSPM and ARPM	937
Table 746 – QUB ARPM states	939
Table 747 – QUB-PC ARPM state table – sender transactions	940
Table 748 – QUB-PC ARPM state table – receiver transactions	942
Table 749 – Function GetArepld ()	945
Table 750 – Function BuildFAL-PDU	945
Table 751 – Function FAL_Pdu_Type	946
Table 752 – Function AbortIdentifier	946
Table 753 – Function AbortReason	946
Table 754 – Function AbortDetail	946
Table 755 – Function ApExplicitConnection	946
Table 756 – Function ImmediateAcknowledge	946
Table 757 – Function ArFspmService	947
Table 758 – Function ArAcceeSupported	947
Table 759 – Function MaxFalPduLengthTest	947
Table 760 – Primitives issued by FSPM to ARPM	947
Table 761 – Primitives issued by ARPM to FSPM	947

Table 762 – Parameters used with primitives exchanged between FSPM and ARPM	948
Table 763 – BNU ARPM states	949
Table 764 – BNU-PC ARPM state table - sender transactions	950
Table 765 – BNU-PC ARPM state table - receiver transactions	951
Table 766 – Function GetArepld ()	951
Table 767 – Function BuildFAL-PDU	952
Table 768 – Function FAL_Pdu_Type	952
Table 769 – Function MaxFalPduLengthTest	952
Table 770 – Primitives issued by ARPM to DMPM	953
Table 771 – Primitives issued by DMPM to ARPM	954
Table 772 – Parameters used with primitives exchanged between ARPM and DMPM	954
Table 773 – Primitives exchanged between Data Link Layer and DMPM	955
Table 774 – DMPM state descriptions	955
Table 775 – DMPM state table – sender transactions	956
Table 776 – DMPM state table – receiver transactions	958
Table 777 – Function PickArep	960
Table 778 – Function FindAREP	960
Table 779 – Function LocateArep	960
Table 780 – Example of encoding of a SEQUENCE	963
Table 781 – Example of encoding of a SEQUENCE OF	964
Table 782 – Example of encoding of a CHOICE	964
Table 783 – Example of encoding of an object identifier	965
Table 784 – Example of encoding of a PDU	966
Table 785 – MPS PDU types	969
Table 786 – Fields of a CompactValuePDU	970
Table 787 – Fields of a VariableDescriptionPDU	976
Table 788 – Fields of AccessDescriptionPDU	978
Table 789 – Fields of TypeDescriptionPDU	979
Table 790 – Fields of a ListDescriptionPDU	979
Table 791 – Coding of the different MCS PDU types	981
Table 792 – Coding of the variable part of the PDU	981
Table 793 – Structure of Association Establishment Request	982
Table 794 – Structure of an associated establishment response	986
Table 795 – Structure of an association termination request	988
Table 796 – Structure of an association termination response	988
Table 797 – Structure of an association revocation request	989
Table 798 – Structure of an associated transfer request	990
Table 799 – Structure of an associated transfer acknowledgement	990
Table 800 – Structure of a non-associated transfer request	991
Table 801 – Structure of a non-associated transfer acknowledgement	992
Table 802 – Definitions of object classes	994
Table 803 – Definition of Sub-MMS Services	995
Table 804 – Structure of the antiduplication list	1069
Table 805 – Structure of the reassembly list	1074
Table 806 – PV_R/W parameter values	1082
Table 807 – PV_IND parameter values	1082

Table 808 – PV_LIS parameter values	1083
Table 809 – Constraints on PV_LIS parameter.....	1083
Table 810 – PV_AT parameter values	1083
Table 811 – PV_RE parameter values	1084
Table 812 – PV_UT parameter values.....	1084
Table 813 – Constraints on PV_RE parameter	1084
Table 814 – PH_R_A parameter values	1084
Table 815 – PH_R_S parameter values	1084
Table 816 – PH_R_P parameter values	1085
Table 817 – PH_P_A parameter values.....	1085
Table 818 – PH_P_S parameter values.....	1085
Table 819 – PH_P_P parameter values.....	1086
Table 820 – PH_COH parameter values.....	1086
Table 821 – PH_FIA parameter values.....	1086
Table 822 – PH_SPF parameter values	1086
Table 823 – PH_SPM parameter values.....	1087
Table 824 – PH_ACC parameter values.....	1087
Table 825 – PH_RES parameter values	1087
Table 826 – PH_AK parameter values	1087
Table 827 – PH_RA parameter values	1088
Table 828 – PH_SR parameter values	1088
Table 829 – PH_CF parameter values	1088
Table 830 – Constraints on PH_RA parameter	1088
Table 831 – Constraints on PH_SR parameter.....	1088
Table 832 – PT_OCT parameter values	1089
Table 833 – PT_BIN parameter values.....	1089
Table 834 – PT_VIS parameter values.....	1089
Table 835 – PT_BOO parameter values.....	1089
Table 836 – PT_BCD parameter values	1090
Table 837 – PT_BTM parameter values	1090
Table 838 – PT_INT parameter values.....	1090
Table 839 – PT_UNO parameter values	1090
Table 840 – PT_FPT parameter values	1091
Table 841 – PT_GTM parameter values.....	1091
Table 842 – PT_TAB parameter values.....	1091
Table 843 – PT_STR parameter values.....	1092
Table 844 – Constraints on PT_TAB parameter	1092
Table 845 – Constraints on PT_STR parameter	1092
Table 846 – Conformance classes for environment management.....	1095
Table 847 – Conformance classes for VMD management.....	1096
Table 848 – Conformance classes for PI management.....	1097
Table 849 – Conformance classes for domain management.....	1098
Table 850 – Conformance classes for variable/variable list management.....	1099
Table 851 – Conformance classes for event management.....	1100
Table 852 – Primitives issued by FAL to ARPM.....	1124
Table 853 – Primitives issued by ARPM to FAL.....	1124

Table 854 – Parameters used with Primitives Exchanged between FAL and ARPM.....	1124
Table 855 – QUB-TM ARPM States	1126
Table 856 – QUB-TM State Table - Sender Transactions	1126
Table 857 – QUB-TM State Table - Receiver Transactions	1127
Table 858 – Function GetArepld ()	1127
Table 859 – Function BuildFAL-PDU	1127
Table 860 – Function FAL_Pdu_Type	1127
Table 861 – Function ResetCounters	1128
Table 862 – Function IncrementCounter	1128
Table 863 – Function DecrementCounter	1128
Table 864 – Function GetCounterValue	1128
Table 865 – Primitives issued by ARPM to DMPM.....	1129
Table 866 – Primitives issued by DMPM to ARPM.....	1130
Table 867 – Parameters used with Primitives Exchanged between ARPM and DMPM.....	1130
Table 868 – Primitives exchanged between Type8 Data Link Layer and DMPM	1131
Table 869 – Type-8 DMPM state descriptions	1131
Table 870 – Type-8 DMPM state table – sender transactions.....	1132
Table 871 – Type-8 DMPM state table – receiver transactions	1136
Table 872 – Function PickArep	1137
Table 873 – Function FindAREP	1137
Table 874 – Function SelectNextArep	1137
Table 875 – Function ArepRole.....	1138
Table 876 – Function FalArHeader.....	1138
Table 877 – Function AddUcsPduHeader.....	1138
Table 878 – Function RemoveUcsPduHeader	1138
Table 879 – Function DILinkStatus.....	1138
Table 880 – Field with Fixed Format	1141
Table 881 – Object codes	1157
Table 882 – Standard data types	1157
Table 883 – Reason codes	1157
Table 884 – Coding for Date Type	1163
Table 885 – Primitives issued by VCR PM to FSPM	1169
Table 886 – Primitives issued by FSPM to VCR PM.....	1170
Table 887 – Primitives issued by FAL-User to VCR PM.....	1170
Table 888 – Primitives issued by VCR PM to FAL-User.....	1171
Table 889 – AP-VCR state machine transactions	1173
Table 890 – FSPM state table – sender transactions	1194
Table 891 – FSPM state table – receiver transactions.....	1195
Table 892 – QUU ARPM states.....	1196
Table 893 – QUU ARPM state table – sender transactions.....	1197
Table 894 – QUU ARPM state table – receiver transactions.....	1198
Table 895 – QUB ARPM states	1199
Table 896 – QUB ARPM state table – sender transactions.....	1200
Table 897 – QUB ARPM state table – receiver transactions	1202
Table 898 – BNU ARPM states	1207
Table 899 – BNU ARPM state table – sender transactions	1208

Table 900 – BNU ARPM state table – receiver transactions	1210
Table 901 – DMPM state descriptions	1212
Table 902 – DMPM state table – sender transactions.....	1213
Table 903 – DMPM state table – receiver transactions.....	1216
Table 904 – Primitives issued by FAS User to FSPM	1220
Table 905 – Primitives issued by FSPM to FAS User	1220
Table 906 – Primitives issued by FSPM to ARPM	1221
Table 907 – Primitives issued by ARPM to FSPM	1222
Table 908 – Parameters used with primitives exchanged between FSPM and ARPM	1222
Table 909 – Primitives issued from ARPM to DMPM	1223
Table 910 – Primitives issued by DMPM to ARPM.....	1223
Table 911 – Parameters used with primitives exchanged between ARPM and DMPM	1224
Table 912 – Primitives exchanged between Data Link Layer and DMPM.....	1225
Table 913 – Function SelectArep().....	1227
Table 914 – Function GetArepId()	1227
Table 915 – Function BuildFAS-PDU	1227
Table 916 – Function FAS_Pdu_Type	1227
Table 917 – Function AbortIdentifier	1228
Table 918 – Function AbortReason	1228
Table 919 – Function AbortDetail.....	1228
Table 920 – Function PickArep	1228
Table 921 – Function FindAREP	1228
Table 922 – Function LocateQubArep	1229
Table 923 – Function SetIdentifier().....	1229
Table 924 – Error Messages	1231
Table 925 – VARTYPE.....	1233
Table 926 – ITEMQUALITYDEF values.....	1234
Table 927 – STATEDEF values.....	1235
Table 928 – GROUPEXCEPTIONDEF values	1235
Table 929 – ACCESSRIGHTSDEF values.....	1235
Table 930 – PERSISTDEF values	1236
Table 931 – UUID values	1238
Table 932 – Data format for serialized connection data.....	1247
Table 933 – Primitives issued by FAL User to FSPM.....	1249
Table 934 – Primitives issued by FSPM to FAL User.....	1254
Table 935 – FSPM state descriptions.....	1258
Table 936 – FSPM state table.....	1259
Table 937 – Primitives issued by FSPM to ARPM	1303
Table 938 – Primitives issued by ARPM to FSPM	1304
Table 939 – Parameters used with primitives exchanged between FSPM and ARPM	1304
Table 940 – ARPM state descriptions	1304
Table 941 – ARPM state table	1305
Table 942 – Primitives issued by ARPM to DMPM.....	1306
Table 943 – Primitives issued by DMPM to ARPM.....	1306
Table 944 – Parameters used with primitives exchanged between ARPM and DMPM	1307
Table 945 – Primitives issued by DMPM to ORPC model	

Table 946 – Primitives issued by ORPC model to DMPM	1308
Table 947 – Parameters used with primitives exchanged between DMPM and ORPC model	1308
Table 948 – DMPM state descriptions	1308
Table 949 – DMPM state table	1309

INTERNATIONAL ELECTROTECHNICAL COMMISSION

DIGITAL DATA COMMUNICATIONS FOR MEASUREMENT AND CONTROL – FIELDBUS FOR USE IN INDUSTRIAL CONTROL SYSTEMS –

Part 6: Application Layer protocol specification

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the 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, the IEC publishes International Standards. 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. The 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 the 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 National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a patent as follows, where the [xx] notation indicates the holder of the patent right:

Type 8 and possibly other types:

DE 197 39 297 A1 [PxC] "Automatisierungssystem und Steuervorrichtung zur transparenten Kommunikation zwischen verschiedenen Netzwerken."

US Application number 09/145,848 Filing Date 09/02/98

[PxC] "Automation System and connecting Apparatus for the Transparent Communication between two Networks."

The IEC takes no position concerning the evidence, validity and scope of these patent rights.:

The holders of these patent rights have assured the IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holders of these patent rights are registered with the IEC. Information may be obtained from:

[PxC]: Phoenix Contact GmbH & Co. KG
Referat Patente / Patent Department
Postfach 1341
D-32819 Blomberg
Germany

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

International standard IEC 61158-6 has been prepared by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This third edition cancels and replaces the second edition published in 2000. This edition constitutes a technical revision.

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

FDIS	Report on voting
65C/293/FDIS	65C/301/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 edition includes the following significant changes from the prior edition:

- a) updates to Types 1, 2, 3, 4, 5, 7 and 8;
- b) addition of Types 6, 9 and 10.

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

The committee has decided that the contents of this publication will remain unchanged until 2007. At this date, the publication will be

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

IEC 61158 consists of the following parts, under the general title *Digital data communications for measurement and control — Fieldbus for use in industrial control systems*:

Part 1: Overview and guidance for the IEC 61158 series

Part 2: Physical Layer specification and service definition

Part 3: Data Link Service definition

Part 4: Data Link Protocol specification

Part 5: Application Layer Service definition

Part 6: Application Layer protocol specification

The contents of the corrigendum of July 2004 have been included in this copy.

0 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, which is based in part on the Basic Reference Model for Open Systems Interconnection. Both Reference Models subdivide the area of standardisation for interconnection into a series of layers of specification, each of manageable size.

This standard describes the Fieldbus Application Layer (FAL) protocol that defines the information interchange and the interactions between Application Entity invocations (AE-Is) to support the Application Services.

The Application Protocol provides the Application Service by making use of the services available from the Data Link Layer or other immediately lower layer. The relationship between the International Standards for Fieldbus Application Service, Fieldbus Application Protocol, Fieldbus Data Link Service and System Management is illustrated in Figure 1.

NOTE Systems Management, as used in this standard, is a local mechanism for managing the layer protocols

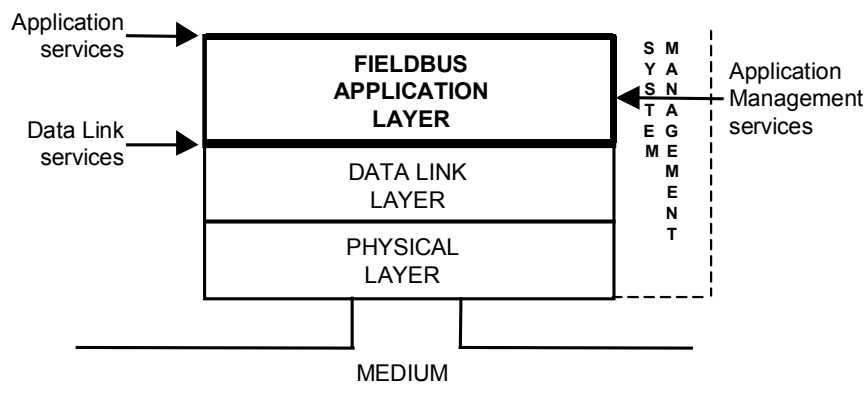


Figure 1 – Relationship of IEC 61158-6 to other fieldbus layers and to users of the fieldbus Application service

An Application Process uses the Fieldbus Application Layer services to exchange information with other Application Processes. The services define the abstract interface between the application process and the Application Layer.

The Application Layer protocol is the set of rules that governs the format and meaning of the information exchange between the Application Layers in various devices. The Application Layer uses the protocol to implement the Application Layer services definitions.

The protocol machine defines the various states of an Application Layer and the valid transitions between the states. It may be considered as a finite state machine. The protocol machine is described using state tables. The information is exchanged between the application process and the protocol machine through application service data units. The protocol machine exchanges information with other protocol machines through application protocol data units (APDU).

This set of Application Layer standards does not specify individual implementations or products, nor does it constrain the implementations of Application Entities (AEs) and interfaces within the industrial automation system.

This set of Application Layer standards does not contain test procedures to ensure compliance with such requirements.

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 Data Link entities 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:

- a) as a guide for implementors and designers;
- b) for use in the testing and procurement of equipment;
- c) as part of an agreement for the admittance of systems into the open systems environment;
- d) 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.

DIGITAL DATA COMMUNICATIONS FOR MEASUREMENT AND CONTROL – FIELDBUS FOR USE IN INDUSTRIAL CONTROL SYSTEMS –

Part 6: Application Layer protocol specification

1 Scope

The Fieldbus Application Layer (FAL) is an Application Layer communication standard designed to support the conveyance of time-critical application requests and responses among devices in an automation environment. 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 part of this standard specifies interactions between remote applications in terms of

- the encoding rules that are applied to all the Application Layer Protocol Data Units (APDUs);
- the formal Abstract Syntax definitions of such APDUs;
- the protocol state machine descriptions that handle the APDUs and the primitives in the correct sequences;
- the mappings of the APDUs to and from the Data Link Layer services defined in IEC 61158-3.

The FAL encoding rules are designed assuming that both the encoder (sender) and the decoder (receiver) have the common knowledge of the abstract syntax. Wherever possible, data types identifiers are not encoded and transferred over the network.

NOTE This is why the Abstract Syntax Notation One / Basic Encoding Rule is not practical for the FAL.

The purpose of this part of this standard is to define the protocol provided

- to the Fieldbus Data Link Layer at the boundary between the Application and Data Link Layers of the Fieldbus Reference Model, and
- to the System Management at the boundary between the System Management and Application Layers of the Fieldbus Reference Model.

This standard defines 10 different Application Layer protocols; each corresponds to the Application Layer service definitions specified in the IEC 61158-5 standard. They are identified in the 61158 series of specifications as Type 1, Type 2, Type 3, Type 4, Type 5, Type 6, Type 7, Type 8, Type 9, and Type 10. Although it may be possible to use different types of protocols on the same network provided that the underlying lower layers are compatible, this specification does not assure such interoperability among different types.

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 60559:1989, *Binary Floating-point Arithmetic for Microprocessor Systems*

IEC 61131-3:1993, *Programmable controllers – Part 3: Programming languages*

IEC 61158, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems*

IEC 61158-3:2003, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems — Data Link service definition*

IEC 61158-4:2003, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems — Data Link protocol specification*

IEC 61158-5:2003, *Digital data communications for measurement and control — Fieldbus for use in industrial control systems — Application layer service definition*

ISO/IEC 7498 (all parts), *Information technology – Open Systems Interconnection – Basic Reference Model*

ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*

ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*

ISO/IEC 9506-2:1990, *Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*

ISO/IEC 9545:1994, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 10646 (all parts), *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ISO 8649:1996, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element Protocol specification*

ISO 8650 (all parts), *Information technology – Open Systems Interconnection – Connection-oriented protocol for the Association Control Service Element Protocol specification*

3 Terms and definitions

3.1 Summary

For the purposes of this document, the following definitions apply.

3.2 Terms and definitions from other ISO/IEC standards

3.2.1 Terms and definitions from ISO/IEC 7498-1

- a) application entity;
- b) application process;
- c) application protocol data unit;
- d) application service element;
- e) application entity invocation;
- f) application process invocation;
- g) application transaction;
- h) real open system;
- i) transfer syntax.

3.2.2 Terms and definitions from ISO/IEC 8822

- a) abstract syntax;
- b) presentation context.

3.2.3 Terms and definitions from ISO/IEC 9545

- a) application-association;
- b) application-context;
- c) application context name;
- d) application-entity-invocation;
- e) application-entity-type;
- f) application-process-invocation;
- g) application-process-type;
- h) application-service-element;
- i) application control service element.

3.2.4 Terms and definitions from ISO/IEC 8824

- a) object identifier;
- b) type;
- c) value;
- d) simple type;
- e) structured type;
- f) component type;
- g) tag;
- h) Boolean type;
- i) true;
- j) false;
- k) integer type;
- l) bitstring type;
- m) octetstring type;
- n) null type;
- o) sequence type;
- p) sequence of type;
- q) choice type;
- r) tagged type;
- s) any type;
- t) module;
- u) production.

3.2.5 Terms and definitions from ISO/IEC 8825

- a) encoding (of a data value);
- b) data value;
- c) Identifier Octets (the singular form is used in this standard);
- d) Length Octet(s) (both singular and plural forms are used in this standard);
- e) Contents Octets.

3.3 Terms and definitions from IEC 61158-5

- a) application relationship;
- b) conveyance path;
- c) client;
- d) dedicated AR;
- e) dynamic AR;
- f) error class;
- g) error code;
- h) name;
- i) numeric identifier;
- j) peer;
- k) pre-defined AR endpoint;
- l) pre-established AR endpoint;
- m) publisher;
- n) subscriber;
- o) server.

3.4 Other terms and definitions

3.4.1 Summary

NOTE Type 1 definitions may apply to all the types except if the same item is used with different definitions in other type. In this case the latter has precedence.

The following terms and definitions are used in this standard.

3.4.2 Type 1 definitions

3.4.2.1 called

service user or a service provider that receives an indication primitive or a request APDU

3.4.2.2 calling

service user or a service provider that initiates a request primitive or a request APDU

3.4.2.3 interoperability

capability of User Layer entities to perform coordinated and cooperative operations using the services of the FAL

3.4.2.4 management information

network accessible information that supports the management of the Fieldbus environment

3.4.2.5 receiving

service user that receives a confirmed primitive or an unconfirmed primitive, or a service provider that receives a confirmed APDU or an unconfirmed APDU

3.4.2.6 resource

resource is a processing or information capability of a subsystem

3.4.2.7 sending

service user that sends a confirmed primitive or an unconfirmed primitive, or a service provider that sends a confirmed APDU or an unconfirmed APDU

3.4.3 Type 2 definitions

3.4.3.1 allocate

take a resource from a common area and assign that resource for the exclusive use of a specific entity

3.4.3.2 application

function or data structure for which data is consumed or produced

3.4.3.3 application objects

multiple object classes that manage and provide a run time exchange of messages across the network and within the network device

3.4.3.4 attribute

description of an externally visible characteristic or feature of an object

NOTE The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behaviour of an object. Attributes are divided into class attributes and instance attributes.

3.4.3.5 behaviour

indication of how an object responds to particular events

3.4.3.6 class

set of objects, all of which represent the same kind of system component

NOTE A class is a generalisation of an object; a template for defining variables and methods. All objects in a class are identical in form and behaviour, but usually contain different data in their attributes.

3.4.3.7 class attributes

attribute that is shared by all objects within the same class

3.4.3.8 class code

unique identifier assigned to each object class

3.4.3.9 class specific service

service defined by a particular object class to perform a required function which is not performed by a common service

NOTE A class specific object is unique to the object class which defines it

3.4.3.10 client

- a) object which uses the services of another (server) object to perform a task
- b) initiator of a message to which a server reacts

3.4.3.11 communication objects

components that manage and provide a run time exchange of messages across the network

EXAMPLES Connection Manager object, Unconnected Message Manager (UCMM) object, Message Router object.

3.4.3.12 connection

logical binding between application objects that may be within the same or different devices

NOTE Connections may be either point-to-point or multipoint.

3.4.3.13 connection ID (CID)

identifier assigned to a transmission that is associated with a particular connection between producers and consumers, providing a name for a specific piece of application information

3.4.3.14 connection path

an octet stream which defines the application object to which a connection instance applies

3.4.3.15 connection point

buffer which is represented as a subinstance of an Assembly object

3.4.3.16 consume

act of receiving data from a producer

3.4.3.17 consumer

node or sink that is receiving data from a producer

3.4.3.18 consuming application

application that consumes data

3.4.3.19 cyclic

repetitive in a regular manner

3.4.3.20 device

physical hardware connected to the link

NOTE: A device may contain more than one node.

3.4.3.21 device profile

collection of device dependent information and functionality providing consistency between similar devices of the same device type

3.4.3.22 end node

producing or consuming node

3.4.3.23 end point

one of the communicating entities involved in a connection

3.4.3.24 error

discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.4.3.25 frame

denigrated synonym for DLPDU

3.4.3.26 instance

the actual physical occurrence of an object within a class, identifying one of many objects within the same object class.

EXAMPLE California is an instance of the object class state.

NOTE The terms object, instance, and object instance are used to refer to a specific instance.

3.4.3.27 instance attribute

attribute that is unique to an object instance and not shared by the object class

3.4.3.28 instantiated

object that has been created in a device

3.4.3.29 keeper

object responsible for distributing link configuration data to all nodes on the link

3.4.3.30 little endian

Describes a model of memory organisation which stores the least significant octet at the lowest address, or for transfer, which transfers the lowest order octet first

3.4.3.31 Lpacket

a piece of application information that contains a size, control octet, tag, and link data.

NOTE Peer Data Link Layers use Lpackets to send and receive service data units from higher layers in the OSI stack.

3.4.3.32 member

piece of an attribute that is structured as an element of an array

3.4.3.33 message router

object within a node that distributes messaging requests to appropriate application objects

3.4.3.34 multipoint connection

connection from one node to many

NOTE Multipoint connections allow messages from a single producer to be received by many consumer nodes.

3.4.3.35 network

a set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.4.3.36 object

abstract representation of a particular component within a device, usually a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behaviour

3.4.3.37 object specific service

service unique to the object class which defines it

3.4.3.38 originator

client responsible for establishing a connection path to the target

3.4.3.39 point-to-point connection

connection that exists between exactly two application objects

3.4.3.40 produce

act of sending data to be received by a consumer

3.4.3.41 producer

node that is responsible for sending data

3.4.3.42 server

- a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request
- b) object which provides services to another (client) object

3.4.3.43 service

operation or function that an object and/or object class performs upon request from another object and/or object class

3.4.3.44 target

end-node to which a connection is established

3.4.3.45 temporary node

transient node

3.4.3.46 transaction id

field within a UCMM header that matches a response with the associated request

3.4.3.47 unconnected message manager (UCMM)

component within a node that transmits and receives unconnected explicit messages and sends them directly to the Message Router object

3.4.3.48 unconnected service

messaging service which does not rely on the set up of a connection between devices before allowing information exchanges

3.4.4 Type 3 definitions

3.4.4.1 access protection

limitation of the usage of an application object to one client

3.4.4.2 address-assignment-table

mapping of the client's internal I/O-Data object storage to the decentralised input and output data objects

3.4.4.3 channel

single physical or logical link of an input or output application object of a server to the process

3.4.4.4 channel related diagnosis

information concerning a specific element of an input or output application object, provided for maintenance purposes

EXAMPLE: validity of data

3.4.4.5 configuration check

comparison of the expected I/O-Data object structuring of the client with the real I/O-Data object structuring to the server in the start-up phase

3.4.4.6 configuration fault

an unacceptable difference between the expected I/O-Data object structuring and the real I/O-Data object structuring, as detected by the server

3.4.4.7 configuration identifier

representation of a portion of I/O Data of a single input- and/or output-module of a server.

3.4.4.8 configuration identifier related diagnosis

compromises all module specific data available at the server for maintenance purposes

3.4.4.9 control commands

action invocations transferred from client to server to clear outputs, freeze inputs and/or synchronise outputs

3.4.4.10 data consistency

means for coherent transmission and access of the input- or output-data object between and within client and server

3.4.4.11 Data-eXchange-Broadcast (DXB)

service for conveyance of information between the DP-master (Class 1) and the assigned DP-slaves initiating the Publisher and Subscriber functionality in the DP-slaves

3.4.4.12 default DL-address

value 126 as an initial value for DL-address, which has to be changed (e.g. by assignment of an DL-address via the fieldbus) before operation with a DP-master (class 1)

3.4.4.13 device related diagnosis

compromises all data related to the whole DP-slave available at the server for maintenance purposes

3.4.4.14 diagnosis information

all data available at the server for maintenance purposes

3.4.4.15 diagnosis information collection

system diagnosis information that is assembled at the client side

3.4.4.16 DP-master (class 1)

a controlling device which controls several DP-slaves (field devices)

NOTE: This is usually a programmable controller or a distributed control system.

3.4.4.17 DP-master (class 2)

controlling device which manages configuration data (parameter sets) and diagnosis data of a DP-master (Class 1), and that additionally performs all communication capabilities of a DP-master (Class 1)

3.4.4.18 DP-slave

field device that can be assigned to one DP-master (Class 1) as a provider for cyclic I/O data exchange, in addition acyclic functions and alarms could be provided

3.4.4.19 DXB request

cyclic data exchange service request with a specific DL service between the DP-master (Class 1) and the assigned DP-slaves initiating the Publisher functionality in the DP-slaves

3.4.4.20 DXB response

cyclic data exchange service response not to the individual address of the requester (DP-master (Class 1)) but to the broadcast address (=127)

3.4.4.21 freeze

function at the DP-slaves for simultaneous data transfer between the input data object and the process

3.4.4.22 group

set of DP-slaves which perform a Freeze or Sync function

3.4.4.23 I/O data

object designated to be transferred cyclically for the purpose of processing.

3.4.4.24 ident number

DP-master (Class 1) or DP-slave device type

3.4.4.25 index

address of an object within an application process

3.4.4.26 isochronous mode

special operation mode of a DP system that implies both a constant DP cycle with a fixed schedule of the cyclic and acyclic DP services, and the synchronisation of the application in the DP-master (Class 1) and the DP-slaves with this constant DP cycle

3.4.4.27 master parameter set

the configuration and parameterisation data of all DP-slaves that are assigned to the corresponding DP-master and the bus parameters

3.4.4.28 module

addressable unit inside the DP-slave

3.4.4.29 process data

object(s) which are already pre-processed and transferred acyclically for the purpose of information or further processing

3.4.4.30 publisher

role of a CR endpoint that transmits APDUs onto the Fieldbus for consumption by one or more subscribers

NOTE: A publisher may not be aware of the identity or the number of subscribers and it may publish its APDUs using a dedicated CR.

3.4.4.31 real configuration

input and output data structure of a DP-slave, including definition of data consistency

3.4.4.32 subscriber

role of a CREP in which it receives APDUs produced by a publisher

3.4.4.33 slot

address of a module within a DP-slave

3.4.4.34 sync

function at the DP-slaves for simultaneous data transfer between the output data object and the process

3.4.5 Type 4 definitions

None.

3.4.6 Type 5 definitions

None.

3.4.7 Type 6 definitions

3.4.7.1 called

service user or a service provider that receives an indication primitive or a request APDU

3.4.7.2 calling

service user or a service provider that initiates a request primitive or a request APDU

3.4.7.3 master

AREP which can initiate a connection on the associated QUB-PC AR

3.4.7.4 receiving

service user that receives a confirmed primitive or an unconfirmed primitive, or a service provider that receives a confirmed APDU or an unconfirmed APDU

3.4.7.5 sending

service user that sends a confirmed primitive or an unconfirmed primitive, or a service provider that sends a confirmed APDU or an unconfirmed APDU

3.4.7.6 slave

AREP which can not initiate a connection on the associated QUB-PC AR

3.4.8 Type 7 definitions

None.

3.4.9 Type 8 definitions

None.

3.4.10 Type 9 definitions

Type 9 uses the same definitions as Type 1.

3.4.11 Type 10 definitions**3.4.11.1 active connection control object**

instance of a certain FAL class that abstracts the interconnection facility (as Consumer and Provider) of an automation device

3.4.11.2 configuration data base

interconnection information maintained by the ACCO ASE

3.4.11.3 connection

the logical link between sink and source of attributes and services at different custom interfaces of RT-Auto ASEs is referred to as connection

NOTE There is a distinction between data and event connections. The logical link and the data flow between sink and source of automation data items is referred to as data connection. The logical link and the data flow between sink (method) and source (event) of operational services is referred to as event connection

3.4.11.4 connection channel

description of a connection between a sink and a source of data items

3.4.11.5 consumer

node or sink that is receiving data from a producer

3.4.11.6 consumerID

unambiguous identifier within the scope of the ACCO assigned by the consumer to recognize the internal data of a configured interconnection sink

3.4.11.7 data marshaling

the encoding of parameters of the FAL service primitives with respect to their interface definition

NOTE This is part of the abstract ORPC model.

3.4.11.8 engineering

abstract term that characterizes the client application or device responsible for configuring an automation system via interconnecting data items

3.4.11.9 event

an instance of a change of conditions

3.4.11.10 interface

collection of FAL class attributes and services that represents a specific view on the FAL class

3.4.11.11 interface definition language

syntax and semantics of describing service parameters in a formal way

NOTE This description is the input for the ORPC model, especially for the ORPC wire protocol.

3.4.11.12 interface pointer

key attribute that unambiguously addresses an object interface instance

3.4.11.13 logical device

a certain FAL class that abstracts a software component or a firmware component as an autonomous self-contained facility of an automation device

3.4.11.14 method

<object> a synonym for an operational service which is provided by the server ASE and invoked by a client

3.4.11.15 object remote procedure call

model for object oriented or component based remote method invocation

3.4.11.16 physical device

a certain FAL class that abstracts the hardware facilities of an automation device

3.4.11.17 property

a synonym for ASE attributes which are readable or writeable via operational ASE services

NOTE These services are generally named "get_<Attribute Name>" or "set_<Attribute Name>" and correspond with the IDL keywords "propget" and "propput" used in Type 10 clauses of IEC 61158-6.