

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 5-10: Application layer service definition – Type 10 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-10: Définition des services de la couche application – Éléments
de type 10**

Withdrawing



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 IEC, Geneva, Switzerland

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

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Industrial communication networks – Fieldbus specifications –
Part 5-10: Application layer service definition – Type 10 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-10: Définition des services de la couche application – Éléments
de type 10**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-1736-8

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	13
INTRODUCTION.....	15
1 Scope.....	16
1.1 General.....	16
1.2 Specifications.....	17
1.3 Conformance.....	17
2 Normative references	17
3 Terms, definitions, abbreviations, symbols and conventions	19
3.1 Referenced terms and definitions	20
3.2 Additional terms and definitions for decentralized periphery	20
3.3 Additional terms and definitions for media redundancy	28
3.4 Abbreviations and symbols.....	29
3.5 Conventions	31
4 Concepts.....	37
5 Data type ASE.....	38
5.1 General.....	38
5.2 Formal definition of data type objects.....	43
5.3 FAL defined data types.....	44
5.4 Data type ASE service specification.....	66
6 Communication model for common services.....	66
6.1 Concepts.....	66
6.2 ASE data types.....	67
6.3 Application Service Elements.....	68
7 Communication model for decentralized periphery.....	189
7.1 Concepts.....	189
7.2 ASE data types.....	207
7.3 ASEs.....	207
7.4 Behavior of an IO device	524
7.5 Behavior of an IO controller.....	583
7.6 Application characteristics.....	588
7.7 Summary of FAL services.....	589
Annex A (informative) Device instances	592
Annex B (informative) Components of an Ethernet interface	594
Annex C (informative) Scheme of MAC address assignment	598
Annex D (informative) Collection of objects.....	599
Annex E (informative) Measurement of the fast startup time	600
Annex F (informative) Dynamic Frame Packing.....	601
Annex G (informative) Building IR Data.....	609
Bibliography.....	614
Figure 1 – Data type class hierarchy example.....	38
Figure 2 – NetworkTime date relation	59
Figure 3 – FAL ASEs communication architecture.....	67
Figure 4 – PTCP applications.....	87
Figure 5 – Clock drift measurement	100

Figure 6 – Multiple synchronization	101
Figure 7 – Media redundancy diagnosis dependencies	107
Figure 8 – Example of periods at a local port	173
Figure 9 – Example of communication between controlling devices and field devices.....	190
Figure 10 – Example of communication between an engineering station and several controlling and field devices.....	191
Figure 11 – Example of communication between field devices and a server station	191
Figure 12 – Example of communication between field devices	191
Figure 13 – Structural units of one arbitrary API of an IO device (general)	193
Figure 14 – Example 1 structural units for interfaces and ports within API 0	194
Figure 15 – Example 2 structural units for interfaces and ports within API 0	195
Figure 16 – Identification hierarchy	197
Figure 17 – Overview of application processes	200
Figure 18 – IO device with APs, slots and subslots	200
Figure 19 – Application Process with application process objects (APOs).....	203
Figure 20 – Access to a remote APO	204
Figure 21 – Access to a remote APO for provider/consumer association.....	205
Figure 22 – Example of one AR with two AREPs	206
Figure 23 – FAL ASEs communication architecture	207
Figure 24 – Relation of a record data object to one real object.....	209
Figure 25 – Relation of a record data object to two real objects	209
Figure 26 – Overview IO ASE service interactions	230
Figure 27 – Severity classification of diagnosis and maintenance	261
Figure 28 – State transition diagram DIAG_DIAG.....	296
Figure 29 – State transition diagram DIAG_MR.....	300
Figure 30 – State transition diagram DIAG_MD.....	304
Figure 31 – State transition diagram DIAG_QUALIFIED	307
Figure 32 – Example of a resource model at the alarm source	328
Figure 33 – Basic model for isochronous applications	397
Figure 34 – General isochronous application model (example CACF == 1)	398
Figure 35 – General isochronous application model (example CACF == 2)	399
Figure 36 – ASE relations in an IO device operating in isochronous mode for a submodule	406
Figure 37 – State transition diagram of ISOM_SYNC	408
Figure 38 – State transition diagram ISOM_OUT	411
Figure 39 – State transition diagram ISOM_IN	416
Figure 40 – Assignment of communication relationship to application relationship	491
Figure 41 – Implicit application relationship	495
Figure 42 – Example IO application relationship (one-to-one)	496
Figure 43 – Example IO application relationship one-to-many	497
Figure 44 – Overview ASE state machines for IO device.....	524
Figure 45 – State transition diagram DEVSM	527
Figure 46 – State transition diagram REM_CHK.....	534
Figure 47 – State transition diagram LOC_LNK.....	541

Figure 48 – State transition diagram FOMR	547
Figure 49 – State transition diagram FOMD	549
Figure 50 – State transition diagram FODIAG	552
Figure 51 – State transition diagram RSMSM.....	556
Figure 52 – Ownership handling.....	560
Figure 53 – State transition diagram OWNSM.....	563
Figure 54 – State transition diagram ASSSM	563
Figure 55 – State transition diagram PLUGSM.....	575
Figure 56 – State transition diagram PULLSM.....	578
Figure 57 – State transition diagram SYNC_DIAG.....	580
Figure 58 – State diagram CTLSM	585
Figure 59 – Example of network topology including slower wireless segments	589
Figure 60 – Example of media redundancy including wireless segments	589
Figure A.1 – Instance model	592
Figure B.1 – Scheme of an Ethernet interface.....	594
Figure B.2 – Scheme of an Ethernet interface with bridging ability	595
Figure B.3 – Scheme of an Ethernet interface with optical ports.....	596
Figure B.4 – Scheme of an Ethernet interface with bridging ability using radio communication.....	597
Figure B.5 – Scheme of an Ethernet interface with radio communication.....	597
Figure C.1 – Scheme of MAC address assignment.....	598
Figure D.1 – Example for an intersection of IO device, slot, and AR.....	599
Figure E.1 – Measurement of the fast startup time	600
Figure F.1 – Frame Layout.....	601
Figure F.2 – Sub frame Layout.....	602
Figure F.3 – End to End.....	603
Figure F.4 – Dynamic frame packing.....	603
Figure F.5 – Dynamic frame packing – Truncation of outputs	604
Figure F.6 – Dynamic frame packing – Outbound Pack	604
Figure F.7 – Dynamic frame packing – Concatenation of inputs	605
Figure F.8 – Dynamic frame packing – Inbound Pack.....	606
Figure F.9 – Dynamic frame packing – Distributed watchdog	607
Figure F.10 – Interrelation between IO CR and dynamically packed frame	608
Figure G.1 – Bridge- and LineDelay	610
Figure G.2 – Sample Topology.....	610
Figure G.3 – Slip Stream Effect downstream.....	611
Figure G.4 – Using the slip stream effect in a comb topology downstream	612
Table 1 – State machine description elements	35
Table 2 – Description of state machine elements	35
Table 3 – Conventions used in state machines	35
Table 4 – Conventions for services used in state machines	36
Table 5 – Data type overview.....	41
Table 6 – V2 octets.....	43

Table 7 – L2 octets	45
Table 8 – E2 octets	48
Table 9 – E2 value range	48
Table 10 – Unipolar2.16 octets	48
Table 11 – Unipolar2.16 value range	49
Table 12 – N2 value range	50
Table 13 – N4 value range	51
Table 14 – X2 value range	52
Table 15 – X4 value range	53
Table 16 – C4 value range	53
Table 17 – T2 value range	55
Table 18 – T2 value range	56
Table 19 – D2 value range	56
Table 20 – R2 value range	57
Table 21 – UUID for decentralized peripherals	58
Table 22 – Status least significant Bit of the fractional portion (2^0)	59
Table 23 – Status value range	60
Table 24 – OctetString2+Unsigned8	62
Table 25 – Float32+Unsigned8 octets	62
Table 26 – Unsigned8+Unsigned8 octets	63
Table 27 – Unsigned16_S octets	63
Table 28 – Unsigned16_S meaning	63
Table 29 – Integer16_S octets	64
Table 30 – Integer16_S meaning	64
Table 31 – Unsigned8_S octets	64
Table 32 – Unsigned8_S meaning	65
Table 33 – OctetString_S octets	65
Table 34 – OctetString_S status bits	65
Table 35 – F message trailer with 4 octets	66
Table 36 – F message trailer with 5 octets	66
Table 37 – Get	74
Table 38 – Set	77
Table 39 – Local Set Command	81
Table 40 – Identify	83
Table 41 – Hello	85
Table 42 – Start bridge	93
Table 43 – Start slave	94
Table 44 – Start master	95
Table 45 – Stop bridge	96
Table 46 – Stop slave	97
Table 47 – Stop master	98
Table 48 – Sync state change	99
Table 49 – Line Delay change	99

Table 50 – PPM Set Prov Data	113
Table 51 – PPM Set Prov Status	113
Table 52 – PPM Activate	114
Table 53 – PPM Close	117
Table 54 – PPM Start	117
Table 55 – PPM Error	117
Table 56 – Get Cons Data	118
Table 57 – CPM Get cons status	119
Table 58 – CPM Set RedRole	120
Table 59 – CPM Activate	120
Table 60 – CPM NoData	122
Table 61 – CPM Stop	123
Table 62 – APMS Activate	126
Table 63 – APMR Activate	128
Table 64 – APMS A Data	129
Table 65 – APMR A Data	130
Table 66 – APMR Ack	130
Table 67 – APMS Error	131
Table 68 – APMS Error ERRCLS/ERRCODE	132
Table 69 – APMR Error	132
Table 70 – APMR Error ERRCLS/ERRCODE	132
Table 71 – APMS_Close	133
Table 72 – APMR_Close	133
Table 73 – Connect	135
Table 74 – Release	137
Table 75 – Read	138
Table 76 – Write	139
Table 77 – Control	140
Table 78 – System capabilities	145
Table 79 – Auto negotiation support and status	147
Table 80 – MDI Power Support	147
Table 81 – Link aggregation status	148
Table 82 – Remote systems data change	151
Table 83 – Local Get Time	154
Table 84 – Local Set Time	154
Table 85 – Local time changed event	155
Table 86 – Allowed values of Forwarding Mode	158
Table 87 – Allowed values of Fast Forwarding Multicast MAC Add	159
Table 88 – Allowed values of Reduction Ratio	160
Table 89 – Frame ID	160
Table 90 – Tx Port Entry	162
Table 91 – Dependencies of RedOrangePeriodBegin, OrangePeriodBegin, and GreenPeriodBegin	165

Table 92 – Port state change	169
Table 93 – Set port state	169
Table 94 – Flush filtering data base	170
Table 95 – MAU Type change	175
Table 96 – Set MAU Type	175
Table 97 – IP Multicast address	179
Table 98 – Set ARP Cache	179
Table 99 – Enterprise number	182
Table 100 – Vendor OUI	183
Table 101 – P Data	184
Table 102 – N Data	186
Table 103 – A Data	187
Table 104 – C Data	188
Table 105 – Requirements and features	189
Table 106 – Binding Record Data services	208
Table 107 – Persistence behavior for record data objects	211
Table 108 – Read	214
Table 109 – Read Services	215
Table 110 – Read Query	218
Table 111 – Write	220
Table 112 – Write Services	222
Table 113 – Data elements of Write Combined Object Container	224
Table 114 – Local Write Multiple	225
Table 115 – Local New Write Multiple	227
Table 116 – Local Set Input	240
Table 117 – Local Set Input IOCS	241
Table 118 – Local Get Input	242
Table 119 – Local Get Input IOCS	243
Table 120 – Local New Input	244
Table 121 – Local Set Redundancy	244
Table 122 – Local Set State	245
Table 123 – Local Data State Changed	246
Table 124 – Data elements of Read Record Input Data Object Element	247
Table 125 – Local Set Output	248
Table 126 – Local Set Output IOCS	249
Table 127 – Local Get Output	250
Table 128 – Local Get Output IOCS	251
Table 129 – Local New Output	252
Table 130 – Local Set Provider State	253
Table 131 – Data elements of Read Record Output Data Object Element	254
Table 132 – Data elements of Read Substitute Value	256
Table 133 – Data elements of Write Substitute Value	257
Table 134 – Data elements of Read LogBook Data	259

Table 135 – Local Create LogBook Entry	260
Table 136 – Channel Error Type	267
Table 137 – Ext Channel Error type	268
Table 138 – Allowed combinations of Channel Error Type and Ext Channel Error Type.....	272
Table 139 – Ext Channel Add Value for Accumulative Info	273
Table 140 – Local Add Diagnosis Entry.....	274
Table 141 – Local Remove Diagnosis Entry	276
Table 142 – Local Diagnosis Event	278
Table 143 – General Data definition for Diagnosis services.....	279
Table 144 – Data elements of Read Nested Diagnosis Information	295
Table 145 – Remote primitives issued or received by DIAG_DIAG	296
Table 146 – Local primitives issued or received by DIAG_DIAG	296
Table 147 – State table DIAG_DIAG	297
Table 148 – Functions, Macros, Timers and Variables used by DIAG_DIAG	298
Table 149 – Remote primitives issued or received by DIAG_MR	299
Table 150 – Local primitives issued or received by DIAG_MR.....	299
Table 151 – State table DIAG_MR	300
Table 152 – Functions, Macros, Timers and Variables used by DIAG_MR.....	302
Table 153 – Remote primitives issued or received by DIAG_MD	303
Table 154 – Local primitives issued or received by DIAG_MD	303
Table 155 – State table DIAG_MD	304
Table 156 – Functions, Macros, Timers and Variables used by the maintenance demanded entry	305
Table 157 – Remote primitives issued or received by DIAG_QUALIFIED	307
Table 158 – Local primitives issued or received by DIAG_QUALIFIED	307
Table 159 – State table DIAG_QUALIFIED	308
Table 160 – Functions, Macros, Timers and Variables used by DIAG_QUALIFIED	308
Table 161 – Alarm type	311
Table 162 – Alarm Notification	314
Table 163 – Channel Diagnosis	317
Table 164 – Manufacturer Specific Diagnosis.....	318
Table 165 – Submodule Diagnosis State.....	318
Table 166 – AR Diagnosis State	318
Table 167 – User Structure Identifier	320
Table 168 – Semantics of Specifier	321
Table 169 – Module State	336
Table 170 – Usage with respect to CR type	337
Table 171 – Detail	338
Table 172 – AR Info	339
Table 173 – Ident Info	340
Table 174 – Binding Context services	351
Table 175 – Connect.....	352
Table 176 – Connect Device Access	364

Table 177 – Release	367
Table 178 – Abort	368
Table 179 – Local AR Abort	369
Table 180 – Prm Begin	370
Table 181 – Prm End	373
Table 182 – Application Ready	374
Table 183 – General Data definition for identification services	377
Table 184 – Data elements of Read Module Diff Block	381
Table 185 – Data elements of Read API Data	382
Table 186 – Data elements of Read I&M0 Filter Data	383
Table 187 – Data elements of Read I&M0 Data	385
Table 188 – Data elements of Write I&M1 Data	387
Table 189 – Data elements of Read I&M1 Data	387
Table 190 – Data elements of Write I&M2 Data	388
Table 191 – Data elements of Read I&M2 Data	388
Table 192 – Data elements of Write I&M3 Data	389
Table 193 – Data elements of Read I&M3 Data	389
Table 194 – Data elements of Write I&M4 Data	389
Table 195 – Data elements of Read I&M4 Data	390
Table 196 – Data elements of Read Autoconfiguration Data	390
Table 197 – Data elements of Write Expected Fast Startup Data	393
Table 198 – Data elements of Read Expected Fast Startup Data	394
Table 199 – Data elements of Read GSD Data	395
Table 200 – Data elements of Read GSD Data	396
Table 201 – Data elements of Write Isochronous Mode Data	404
Table 202 – Data elements of Read Isochronous Mode Data	404
Table 203 – Local SYNCH Event	405
Table 204 – Remote primitives issued or received by ISOM_SYNC	408
Table 205 – Local primitives issued or received by ISOM_SYNC	408
Table 206 – State table ISOM_SYNC	409
Table 207 – Functions, Macros, Timers and Variables used by the ISOM_SYNC	409
Table 208 – Remote primitives issued or received for ISOM_OUT	410
Table 209 – Local primitives issued or received for ISOM_OUT	410
Table 210 – State table ISOM_OUT	412
Table 211 – Functions, Macros, Timers and Variables used by the ISOM_OUT	414
Table 212 – Remote primitives issued or received for ISOM_IN	415
Table 213 – Local primitives issued or received for ISOM_IN	415
Table 214 – State table ISOM_IN	417
Table 215 – Functions, Macros, Timers and Variables used by the ISOM_IN	418
Table 216 – Subslot number for interface submodules	424
Table 217 – Sync Properties Role	428
Table 218 – Sync Class	428
Table 219 – Distributed Watchdog Factor	429

Table 220 – Restart Factor For Distributed Watchdog	430
Table 221 – DFP Mode	430
Table 222 – SFIOCRProperties.DFPRedundantPathLayout.....	430
Table 223 – SFCRC16	431
Table 224 – Subslot number for port submodules	434
Table 225 – Fiber Optic Types	434
Table 226 – Fiber Optic Cable Types	435
Table 227 – TimePLLWindow.....	439
Table 228 – Data elements of Read PDev Data	439
Table 229 – Data elements of Read PD Real Data	446
Table 230 – Data elements of Read PD Expected Data.....	450
Table 231 – Read PD Interface Data Real	454
Table 232 – Data elements of Write PD Interface Adjust.....	455
Table 233 – Data elements of Read PD Interface Adjust.....	455
Table 234 – Data elements of Write PD IR Data	456
Table 235 – Data elements of Read PD IR Data	460
Table 236 – Data elements of Write PD Sync Data.....	464
Table 237 – Data elements of Read PD Sync Data	465
Table 238 – Local Sync State Info	467
Table 239 – Data elements of Write PD IR Subframe Data.....	468
Table 240 – Data elements of Read PD IR Subframe Data	470
Table 241 – Data elements of Write PD Time Data	471
Table 242 – Data elements of Read PD Time Data	471
Table 243 – Data elements of Read PD Interface MRP Data Real.....	472
Table 244 – Data elements of Write PD Interface MRP Data Check	473
Table 245 – Data elements of Read PD Interface MRP Data Check	473
Table 246 – Data elements of Write PD Interface MRP Data Adjust	474
Table 247 – Data elements of Read PD Interface MRP Data Adjust	475
Table 248 – Data elements of Write PD Interface FSU Data Adjust.....	475
Table 249 – Data elements of Read PD Interface FSU Data Adjust.....	476
Table 250 – Data elements of Write PD NC Data Check	477
Table 251 – Data elements of Read PD NC Data Check	478
Table 252 – Data elements of Read PD Port Data Real	478
Table 253 – Data elements of Write PD Port Data Check.....	480
Table 254 – Data elements of Read PD Port Data Check.....	481
Table 255 – Data elements of Write PD Port Data Adjust.....	482
Table 256 – Data elements of Read PD Port Data Adjust.....	482
Table 257 – Data elements of Read Port FO Data Real	483
Table 258 – Data elements of Write PD Port FO Data Check	484
Table 259 – Data elements of Read PD Port FO Data Check	485
Table 260 – Data elements of Write PD Port FO Data Adjust	486
Table 261 – Data elements of Read PD Port FO Data Adjust	486
Table 262 – Data elements of Read PD Port MRP Data Real	487

Table 263 – Data elements of Write PD Port MRP Data Adjust	487
Table 264 – Data elements of Read PD Port MRP Data Adjust	488
Table 265 – Data elements of Read PD Port Statistic	488
Table 266 – Device Access	501
Table 267 – Companion AR	501
Table 268 – Acknowledge Companion AR	501
Table 269 – Startup Mode	501
Table 270 – Pull Module Alarm Allowed	502
Table 271 – Input Valid on Backup AR	505
Table 272 – Activate Redundancy Alarm	505
Table 273 – APStructureIdentifier with API := 0	506
Table 274 – APStructureIdentifier with API != 0	506
Table 275 – Traffic Classes versus RT Class	510
Table 276 – Frame ID	511
Table 277 – Reduction Ratios	512
Table 278 – Data elements of Read AR Data	518
Table 279 – Local Set AR State	523
Table 280 – Local AR In Data	523
Table 281 – Remote primitives issued or received by DEVSM	525
Table 282 – Local primitives issued or received by DEVSM	525
Table 283 – State table DEVSM	528
Table 284 – Functions, Macros, Timers and Variables by DEVSM	532
Table 285 – Remote primitives issued or received by REM_CHK	533
Table 286 – Local primitives issued or received by REM_CHK	534
Table 287 – State table REM_CHK	535
Table 288 – Functions, Macros, Timers and Variables by REM_CHK	540
Table 289 – Remote primitives issued or received LOC_LNK	540
Table 290 – Local primitives issued or received LOC_LNK	541
Table 291 – State table LOC_LNK	542
Table 292 – Functions, Macros, Timers and Variables used by LOC_LNK	545
Table 293 – Remote primitives issued or received by FOMR	546
Table 294 – Local primitives issued or received by FOMR	546
Table 295 – State table FOMR	547
Table 296 – Functions, Macros, Timers and Variables used by FOMR	548
Table 297 – Remote primitives issued or received by FOMD	549
Table 298 – Local primitives issued or received by FOMD	549
Table 299 – State table FOMD	550
Table 300 – Functions, Macros, Timers and Variables by FOMD	551
Table 301 – Remote primitives issued or received by FODIAG	551
Table 302 – Local primitives issued or received by FODIAG	551
Table 303 – State table FODIAG	552
Table 304 – Functions, Macros, Timers and Variables by FODIAG	553
Table 305 – Remote primitives issued or received by RSMSM	555

Table 306 – Local primitives issued or received by RSMSM.....	556
Table 307 – State table RSMSM	556
Table 308 – Functions, Macros, Timers and Variables used by RSMSM	557
Table 309 – Rules for Submodule State.Ident Info	558
Table 310 – Remote primitives issued or received by OWNSM and ASSSM.....	562
Table 311 – Local primitives issued or received by OWNSM	563
Table 312 – State table OWNSM	564
Table 313 – State table ASSSM.....	570
Table 314 – Functions, Macros, Timers and Variables used by OWNSM.....	571
Table 315 – Functions, Macros, Timers and Variables used by ASSSM	572
Table 316 – Rules for Submodule State.AR Info	572
Table 317 – Remote primitives issued or received by PLUGSM	573
Table 318 – Local primitives issued or received by PLUGSM	574
Table 319 – State table PLUGSM	576
Table 320 – Functions, Macros, Timers and Variables used by PLUGSM.....	577
Table 321 – Remote primitives issued or received by PULLSM.....	577
Table 322 – Local primitives issued or received by PULLSM.....	578
Table 323 – State table PULLSM	579
Table 324 – Functions, Macros, Timers and Variables used by PULLSM	579
Table 325 – Remote primitives issued or received by SYNC_DIAG	580
Table 326 – Local primitives issued or received by SYNC_DIAG.....	580
Table 327 – State table SYNC_DIAG	581
Table 328 – Functions, Macros, Timers and Variables used by SYNC_DIAG	582
Table 329 – Remote primitives issued or received by CTLSM	584
Table 330 – Local primitives issued or received by CTLSM.....	584
Table 331 – State table CTLSM	586
Table 332 – Functions, Macros, Timers and Variables used by CTLSM.....	588
Table 333 – FAL services of the IO device	590
Table 334 – FAL services of the IO controller	590

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 5-10: Application layer service definition –
Type 10 elements**

FOREWORD

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

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

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

International Standard IEC 61158-5-10 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This third edition cancels and replaces the second edition published in 2010. This edition constitutes a technical revision.

The main changes with respect to the previous edition are listed below:

- Corrections and Improvements
- Change from MRP integration to MRP reference
- Integration of dynamic frame packing, fragmentation and fast forwarding
- Integration of autoconfiguration
- Integration of seamless media redundancy MRPD
- Basic integration of the System Redundancy Layer
- Basic integration of the Configure In Run functionality
- Optimization of RT_CLASS_3 startup and forwarding
- Optimization of the startup time from power down
- New fiber cable type GI-PCF
- Removal of MRRT
- Update of the LLDP-EXT-MIB

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

FDIS	Report on voting
65C/763/FDIS	65C/773/RVD

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

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

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

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

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

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

The application service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This standard defines the application service characteristics that fieldbus applications and/or system management may exploit.

Throughout the set of fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the application layer service defined in this standard is a conceptual architectural service, independent of administrative and implementation divisions.

Withdrawn

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-10: Application layer service definition – Type 10 elements

1 Scope

1.1 General

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

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to type 10 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the Type 10 fieldbus Application Layer in terms of

- a) an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,
- b) the primitive actions and events of the service;
- c) the parameters associated with each primitive action and event, and the form which they take; and
- d) the interrelationship between these actions and events, and their valid sequences.

The purpose of this standard is to define the services provided to

- a) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and
- b) Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

This standard specifies the structure and services of the type 10 IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented Application Service Elements (ASEs) and a Layer Management Entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can

send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various Types of IEC 61158, and the corresponding protocols standardized in subparts of IEC 61158-6.

This standard may be used as the basis for formal Application Programming Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this standard, including

- a) the sizes and octet ordering of various multi-octet service parameters, and
- b) the correlation of paired request and confirm, or indication and response, primitives.

1.3 Conformance

This standard does not specify individual implementations or products, nor do they constrain the implementations of application layer entities within industrial automation systems.

There is conformance of equipment to this application layer service definition standard mainly achieved through implementation of the modeled behavior of an application layer user (e.g. see user state machines) accompanied by implementation of conforming application layer protocols that fulfill the application layer services as defined in this standard.

2 Normative references

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

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

IEC 61158-1:2014, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-5-3:2014, *Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definitions – Type 3 elements*

IEC 61158-6-3:2014, *Industrial communication networks – Fieldbus specifications – Part 6-3: Application layer protocol specification – Type 3 elements*

IEC 61158-6-10:2014, *Industrial communication networks – Fieldbus specifications – Part 6-10: Application layer protocol specification – Type 10 elements*

IEC 61800-7-203:2008, *Adjustable speed electrical power drive systems – Part 7-203: Generic interface and use of profiles for power drive systems – Profile type 3 specification*

IEC 62439-2, *Industrial communication networks – High availability automation networks – Part 2: Media Redundancy Protocol (MRP)*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

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

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

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

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

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

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

ISO/IEC 15802-1, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 1: Medium Access Control (MAC) service definition*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

IEEE 754, *IEEE Standard for Floating-Point Arithmetic*, available at <<http://www.ieee.org>>

IEEE 802, *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*, available at <<http://www.ieee.org>>

IEEE 802.1AB-2005, *IEEE Standard for Local and Metropolitan Networks: Station and Media Access Control Connectivity Discovery*, available at <<http://www.ieee.org>>

IEEE 802.1AS, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – IEEE standard for Local and metropolitan area networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, available at <<http://www.ieee.org>>

IEEE 802.1D, *IEEE Standard for Local and Metropolitan Area Networks – Media Access Control (MAC) Bridges*, available at <<http://www.ieee.org>>

IEEE 802.1Q, *IEEE Standard for Local and Metropolitan Area Networks – Media Access Control (MAC) Bridges and Virtual Bridge Local Area Networks*, available at <<http://www.ieee.org>>

IEEE 802.3, *IEEE Standard for Information technology – Telecommunications and Information exchange between systems – Local and Metropolitan Area Networks – Specific Requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer*, available at <<http://www.ieee.org>>

IETF RFC 768, *User Datagram Protocol*; available at <<http://www.ietf.org>>

IETF RFC 791, *Internet Protocol*; available at <<http://www.ietf.org>>

IETF RFC 792, *Internet Control Message Protocol*; available at <<http://www.ietf.org>>

IETF RFC 826, *An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*; available at <<http://www.ietf.org>>

IETF RFC 1034, *Domain names - concepts and facilities*; available at <<http://www.ietf.org>>

IETF RFC 1112, *Host Extensions for IP Multicasting*; available at <<http://www.ietf.org>>

IETF RFC 1573, *Evolution of the Interfaces Group of MIB-II*; available at <<http://www.ietf.org>>

IETF RFC 2131, *Dynamic Host Configuration Protocol*; available at <<http://www.ietf.org>>

IETF RFC 2020, *IEEE 802.12 Interface MIB*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 2132, *DHCP Options and BOOTP Vendor Extensions*; available at <<http://www.ietf.org>>

IETF RFC 2365, *Administratively Scoped IP Multicast*; available at <<http://www.ietf.org>>

IETF RFC 2674, *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions*; available at <<http://www.ietf.org>>

IETF RFC 2737, *Entity MIB (Version 2)*; available at <<http://www.ietf.org>>

IETF RFC 2863, *The Interfaces Group MIB*; available at <<http://www.ietf.org>>

IETF RFC 3330, *Special-Use IPv4 Addresses*; available at <<http://www.ietf.org>>

IETF RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*; available at <<http://www.ietf.org>>

IETF RFC 3490, *Internationalizing Domain Names in Applications (IDNA)*; available at <<http://www.ietf.org>>

IETF RFC 3621, *Power Ethernet MIB*; available at <<http://www.ietf.org>>

IETF RFC 4836, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, available at <<http://www.ietf.org>>

IETF RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, available at <<http://www.ietf.org>>

The Open Group – Publication C706, *Technical standard DCE1.1: Remote Procedure Call* (available at <<http://www.opengroup.org/onlinepubs/9629399/toc.htm>>)

3 Terms, definitions, abbreviations, symbols and conventions

For the purposes of this document, the following terms, definitions, symbols, abbreviations and conventions apply.

3.1 Referenced terms and definitions

3.1.1 ISO/IEC 7498-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 7498-1 apply:

- 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

3.1.2 ISO/IEC 8822 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8822 apply:

- a) abstract syntax
- b) presentation context

3.1.3 ISO/IEC 9545 terms

For the purposes of this document, the following terms as defined in ISO/IEC 9545 apply:

- 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

3.1.4 ISO/IEC 8824-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8824-1 apply:

- a) object identifier
- b) type

3.2 Additional terms and definitions for decentralized periphery

3.2.1

alarm

activation of an event that shows a critical state

3.2.2

alarm ack

acknowledgment of an event that shows a critical state

3.2.3**alarm data object**

object(s) which represent critical states referenced by device/slot/subslot/alarm type

3.2.4**allocate**

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

3.2.5**application**

function or data structure for which data is consumed or produced

3.2.6**application layer interoperability**

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

3.2.7**application objects**

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

3.2.8**application process**

part of a distributed application on a network, which is located on one device and unambiguously addressed

3.2.9**application process identifier**

distinguishes multiple application processes used in a device

3.2.10**application process object**

component of an application process that is identifiable and accessible through an FAL application relationship

Note 1 to entry: Application process object definitions are composed of a set of values for the attributes of their class (see the definition for Application Process Object Class Definition). Application process object definitions may be accessed remotely using the services of the FAL Object Management ASE. FAL Object Management services can be used to load or update object definitions, to read object definitions, and to dynamically create and delete application process objects and their corresponding definitions.

3.2.11**application process object class**

class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.12**application relationship**

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities

3.2.13**application relationship application service element**

application-service-element that provides the exclusive means for establishing and terminating all application relationships

3.2.14

application relationship endpoint

context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

Note 1 to entry: Each application process involved in the application relationship maintains its own application relationship endpoint.

3.2.15

attribute

description of an externally visible characteristic or feature of an object

Note 1 to entry: 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 behavior of an object. Attributes are divided into class attributes and instance attributes.

3.2.16

backup

status of the IO AR, which indicates that it, is in the standby state

3.2.17

behavior

indication of how an object responds to particular events

3.2.18

channel

representation of a single physical or logical link of an input or output application process object of a server to the process in order to support addressing of diagnosis information

Note 1 to entry: The channel typically represents a single connector or clamp as a real interface of a module or sub-module. This reference is used to identify points of failure within diagnosis PDUs.

3.2.19

channel related diagnosis

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

EXAMPLE open loop

3.2.20

class

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

Note 1 to entry: A class is a generalization of an object; a template for defining variables and methods. All objects in a class are identical in form and behavior, but usually contain different data in their attributes.

3.2.21

class attributes

attribute that is shared by all objects within the same class

3.2.22

class code

unique identifier assigned to each object class

3.2.23

class specific service

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

Note 1 to entry: A class specific object is unique to the object class which defines it.

3.2.24**clear**

status of the IO controller, which indicates that the control algorithm is currently not running

3.2.25**client**

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

3.2.26**common profile**

collection of device independent information and functionality providing consistency between all devices

3.2.27**communication data object**

object(s) which are parameter of communication relationships and referenced by device/ slot/ subslot/ index

3.2.28**configuration check**

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

3.2.29**configuration fault**

unacceptable difference between the expected IO-Data object structuring and the real IO-Data object structuring, as detected by the server

3.2.30**configuration identifier**

representation of a portion of IO Data of a single input- and/or output-module of a server

3.2.31**consume**

act of receiving data from a provider

3.2.32**consumer**

node or sink receiving data from a provider

3.2.33**context management**

network-accessible information (communication objects) that supports managing the operation of the fieldbus system, including the application layer

Note 1 to entry: Managing includes functions such as controlling, monitoring, and diagnosing.

3.2.34**conveyance path**

unidirectional flow of APDUs across an application relationship

3.2.35**cyclic**

repetitive in a regular manner

3.2.36

data consistency

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

3.2.37

device

physical hardware connected to the link

Note 1 to entry: A device may contain more than one node.

3.2.38

device ID

vendor assigned device type identification

3.2.39

device profile

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

3.2.40

diagnosis data object

object(s) which contain diagnosis information referenced by device/slot/subslot/index

3.2.41

diagnosis information

all data available at the server for maintenance purposes

3.2.42

dynamic reconfiguration

change of IO data objects without interruption of an established application relationship and continuous updating of non-changed IO data objects

3.2.43

endpoint

one of the communicating entities involved in a connection

3.2.44

engineering

abstract term that characterizes the client application or device responsible for configuring an automation system

3.2.45

error

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

3.2.46

error class

general grouping for related error definitions and corresponding error codes

3.2.47

error code

identification of a specific type of error within an error class

3.2.48

event

instance of a change of conditions

3.2.49**extended channel related diagnosis**

information concerning a specific element of a specific application process object, provided for maintenance purposes

EXAMPLE Link Fail

3.2.50**fast startup**

accelerated procedure to setup an IO device after a power cycle

3.2.51**frame**

unit of data transfer on a link

3.2.52**identification data object**

object(s) that contain information about device, module and sub-module manufacturer and type referenced by device/slot/subslot/index

3.2.53**implicit AR endpoint**

AR endpoint that is defined locally within a device without use of the create service

3.2.54**index**

address of a record data object within an application process

3.2.55**instance**

the actual physical occurrence of an object within a class that identifies one of many objects within the same object class

3.2.56**instance attributes**

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

3.2.57**instantiated**

object that has been created in a device

3.2.58**invocation**

act of using a service or other resource of an application process

Note 1 to entry: Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation. Also for service invocations, an Invoke ID may be used to unambiguously identify the service invocation and differentiate it from other outstanding service invocations.

3.2.59**IO controller**

controlling device, which acts as client for several IO devices (field devices)

Note 1 to entry: This is usually a programmable controller or a distributed control system.

3.2.60

IO data object

object designated to be transferred cyclically for the purpose of processing and referenced by device/slot/subslot

3.2.61

IO device

field device which acts as server for IO operation

3.2.62

IO parameter server

server for application parameter of IO devices (client)

Note 1 to entry: This is usually a device to backup parameter data and to log online changes of device parameter.

3.2.63

IO subsystem

subsystem composed of one IO controller and all its associated IO devices

3.2.64

IO supervisor

engineering device which manages commissioning and diagnosis of an IO system

3.2.65

IO system

system composed of all its IO subsystems

Note 1 to entry: As an example a PLC with more than one IO controller (network interface) controls one IO system composed of an IO subsystem for each IO controller.

3.2.66

Isochronous mode

IO system operating tightly synchronized with a jitter of less than 1 μ s

3.2.67

member

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

3.2.68

method

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

3.2.69

module

hardware or logical component of a physical device

3.2.70

network

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

3.2.71

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 a clearly defined interface and behavior

3.2.72**object specific service**

service unique to the object class which defines it

3.2.73**operate**

status of the IO controller that indicates that the control algorithm is currently running

3.2.74**peer**

role of an AR endpoint in which it is capable of acting as both client and server

3.2.75**physical device**

automation or other network device

3.2.76**point-to-point connection**

connection that exists between exactly two application process objects

3.2.77**primary**

status of the IO AR that indicates that it is in the operating state

Note 1 to entry: Besides a primary IO AR a backup IO AR may exist. In example used for redundancy and dynamic reconfiguration of IO data.

3.2.78**provider**

node or source sending data to one or many consumer

3.2.79**PTCP domain**

certain number of PTCP subdomains in one IP subnet

3.2.80**PTCP subdomain**

certain amount of DTEs with synchronized clocks

3.2.81**record data object**

object(s) which are already pre-processed and transferred acyclically for the purpose of information or further processing and referenced by device/slot/subslot/index

3.2.82**resource**

processing or information capability

3.2.83**run**

status of the IO controller which indicates that the control algorithm is currently operating

3.2.84**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

3.2.85

service

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

3.2.86

slot

address of a structural unit within an IO device

Note 1 to entry: Within a modular device, a slot typically addresses a physical module. Within compact devices, a slot typically addresses a logical function or virtual module.

3.2.87

stop

status of the IO controller which indicates that the control algorithm is currently not running

3.2.88

submodule

hardware or logical component of a module

3.2.89

subslot

address of a structural unit within a slot

Note 1 to entry: A subslot may address a physical interface for submodules within a module. Generally, a subslot is a second level to structure data within a device.

3.2.90

vendor ID

central administrative number used as manufacturer identification

3.3 Additional terms and definitions for media redundancy

3.3.1

failure

termination of the ability of an item to perform a required function

Note 1 to entry: After failure the item has a fault.

Note 2 to entry: "Failure" is an event, as distinguished from "fault", which is a state.

Note 3 to entry: This concept as defined does not apply to items considering of software only.

3.3.2

fault

state of an item characterized by its inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources

Note 1 to entry: A fault is often the result of a failure of the item itself, but may exist without prior failure.

3.3.3

recovery

event when the item regains the ability to perform a required function, after a fault

3.3.4

recovery time

time period required for recovery

3.3.5**redundancy**

existence in an item of two or more means of performing a required function

Note 1 to entry: In this context, the existence of more than one path (consisting of links and switches) between end nodes.

3.3.6**ring**

network where each node is connected in series to two other nodes

Note 1 to entry: Nodes are connected to one another in the logical shape of a circle.

Note 2 to entry: Frames are passed sequentially between active nodes, each able to examine or modify the frame before forwarding it.

3.3.7**ring port**

port of a switch to which a ring link is attached

3.3.8**ring link**

link that connects two switches of a ring

3.3.9**switch node**

MAC bridge as defined in IEEE 802.1D

Note 1 to entry: It is called "switch" in this context.

3.4 Abbreviations and symbols**3.4.1 General**

AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
APMR	Acyclic Protocol Machine Receiver
APMS	Acyclic Protocol Machine Sender
APO	Application Process Object
AR	Application Relationship
AREP	Application Relationship End Point
ARL	Application Relationship List
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
BMA	Best Master Algorithm
CID	Connection ID
CIM	Computer Integrated Manufacturing
CIP	Control and Information Protocol
CR	Communication Relationship
CREP	Communication Relationship End Point
CTLMS	Controller state machine
DEVSM	Device state machine
DIAG_DIAG	Diagnosis entry state machine
DIAG_MD	Maintenance demanded entry state machine
DIAG_MR	Maintenance required entry state machine
DIAG_QUALIFIED	Qualified entry state machine
DL-	(as a prefix) Data Link-
DLC	Data Link Connection

DLL	Data Link Layer
DLSAP	Data Link Service Access Point
DNS	Domain Name Service
DP	Decentralized Peripherals
FAL	Fieldbus Application Layer
FIFO	First In First Out
FODIAG	Fiber Optic Diagnosis
FOMD	Fiber Optic Maintenance Demanded
FOMR	Fiber Optic Maintenance Required
HMI	Human Machine Interface
ID	Identifier
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
LME	Layer Management Entity
LOC_LNK	Local Link
MAU	Medium Attachment Unit
MIB	Management Information Base
MRC	Media Redundancy Client
MRM	Media Redundancy Manager
MRP	Media Redundancy Protocol
MRPD	Media Redundancy with Planned Duplication of frames
OSI	Open Systems Interconnect
OWNSM	Ownership Submodule
PDev	Physical Device
PDU	Protocol Data Unit
PL	Physical Layer
PLUGSM	Plug State Machine
PULLSM	Pull State Machine
REM_CHK	Remote check (neighborhood check)
REP	Route Endpoint
RSMSM	Real Submodule State Machine
SAP	Service Access Point
SCL	Security Level
SDU	Service Data Unit
SEM	State Event Matrix
SMIB	System Management Information Base
SMK	System Management Kernel
SNMP	Simple Network Management Protocol
STD	State Transition Diagram, used to describe object behavior
S-VFD	Simple Virtual Field Device
SYNC_DIAG	Synchronisation Diagnosis
UTC	Universal Time Coordinated

3.4.2 Additional abbreviations and symbols for decentralized periphery

ALPMI	Alarm Protocol Machine Initiator
ALPMR	Alarm Protocol Machine Responder
ARP	Address Resolution Protocol
BBUB	Buffer Buffer Unconfirmed Bidirectional
BBUU	Buffer Buffer Unconfirmed Unidirectional
BBUU-OM	Buffer Buffer Unconfirmed Unidirectional - One to Many
BMC	Best Master Clock
CL-RPC	Connectionless Remote Procedure Call
CM	Context Management
CPM	Cyclic Consumer Protocol Machine
DCE	OSF Distributed Computing Environment
DCP	Discovery and basic Configuration Protocol
DHCP	Dynamic Host Configuration Protocol
DIM	Device Interface Module
DLPDU	Data Link-Protocol Data Unit

DTE	Date Terminal Equipment
GSDML	General Station Description Markup Language
I&M	Identification and Maintenance Profile
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IOCR	Input Output Communication Relation
IOCS	Input Output Object Consumer Status
IOPS	Input Output Object Provider Status
IR	Isochronous Relay
IRT	Isochronous Real Time Protocol
ISO	International Organization for Standardization
IsoM	Isochronous Mode
ISOM_IN	Isochronous Mode Input Data state machine
ISOM_OUT	Isochronous Mode Output Data state machine
ISOM_SYNC	Isochronous Mode Sync state machine
LED	Light Emitting Diode
LLDP	Link Layer Discovery Protocol
LME	Layer Management Entity
lsb	Least Significant Bit
LT	Length/Type
MAC	Medium Access Control
MCR	Multicast Communication Relation
msb	Most Significant Bit
NCA	Network Computing Architecture
PL	Physical Layer
PPM	Cyclic Provider Protocol Machine
PTCP	Precision Transparent Clock Protocol
QQCB-CL	Queue Queue Confirmed Bi-directional - Connectionless
QQCB-CO	Queue Queue Confirmed Bi-directional - Connection-Oriented
RPC	Remote Procedure Call
RT	Real Time
RTA	Real Time Protocol Acyclic
RTC	Real Time Protocol Cyclic
RTC3PSM	Realtime Class 3 Port State Machine
TLV	Type Length Value (coding rule)
UDP	User Datagram Protocol
UUID	Universally Unique Identifier
VLAN	Virtual Local Area Network

3.4.3 Abbreviations and symbols for services

cnf	Confirmation
ind	Indication
req	Request
rsp	Response

3.5 Conventions

3.5.1 Overview

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of two parts, its class specification, and its service specification.

The class specification defines the attributes of the class. The attributes are accessible from instances of the class using the Object Management ASE services specified in Clause 5 of this standard. The service specification defines the services that are provided by the ASE.

3.5.2 General conventions

This standard uses the descriptive conventions given in ISO/IEC 10731.

3.5.3 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is shown below:

FAL ASE:	ASE Name
CLASS:	Class Name
CLASS ID:	#
PARENT CLASS:	Parent Class Name
ATTRIBUTES:	
1	(o) Key Attribute: numeric identifier
2	(o) Key Attribute: name
3	(m) Attribute: attribute name(values)
4	(m) Attribute: attribute name(values)
4.1	(s) Attribute: attribute name(values)
4.2	(s) Attribute: attribute name(values)
4.3	(s) Attribute: attribute name(values)
5	(c) Constraint: constraint expression
5.1	(m) Attribute: attribute name(values)
5.2	(o) Attribute: attribute name(values)
6	(m) Attribute: attribute name(values)
6.1	(s) Attribute: attribute name(values)
6.2	(s) Attribute: attribute name(values)
SERVICES:	
1	(o) OpsService: service name
2	(c) Constraint: constraint expression
2.1	(o) OpsService: service name
3	(m) MgtService: service name

- (1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.
- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2 048 are allocated for identifying user defined classes.
- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

NOTE The parent-class "TOP" indicates that the class being defined is an initial class definition. The parent class TOP is used as a starting point from which all other classes are defined. The use of TOP is reserved for classes defined by this standard.

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.

- a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
- b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
- c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify
 - i) fields of a structured attribute (4.1, 4.2, 4.3),
 - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2),
 - iii) the selection fields of a choice type attribute (6.1 and 6.2).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
 - a) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.
 - b) The label "OpsService" designates an operational service (1).
 - c) The label "MgtService" designates a management service (2).
 - d) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.5.4 Conventions for service definitions

3.5.4.1 General

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

3.5.4.2 Service parameters

Service primitives are used to represent service user/service provider interactions (ISO/IEC 10731). They convey parameters which indicate information available in the user/provider interaction. In any particular interface, not all parameters need be explicitly stated.

The service specifications of this standard use a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

- 1) Parameter name,
- 2) request primitive,
- 3) indication primitive,
- 4) response primitive, and
- 5) confirm primitive.

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column:

M parameter is mandatory for the primitive.

- U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. This does not imply that the support of the parameter is optional. When not provided, a default value for the parameter is assumed.
- C parameter is conditional upon other parameters or upon the environment of the service user.
- (blank) parameter is never present.
- S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

- a) a parameter-specific constraint:
“(=)” indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.
- b) an indication that some note applies to the entry:
“(n)” indicates that the following note "n" contains additional information pertaining to the parameter and its use.

3.5.4.3 Service procedures

The procedures are defined in terms of

- the interactions between application entities through the exchange of fieldbus Application Protocol Data Units, and
- the interactions between an application layer service provider and an application layer service user in the same system through the invocation of application layer service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the Fieldbus Application Layer.

3.5.5 Conventions used in state machines

The protocol sequences are described by means of State Machines.

In state diagrams states are represented as boxes state transitions are shown as arrows. Names of states and transitions of the state diagram correspond to the names in the textual listing of the state transitions.

The initialisation of all parameters (e.g. default values) of a state machine will be realised automatically during the instantiation of the machine (new instance). Implicite transition from POWER-ON to first state calling the INIT service. If it is necessary to have different instances of one machine type, different parameters are set by the INIT service.

The textual listing of the state transitions is structured as follows, see also Table 1.

- The first row contains the name of the transition.
- The second row defines the current state.
- The third row contains an optional event followed by Conditions starting with a “/” as first line character and finally followed by the Actions starting with a “=>” as first line character.
- The last row contains the next state.

If the event occurs and the conditions are fulfilled the transition fires, e.g. the actions are executed and the next state is entered.

The layout of a machine description is shown in Table 1. The meaning of the elements of a State Machine Description is shown in Table 2.

Table 1 – State machine description elements

#	Current state	Event or condition => action	Next state

Table 2 – Description of state machine elements

Description element	Meaning
Current state	Name of the given states.
Next state	
#	Name or number of the state transition.
Event	Name or description of the event.
/Condition	Boolean expression. The preceding "/" is not part of the condition.
=> Action	List of assignments and service or function invocations. The preceding "=>" is not part of the action.

The conventions used in the state machines are shown in Table 3.

Table 3 – Conventions used in state machines

Convention	Meaning
:=	Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.
xxx	A parameter name. Example: Identifier := reason means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'
"xxx"	Indicates fixed value. Example: Identifier := "abc" means value "abc" is assigned to a parameter named 'Identifier.'
= or ==	A logical condition to indicate an item on the left is equal to an item on the right.
<	A logical condition to indicate an item on the left is less than the item on the right.
>	A logical condition to indicate an item on the left is greater than the item on the right.
<> or !=	A logical condition to indicate an item on the left is not equal to an item on the right.
>>	A semantic condition with the meaning "newer"
<<	A semantic condition with the meaning "older"
&&	Logical "AND"
	Logical "OR"
<=	A logical condition to indicate an item on the left is equal or less than the item on the right.
>=	A logical condition to indicate an item on the left is equal or greater than the item on the right.
for (Identifier := start_value to end_value) actions endfor	This construct allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

Convention	Meaning
If (condition) actions else actions	This construct allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition. The parts beginning with "else" can be omitted if there is no action if the condition is not fulfilled.

The conventions used for service in the state machines are shown in Table 3.

Table 4 – Conventions for services used in state machines

Convention	Meaning
xxx.req ()	request according to ISO/IEC 7498-1 invokes the service and passes any required parameter parameters are omitted and listed within the state machine description
xxx.ind ()	indication according to ISO/IEC 7498-1 advices the activation of a requested service
xxx.rsp (+), xxx.cnf (+)	response according to ISO/IEC 7498-1 may positively acknowledge or complete an action previously invoked by a request primitive confirmation according to ISO/IEC 7498-1 primitive returned to the requestor to positively acknowledge or complete an action previously invoked by a request primitive parameters are omitted and listed within the state machine description
xxx.rsp (-), xxx.cnf (-)	response according to ISO/IEC 7498-1 may negatively acknowledge or complete an action previously invoked by a request primitive confirmation according to ISO/IEC 7498-1 primitive returned to the requestor to negatively acknowledge or complete an action previously invoked by a request primitive parameters are omitted and listed within the state machine description
xxx_req (), xxx_cnf (+), xxx_cnf (-)	this construct is used for the description of local and confirmed services, which are transferred between state machines parameters are listed within the state machine description and can be delivered inside the brackets optionally
xxx_ind ()	this construct is used for the description of local services and indicates the reception of this service parameters are listed within the state machine description and can be delivered inside the brackets optionally
xxx ()	this construct is used for the description of state machine internal functions (local) parameters are listed within the state machine description and can be delivered inside the brackets optionally
(), (+), (-)	replacement of the associated parameter list, described in the primitives definition except local services and functions can deliver parameters inside the brackets if the result is not necessary for further executions, the sign + and - can be omitted

Readers are strongly recommended to refer to the clauses for the AREP and CREP attribute definitions, the local functions and the FAL-PDU definitions to understand protocol machines. It is assumed that readers have sufficient knowledge of these definitions and they are used without further explanations.

In addition the following description elements are used:

Wildcard in names

name_XXX: “XXX” is used as wildcard string for all names beginning with “name”.

The typical use of a wildcard is an Event. In this context there are as many state transitions as possible events for this wildcard exists.

Conditional Macro

<CONDITIONAL -MACRO-NAME>

<

CONDITION1: macrobody1

CONDITION2: macrobody2

...

>

CONDITION1, CONDITION2, etc. define all possible cases for the conditional macro. The “CONDITIONAL-MACRO-NAME” acts as place holder for the “macrocode” depending on the result of the condition.

Replacement Macro

<REPLACEMENT-MACRO-NAME>

<

XXX= Name1 : macrobody1

XXX= Name2 : macrobody2

...

>

Name1, Name2, etc. define all possible cases for the replacement macro. The “REPLACEMENT-MACRO-NAME” acts as place holder for the “macrocode” depending on the value of the current use of the wildcard XXX.

EXAMPLE

<SERVICE_REQ_PARA>

<

XXX=Read: Para1, Para2

XXX=Write: Para1, Para2, Para3

>

when called in

MSAB_XXX.req(<SERVICE_REQ_PARA>)

will result in

MSAB_Read.req(Para1, Para2) or MSAB_Write.req(Para1, Para2, Para3)

4 Concepts

NOTE IEC 61158-1, Clause 9 describes the concepts of the application layer service descriptions and the templates used in this document.

5 Data type ASE

5.1 General

5.1.1 Overview

Fieldbus data types specify the machine independent syntax for application data conveyed by FAL services. The fieldbus application layer supports the definition and transfer of both basic and constructed data types. Encoding rules for the data types specified in this clause are provided in IEC 61158-6-10.

Basic types are atomic types that cannot be decomposed into more elemental types. Constructed types are types composed of basic types and other constructed types. Their complexity and depth of nesting is not constrained by this standard.

Data types are defined as instances of the data type class, as shown in Figure 1. Only a subset of the data types defined in this Clause is shown in this figure. Defining new types is accomplished by providing a numeric id and supplying values for the attributes defined for the data type class.

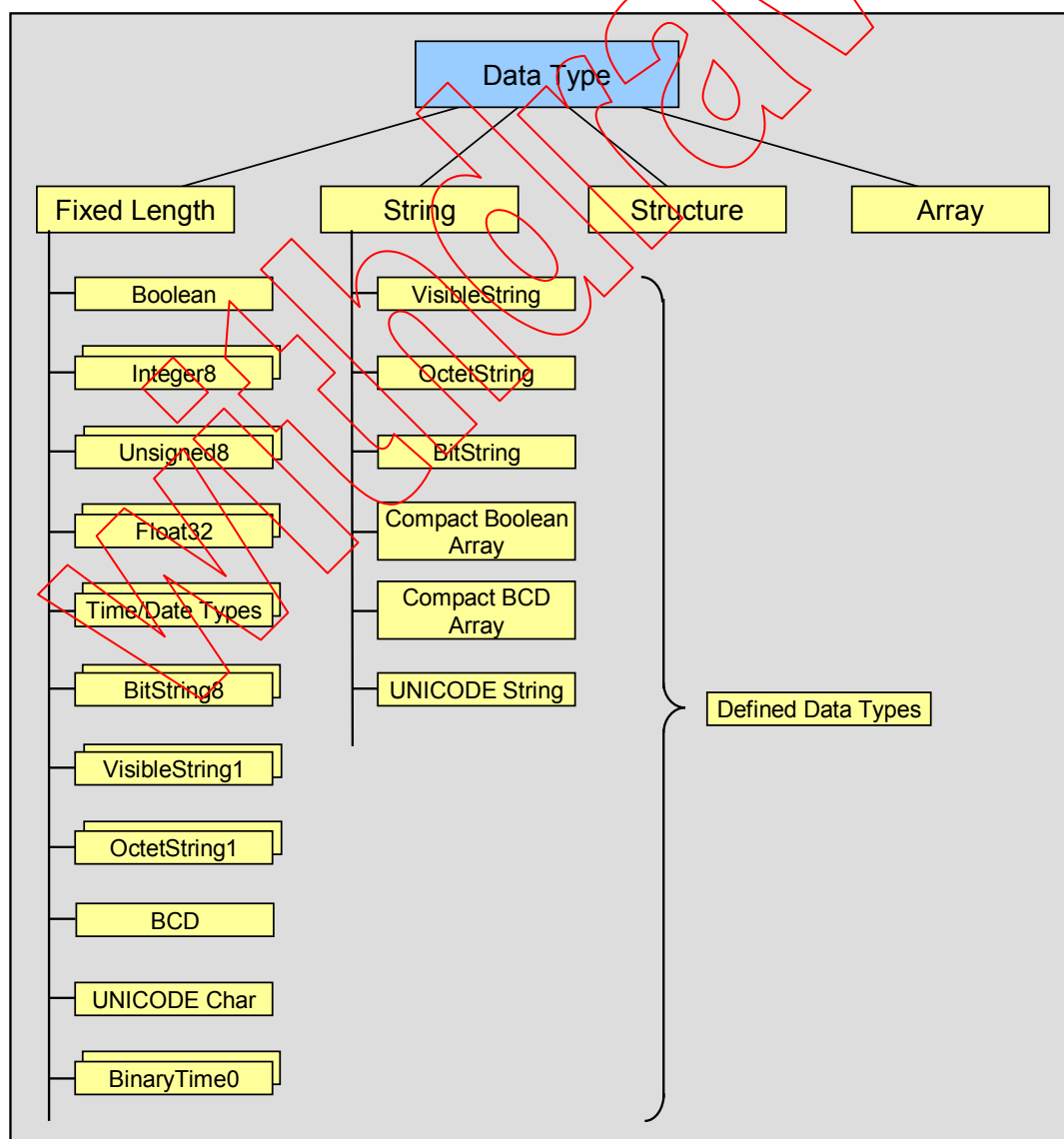


Figure 1 – Data type class hierarchy example

The data type definitions in Figure 1 are represented as a class/format/instance structure beginning with data type class entitled "Data type". The formats for data types are defined by the data type class and are represented in Figure 1.

The basic data classes are used to define fixed length and bitstring data types. Standard types taken from ISO/IEC 8824-1 are referred to as *simple* data types. Other standard basic data types are defined specifically for fieldbus applications and are referred to as *specific types*.

The constructed types specified in this standard are strings, arrays and structures. There are no standard types defined for arrays and structures.

5.1.2 Basic type overview

Most basic types are defined from a set of ISO/IEC 8824-1 types (simple types). Some ISO/IEC 8824-1 types have been extended for fieldbus specific use (specific types).

Simple types are ISO/IEC 8824-1 universal types. They are defined in this standard to provide them with fieldbus class identifiers.

Specific types are basic types defined specifically for use in the fieldbus environment. They are defined as simple class subtypes.

Basic types have a constant length. Two variations are defined, one for defining data types whose length is an integral number of octets, and one for defining data types whose length is bits.

NOTE Boolean, Integer, OctetString, VisibleString, and UniversalTime are defined in this standard for the purpose of assigning fieldbus class identifiers to them. This standard does not change their definitions as specified in ISO/IEC 8824-1.

5.1.3 Fixed length type overview

The length of Fixed length types is an integral number of octets.

5.1.4 Constructed type overview

5.1.4.1 Strings

A string is composed of an ordered set, variable in number, of homogeneously typed fixed-length elements.

5.1.4.2 Arrays

An array is composed of an ordered set of homogeneously typed elements. This standard places no restriction on the data type of array elements, but it does require that each element be of the same type. Once defined, the number of elements in an array may not be changed.

5.1.4.3 Structures

A structure is made of an ordered set of heterogeneously typed elements called fields. Like arrays, this standard does not restrict the data type of fields. However, the fields within a structure do not have to be of the same type.

5.1.4.4 Nesting level

This standard permits arrays and structures to contain arrays and structures. It places no restriction on the number of nesting levels allowed. However, the FAL services defined to access data provide for partial access to the level negotiated by the Initiate service. The default number of levels for partial access is one.

When an array or structure contains constructed elements, access to a single element in its entirety is always provided. Access to subelements of the constructed element is also provided, but only when explicitly negotiated during AR establishment or when explicitly preconfigured on pre-established ARs.

NOTE For example, suppose that a data type named "employee" is defined to contain the structure "employee name", and "employee name" is defined to contain "last name" and "first name". To access the "employee" structure, the FAL permits independent access to the entire structure and to the first level field "employee name". Without explicitly negotiating partial access to more than one level, independent access to "last name" or "first name" would not be possible; their values could only be accessed together as a unit through access to "employee" or "employee name".

5.1.5 Date and time type overview

Any date and time related data type, i.e. BinaryDate, TimeOfDay with and without date indication, TimeDifference with and without date indication, NetworkTime, NetworkTimeDifference, and TimeStamp is always related to a specific timescale.

Possible timescales are:

- Local time / local counter; used e.g. for working duration, ...
- Working clock; used e.g. for SendClock, Tinput, Toutput, SampleTime, ...
- Global time / Wall clock; Used e.g. for a comparable point in time, a start time, a stop time, ...

Usable timescales need to be strict monotonously rising, like TAI (temps atomique international), to allow any TimeDifference to be calculated.

Any usage of these data types is only sensible when having the information, to which timescale the value is related.

Only the data type TimeStamp offers the timescale related to and whether the time is synchronized. All other data types have no indication to which timescale they belong to, or whether the time is synchronized to other devices.

NOTE IEEE 802.1AS provides a mechanism to synchronize devices to the TAI timescale. Thus, no UTC (Universal Time Coordinated) information is available.

5.1.6 Specification of user defined data types

Users may find it necessary to define custom data types for their own applications. User defined types are supported by this standard as instances of data type classes.

User defined types are specified in the same manner that all FAL objects are specified. They are defined by providing values for the attributes specified for their class.

5.1.7 Transfer of user data

User data is transferred between applications by the FAL protocol. All encoding and decoding are performed by the FAL user.

The rules for encoding user data in FAL protocol data units is data type dependent. These rules are defined in IEC 61158-6-10. User-defined data types for which there are no encoding rules are transferred as a variable-length sequence of octets. The format of the data within the OctetString is defined by the user.

5.1.8 Data type overview

Table 5 gives an overview over all data types defined within here.

Table 5 – Data type overview

Data type numeric identifier	Data type name	Usage	Alias name	Definition
1	Boolean	this standard, IEC 61158-x-3	—	see 5.3.1.1
2	Integer8	this standard, IEC 61158-x-3	char	see 5.3.1.5.3.1
3	Integer16	this standard, IEC 61158-x-3	short	see 5.3.1.5.3.2
4	Integer32	this standard, IEC 61158-x-3	long	see 5.3.1.5.3.3
5	Unsigned8	this standard, IEC 61158-x-3	unsigned char	see 5.3.1.5.4.1
6	Unsigned16	this standard, IEC 61158-x-3	unsigned short	see 5.3.1.5.4.2
7	Unsigned32	this standard, IEC 61158-x-3	unsigned long	see 5.3.1.5.4.3
8	Float32	this standard, IEC 61158-x-3	float	see 5.3.1.5.1.1
9	VisibleString	this standard, IEC 61158-x-3	—	see 5.3.2.2
10	OctetString	this standard, IEC 61158-x-3	—	see 5.3.2.1
12	TimeOfDay with date indication	this standard, IEC 61158-x-3	TimeOfDay	see 5.3.1.4.5
13	TimeDifference	obsolete	—	see 5.3.1.4.4
15	Float64	this standard, IEC 61158-x-3	double	see 5.3.1.5.1.2
39	UNICODEString	this standard, IEC 61158-x-3	—	see 5.3.2.3
50	BinaryDate	this standard, IEC 61158-x-3	—	see 5.3.1.4.1
52	TimeOfDay without date indication	this standard, IEC 61158-x-3	—	see 5.3.1.4.3
53	TimeDifference with date indication	this standard, IEC 61158-x-3	—	see 5.3.1.4.5
54	TimeDifference without date indication	this standard, IEC 61158-x-3	—	see 5.3.1.4.6
55	Integer64	this standard, IEC 61158-x-3	—	see 5.3.1.5.3.4
56	Unsigned64	this standard, IEC 61158-x-3	—	see 5.3.1.5.4.4
58	NetworkTime	this standard, IEC 61158-x-3	—	see 5.3.1.7.2
59	NetworkTimeDifference	this standard, IEC 61158-x-3	—	see 5.3.1.7.3
60	TimeStamp	this standard, IEC 61158-x-3	—	see 5.3.1.7.4
101	Float32+Unsigned8	this standard, IEC 61158-x-3	—	see 5.3.3.2
102	Unsigned8+Unsigned8	this standard, IEC 61158-x-3	—	see 5.3.3.3
103	OctetString2+Unsigned8	this standard, IEC 61158-x-3	—	see 5.3.3.1

Data type numeric identifier	Data type name	Usage	Alias name	Definition
104	Unsigned16_S	this standard, IEC 61158-x-3	—	see 5.3.3.4
105	Integer16_S	this standard, IEC 61158-x-3	—	see 5.3.3.5
106	Unsigned8_S	this standard, IEC 61158-x-3	—	see 5.3.3.6
107	OctetString_S	this standard, IEC 61158-x-3	—	see 5.3.3.7
110	F message trailer with 4 octets	this standard, IEC 61158-x-3	—	see 5.3.3.8
111	F message trailer with 5 octets	this standard, IEC 61158-x-3	—	see 5.3.3.9
113	N2	this standard, IEC 61158-x-3	—	see 5.3.1.5.3.5
114	N4	this standard, IEC 61158-x-3	—	see 5.3.1.5.3.6
115	V2	this standard, IEC 61158-x-3	—	see 5.3.1.2.1
116	L2	this standard, IEC 61158-x-3	—	see 5.3.1.2.2
117	R2	this standard, IEC 61158-x-3	—	see 5.3.1.5.8
118	T2	this standard, IEC 61158-x-3	—	see 5.3.1.5.5
119	T4	this standard, IEC 61158-x-3	—	see 5.3.1.5.6
120	D2	this standard, IEC 61158-x-3	—	see 5.3.1.5.7
121	E2	this standard, IEC 61158-x-3	—	see 5.3.1.5.2.1
122	C4	this standard, IEC 61158-x-3	—	see 5.3.1.5.3.9
123	X2	this standard, IEC 61158-x-3	—	see 5.3.1.5.3.7
124	X4	this standard, IEC 61158-x-3	—	see 5.3.1.5.3.8
125	Unipolar2.16	this standard, IEC 61158-x-3	—	see 5.3.1.5.2.2
128 – 255	for this standard: reserved for IEC 61158-x-3: user specific types	IEC 61158-x-3	—	—
256 – 1 024	Reserved	this standard	—	see 5.3.1.6.1
1 025	UUID	this standard	—	see 5.3.1.6.1
1 026 – 65 535	Reserved	this standard	—	see 5.3.1.6.1
other	Reserved	this standard, IEC 61158-x-3	—	—

NOTE The x in IEC 61158-x-3 means the Type 3 specific parts 3, 4, 5, and 6 of IEC 61158 series.

5.2 Formal definition of data type objects

5.2.1 Data type class

5.2.1.1 Template

The data type class specifies the root of the data type class tree. Its parent class “top” indicates the top of the FAL class tree.

FAL ASE:	DATA TYPE ASE
CLASS:	DATA TYPE
CLASS ID:	5 (FIXED LENGTH & STRING), 6 (STRUCTURE), 12 (ARRAY)
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Data type Numeric Identifier
2	(o) Key Attribute: Data type Name
3	(m) Attribute: Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c) Constraint: Format = FIXED LENGTH STRING
4.1	(m) Attribute: Octet Length
5	(c) Constraint: Format = STRUCTURE
5.1	(m) Attribute: Number of Fields
5.2	(m) Attribute: List of Fields
5.2.1	(o) Attribute: Field Name
5.2.2	(m) Attribute: Field Data type
6	(c) Constraint: Format = ARRAY
6.1	(m) Attribute: Number of Array Elements
6.2	(m) Attribute: Array Element Data type

5.2.1.2 Attributes

Data type Numeric Identifier

This attribute identifies the numeric identifier of the related data type. The data type numeric identifier has a data type of Unsigned32.

Data type Name

This optional attribute identifies the name of the related data type.

Format

This attribute identifies the data type as a fixed-length, string, array, or data structure.

Octet Length

This conditional attribute defines the representation of the dimensions of the associated type object. It is present when the value of the format attribute is "FIXED LENGTH" or "STRING". For FIXED LENGTH data types, it represents the length in octets. For STRING data types, it represents the length in octets for a single element of a string.

Number of Fields

This conditional attribute defines the number of fields in a structure. It is present when the value of the format attribute is "STRUCTURE".

List of Fields

This conditional attribute is an ordered list of fields contained in the structure. Each field is specified by its number and its type. Fields are numbered sequentially from 0 (zero) in the order in which they occur. Partial access to fields within a structure is supported by identifying the field by number. This attribute is present when the value of the format attribute is "STRUCTURE".

Field Name

This conditional, optional attribute specifies the name of the field. It may be present when the value of the format attribute is "STRUCTURE".

Field Data type

This conditional attribute specifies the data type of the field. It is present when the value of the format attribute is "STRUCTURE". This attribute may itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the Embedded Data type description shown below is used.

Number of Array Elements

This conditional attribute defines the number of elements for the array type. Array elements are indexed starting at "0" through "n-1" where the size of the array is "n" elements. This attribute is present when the value of the format attribute is "ARRAY".

Array Element Data type

This conditional attribute specifies the data type for the elements of an array. All elements of the array have the same data type. It is present when the value of the format attribute is "ARRAY". This attribute may itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the Embedded Data type description shown below is used.

Embedded Data type Description

This attribute is used to recursively define embedded data types within a structure or array. The template below defines its contents. The attributes shown in the template are defined above in the data type class, except for the Embedded Data type attribute, which is a recursive reference to this attribute. It is used to define nested elements.

ATTRIBUTES:

1	(m) Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
2	(c) Constraint:	Format = FIXED LENGTH STRING
2.1	(m) Attribute:	Data type Numeric ID value
2.2	(m) Attribute:	Octet Length
3	(c) Constraint:	Format = STRUCTURE
3.1	(m) Attribute:	Number of Fields
3.2	(m) Attribute:	List of Fields
3.2.1	(m) Attribute:	Embedded Data type Description
4	(c) Constraint:	Format = ARRAY
4.1	(m) Attribute:	Number of Array Elements
4.2	(m) Attribute:	Embedded Data type Description

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

5.3.1.1.1 Boolean

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 1
2	Data type Name	= Boolean
3	Format	= FIXED LENGTH
4.1	Octet Length	= 1

This data type expresses a Boolean data type with the values TRUE (non zero) and FALSE (zero).

5.3.1.2 Bitstring types

5.3.1.2.1 Bit sequence V2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	115
2	Data type Name	=	V2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This data type expresses a Bitstring data type consisting of 16 single bits combined in two octets and coded according to Table 6.

Each single bit has the values TRUE (1) and FALSE (0).

Table 6 – V2 octets

Bit	8	7	6	5	4	3	2	1
Octet 1	15	14	13	12	11	10	9	8
Octet 2	7	6	5	4	3	2	1	0

5.3.1.2.2 Nibble L2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	116
2	Data type Name	=	L2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This data type expresses a Bitstring data type consisting of four nibbles combined in two octets. A nibble consists of four associated bits. The data type is coded according to Table 7.

The definition of a single nibble is not specified.

Table 7 – L2 octets

Bit	8	7	6	5	4	3	2	1
Octet 1	Nibble 3				Nibble 2			
Octet 2	Nibble 1				Nibble 0			

The definition of a single nibble is not specified.

5.3.1.3 Currency types

NOTE Intentionally blank

5.3.1.4 Date types

5.3.1.4.1 BinaryDate

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	50
2	Data type Name	=	BinaryDate
3	Format	=	FIXED LENGTH

4.1 Octet Length = 7

This data type is composed of six elements of unsigned values and expresses calendar date and time.

The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds.

The second element is an Unsigned8 data type and gives the fraction of an hour in minutes.

The third element is an Unsigned8 data type and gives the fraction of a day in hours with the most significant bit indicating Standard Time or Daylight Saving Time.

The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of the week and its lower five (5) bits give the day of the month.

The fifth element is an Unsigned8 data type and gives the month.

The last element is Unsigned8 data type and gives the year. The values 0 ... 50 correspond to the years 2000 to 2050; the values 51 ... 99 correspond to the years 1951 to 1999.

Note that the time zone is undefined.

5.3.1.4.2 TimeOfDay with date indication

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 12
2	Data type Name	= TimeOfDay with date indication
3	Format	= FIXED LENGTH
4.1	Octet Length	= 6

This data type is composed of two elements of unsigned values and expresses the time of day and the date. The first element is an Unsigned32 data type and gives the time after the midnight in milliseconds. The second element is an Unsigned16 data type and gives the date counting the days from 1984-01-01.

NOTE "TimeOfDay" is an alias name for this data type.

5.3.1.4.3 TimeOfDay without date indication

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 52
2	Data type Name	= TimeOfDay without date indication
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This data type is composed of one element of an unsigned value and expresses the time of day without date indication. The element is an Unsigned32 data type and gives the time after the midnight in milliseconds.

5.3.1.4.4 TimeDifference

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 13
2	Data type Name	= TimeDifference
3	Format	= FIXED LENGTH

4.1 Octet Length = 4 or 6

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The optional second element is an Unsigned16 data type that provides the difference in days.

This data type is obsolete; use the data types TimeDifference with date indication (see 5.3.1.4.2) or TimeDifference without date indication (see 5.3.1.4.3) instead.

5.3.1.4.5 TimeDifference with date indication

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 53
2	Data type Name	= TimeDifference with date indication
3	Format	= FIXED LENGTH
4.1	Octet Length	= 6

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The second element is an Unsigned16 data type that provides the difference in number of days.

5.3.1.4.6 TimeDifference without date indication

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 54
2	Data type Name	= TimeDifference without date indication
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This data type is composed of one element of an unsigned value that express the difference in time. The element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds.

5.3.1.5 Numeric types

5.3.1.5.1 Floating point types

5.3.1.5.1.1 Float32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 8
2	Data type Name	= Float32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This type has a length of four octets. The format for float32 is that defined by IEEE 754 as single precision.

NOTE "float" is an alias name for this data type.

5.3.1.5.1.2 Float64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 15
2	Data type Name	= Float64
3	Format	= FIXED LENGTH

4.1 Octet Length = 8

This type has a length of eight octets. The format for float64 is that defined by IEEE 754 as double precision.

NOTE “double” is an alias name for this data type.

5.3.1.5.2 Fixed point types

5.3.1.5.2.1 Fixed point value E2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 121
2	Data type Name	= E2
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This data type is a signed linear fixed-point value with a length of two octets having a sign-bit, eight binary places before the decimal point and seven binary places after it. The octets are coded according to Table 8.

Table 8 – E2 octets

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
	sign	integer						
Octet 2	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
	integer	fraction						

The value range of E2 is shown in Table 9.

Table 9 – E2 value range

Range of values	Resolution
$-256 \cdot 2^{-7} \rightarrow 256 \cdot 2^{-7}$	$2^{-7} = 0,0078125$

5.3.1.5.2.2 Unipolar2.16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 125
2	Data type Name	= Unipolar2.16
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This data type is an unsigned linear fixed-point value with two octets having two binary places before the decimal point and fourteen binary places after it. The octets are coded according to Table 10 with a value range shown in Table 11. The data type expresses a value in percent of a span.

Table 10 – Unipolar2.16 octets

Bit	8	7	6	5	4	3	2	1
Octet 1	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
	integer		fraction					